NCS35TB Stat Pac

For the HP-35s Calculator

Version 1.00.1

September 1, 2007

$$\sum\sum m_{ir}$$

Table of Contents

Dedication

To a new and brave friend, Pascal Gervaix,
Who took the time to make a difference in our troubled world.

General Introduction

Introduction

The NCS35TB Stat Pac contains a set of statistical programs for the HP-35s programmable scientific calculator. The Stat Pac contains the programs listed in the following table.

Program Name	Purpose
NCS_ACMA1	Calculates the autocorrelation and moving averages for the values in an array.
NCS_ANOCOV1	Calculates the elements of an analysis of covariance (ANOCOV) table.
NCS_ANOVA_ONE1	Calculates the elements of a one-way ANOVA table.
NCS_ANOVA_TWO1	Calculates the elements of a two-way ANOVA table with no replications.
NCS_ANOVA_TWO2	Calculates the elements of a two-way ANOVA table with replications.
NCS_BARTCHI1	Calculates the Bartlett Chi-square statistic for an array of sample variances.
NCS_BEHRENS1	Implements the Behrens-Fisher test for paired and unpaired arrays.
NCS_BEST_LR1	Finds the best linearized regression for a set of user-selected transformations.
NCS_CONTABLE1	Implements $n \times k$ contingency table calculations.
NCS_CRESFIT1	Implements nonlinear regression.
NCS_DAP1	Calculates the statistics related to differences among proportions.
NCS_INTRACLASS1	Calculates the intraclass correlation coefficient for a table of $k \times n$ observations.
NCS_KENDALL1	Calculates the Kendall concordance coefficient for a matrix of rank values.
NCS_KRUSKAL1	Implements the Kruskal-Wallis test to determine if several groups of ranked data belong to a single population.
NCS_LATIN_SQR1	Calculates the elements of a Latin square ANOVA table.
NCS_LR1	Implements linear regression with post regression ANOVA table and additional statistics.
NCS_LR2	A version of NCS_LR1 that supports weighted data.
NCS_MA1	Calculates the moving average for a set of observations.
NCS_MA2	Calculates the moving average for a set of observations. Also calculates the standard deviation, the confidence interval for the moving average, and the confidence interval for the standard deviation.
NCS_MANN1	Calculates the Mann-Whitney test statistic for an array of paired or unpaired ranked values.
NCS_MEANSSDF1	Implements the mean-square successive difference statistic to test the randomness of an array of values.
NCS_MLR1	Implements multiple linear regression between three variables.
NCS_MLR2	Implements multiple linear regression between two variables.

Program Name	Purpose
NCS_MOMENTS1	Calculates statistical moments.
NCS_MOMENTS1	Calculates statistical moments for weighted data.
NCS_PAIRED1	Implements the Student-t test for the means of paired data.
NCS_POLY1	Implements quadratic regression between two variables.
NCS_RANKER1	Calculates the numeric ranks for an array of values.
NCS_SPEARANK1	Calculates the Spearman rank correlation coefficient for an array of paired ranked values.
NCS_UNPAIRED1	Implements the Student-t test for the means of unpaired data.

New Features

Many programs in the NCS35TB Stat Pac implement legacy statistical programs found in the Stat Pacs of the HP-65, HP-67, and HP-41C calculators. The implementation for the HP-35s, which is written from scratch, takes advantage of the new features of the calculator. The highlight of these new features is the ability to store data in the unnamed registers, using indirect addressing. Some programs store the raw data, while others store the statistical summations. In the latter case, the programs allow you to access the statistical summations for the various data groups to further add or remove data. Either approach is new in the Stat Pac and makes use of the rich number of unnamed registers. The various programs also store data as 2D or 3D vectors to further economize data storage.

Consistent Interface

The programs in the NCS35TB Stat Pac have a consistent user interface. The following labels perform similar steps in all the programs:

- XEQ I initializes the program.
- XEQ A adds data.
- XEQ D deletes data.
- XEQ C performs statistical calculations.
- XEQ U displays the name of the program.

Piracy Issues

Numerous hours have gone into planning, writing, testing, and documenting the statistical programs. The documentation, which includes the source code, is available in PDF file format. As such, such files can be duplicated almost effortlessly with a few mouse clicks. You are most welcome to make as many archival copies for yourself as your heart desires. Handing over copies to others for usage does not honor yourself or the developer of the NCS35TB Stat Pac. Plans to possibly update the statistical programs and offer implementations for future HP calculator models depend on your support. So please help us to help you.

Disclaimer

While every effort has been made to make the programs error-free, by using the programs in the NCS35TB Stat Pac you agree not to hold Namir Shammas and any other associates of his responsible for any kind physical, financial, and emotional losses, damages, and injuries resulting from the use of the software.

The code for the NCS35TB Stat Pac was written from the ground up based on the equations shown for the different statistical calculations. No legacy code (such as the ones used for the HP-65, HP-67, and HP-41C) was used and/or adapted for the statistical programs.

Autocorrelation/Moving Average Program NCS_ACMA1

Overview

The NCS_ACMA1 program calculates the autocorrelation and moving averages for the values in an array. The program offers the following features:

1. Storing the data points of a single variable.
2. The ability to delete, edit, view, and swap data points. Thus the program incorporates basic operations to manage the data points.
3. Calculating the autocorrelation values at different offset values.
4. Calculating the moving averages (and optionally the associated standard deviation values) averaged over user-specified periods.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to press the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value for x.	X	R/S	Number of data points		
3	Delete a data point.		XEQ D	I?		
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message	Index of data point to delete	R/S	Number of data points		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	INVALID DATA.					
4	Edit a data point.		XEQ E	I?		
	Enter the index of the data point to edit. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	X?		
	Enter a new value for x.	x	R/S			
5	View a data point.		XEQ V	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	Data point	Index	
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	Data point	Index	
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1		
6	View all data points (Note the program displays the results using the PSE statement.) Note: If you spot an error in the data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	Data point	Index	
7	Swap data points.		XEQ S	I?		
	Enter the index of the	Index	R/S	J?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	first data point. If you enter an invalid index the program displays the message INVALID DATA.	of the first data point				
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S			
8	Calculate the autocorrelations.		XEQ C	N?		
	Enter the number of autocorrelations to calculate. The program displays the index 1 and the first autocorrelation coefficient.	N	R/S	Autocorrelation coefficient	Index	
	View the second autocorrelation coefficient.		R/S	Autocorrelation coefficient	Index	
	Repeat the above substeps to view the remaining autocorrelation coefficients.		R/S	Autocorrelation coefficient	Index	
9	Calculate the moving averages.		XEQ M	N?		
	Enter the number of values in array x to use in calculating the moving averages. The program displays the index 1 and the first moving average.	N	R/S	Moving average	Index	
	View the second moving average.		R/S	Moving average	Index	
	Repeat the above substeps to view the remaining moving averages.		R/S	Moving average	Index	
10	Calculate the moving		XEQ N	N?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	averages and their standard deviations.					
	Enter the number of values in array x to use in calculating the moving averages. The program displays the index 1 and the first moving average.	N	R/S	Moving average	Index	sdev
	View the standard deviation (in the Y stack register)		R↓	Index	sdev	
	View the second moving average.		R/S	Moving average	Index	
	View the standard deviation (in the Y stack register)		R↓	Index	sdev	
	Repeat the above substeps to view the remaining moving averages and their standard deviations.		R/S	Moving average	Index	sdev
	View the standard deviation (in the Y stack register)		R↓	Index	sdev	
11	Display the program name. The program pauses to display the program name.		XEQ U	NCS-ACMA1		

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	C	Number of data points	1	Index invalid when set
D	Delete an existing data point	I	Index	2	Set = view the standard deviation values
E	Edit a data point	J	Index		
V	View a data point	O	Offset value (integer part of		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
			register I)		
W	View all data points	N	Number of autocorrelations to calculate		
I	Initialize	X	x		
S	Swap two data points	1	Data point #1		
C	Calculate the autocorrelations	2	Data point #2		
M	Calculate moving averages.	3	Data point #3		
N	Calculate moving average and the standard deviation for the moving averages.	…			
U	Display program name	#C	Data point #C		
Labels for internal use					
H	Validate indices and display error message if an index is invalid				
G	Prompt for x				
R	Display the message INVALID DATA				

Program Listing

Line	Command	Comments
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	STOP	
A011	GTO A001	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index

Line	Command	Comments
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO– C	
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	Validate index
S006	RCL (I)	Recall data points and place them on the stack
S007	RCL (J)	

Line	Command	Comments
S008	STO (I)	Store data point in swapped indices
S009	x↔y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	FS? 1	Index out or range?
V005	GTO R001	
V006	RCL (I)	Recall data point
V007	STOP	
V008	1	
V009	RCL I	
V010	+	
V011	STO I	
V012	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate regression coefficients
C002	INPUT N	Prompt the user for the number of autocorrelation values
C003	1	
C004	RCL N	
C005	1E3	
C006	÷	
C007	+	
C008	STO I	Store value for outer loop control variable
C009	1	Start of outer loop
C010	RCL C	
C011	RCL I	
C012	IP	
C013	STO O	Let register O store the integer part of loop control variable I
C014	-	
C015	1E3	

Line	Command	Comments
C016	÷	
C017	+	
C018	STO J	Store value for inner loop control variable
C019	ClΣ	Clear statistical registers
C020	RCL (J)	Start inner loop and get x(J)
C021	RCL O	
C022	STO+ J	
C023	R↓	
C024	RCL (J)	Get x(J + Offset)
C025	Σ+	Accumulate x(J) and x(J + Offset) in statistical registers
C026	RCL O	
C027	STO- J	Restore J
C028	ISG J	
C029	GTO C020	End of inner loop
C030	SF 10	
C031	I,r	This is a “tag” created using EQN button
C032	PSE	Display output tag
C033	CF 10	
C034	RCL O	
C035	r	Get the autocorrelation coefficient
C036	STOP	Display index and autocorrelation coefficient
C037	ISG I	
C038	GTO C009	End of outer loop
C039	RTN	
M001	LBL M	Calculate moving averages
M002	CF 2	Clear flag for calculating the standard deviation
M003	INPUT N	Enter the period for calculating the moving averages
M004	1	
M005	RCL N	
M006	1E3	
M007	÷	
M008	+	
M009	STO I	Calculate the loop control used to calculate the initial moving average
M010	ClΣ	Clear the statistical registers
M011	RCL (I)	Start loop to calculate initial moving average value
M012	ENTER	
M013	Σ+	
M014	ISG I	
M015	GTO M011	End of loop that calculates initial moving average value
M016	SF 10	
M017	I,MA	This is a “tag” created using EQN button
M018	PSE	Display output tag

Line	Command	Comments
M019	FS?2	
M020	sx	
M021	1	
M022	x̄	
M023	STOP	Display index and the initial moving average
M024	1	
M025	RCL C	
M026	RCL N	
M027	STO J	Store N in register J to initialize the index of the array element to add to statistical summations
M028	-	
M029	1E3	
M030	÷	
M031	+	
M032	STO I	Set the loop control for the outer loop= 2 + (C-N)/1000
M033	RCL (I)	Start of loop to calculate the moving averages
M034	ENTER	
M035	Σ-	Subtract x(I) from the statistical summations
M036	1	
M037	STO+ J	Set the index of the next array element to add to stat sums
M038	RCL (J)	
M039	ENTER	
M040	Σ+	Add x(I+N-1) to the summations
M041	I,MA	This is a “tag” created using EQN button
M042	PSE	Display output tag
M043	FS? 2	
M044	sx	
M045	RCL I	
M046	IP	
M047	1	
M048	+	
M049	x̄	
M050	STOP	Display index and the first moving average
M051	ISG I	
M052	GTO M033	End of the loop
M053	CF 10	
M054	RTN	
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	RCL X	
G004	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	Enter this statement using the EQN key.

Line	Command	Comments
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-ACMA1	This is a "tag" created using EQN button
U004	PSE	Display output tag
U005	CF 10	
U006	RTN	
N001	LBL N	Calculate moving averages and their standard deviations
N002	SF 2	
N003	XEQ M003	
N004	CF 2	
N005	RTN	

Example

Let's look at an example that exercises the program. We want to calculate the autocorrelations and moving averages for the variable x whose values appear in the next table:

Index	x
1	120
2	122
3	125
4	123
5	121
6	123
7	125
8	126
9	124
10	125

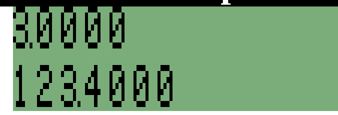
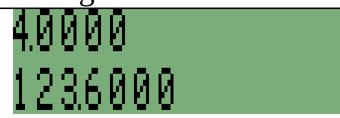
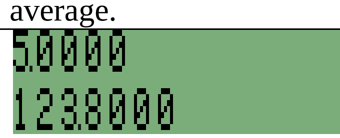
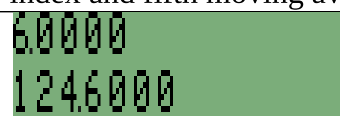
After you enter the above data perform the following:

1. Calculate the first five autocorrelations.
2. Calculate the six moving averages each taken over 5 consecutive periods.

The following table shows the steps using the FIX 4 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0.0000
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	120	R/S	1.0000
3	Add the second data point.		XEQ A	X?

Step #	Step/Substep	Input	Command	Output
	Enter a value for x.	122	R/S	2.0000
4	Repeat step 3 to enter all the other data points.			
5	Calculate first five autocorrelations.		XEQ C	N?
	Enter the number of autocorrelations to calculate. The program displays the first autocorrelation.	5	R/S	1.0000 0.4005 Displays the values for the index and first autocorrelation
	View the second autocorrelation.		R/S	2.0000 -0.2969 Displays the values for the index and second autocorrelation
	View the third autocorrelation.		R/S	3.0000 0.0902 Displays the values for the index and third autocorrelation
	View the fourth autocorrelation.		R/S	4.0000 0.7661 Displays the values for the index and fourth autocorrelation
	View the fifth autocorrelation.		R/S	5.0000 0.7295 Displays the values for the index and fifth autocorrelation
6	Calculate the moving averages.		XEQ M	N?
	Enter the number of periods. The program displays the first moving average.	5	R/S	1.0000 122.2000 Displays the values for the index and first moving average.
	View the second moving average.		R/S	2.0000 122.8000 Displays the values for the index and second moving average.

Step #	Step/Substep	Input	Command	Output
	View the third moving average.		R/S	 Displays the values for the index and third moving average.
	View the fourth moving average.		R/S	 Displays the values for the index and fourth moving average.
	View the fifth moving average.		R/S	 Displays the values for the index and fifth moving average.
	View the sixth moving average.		R/S	 Displays the values for the index and sixth moving average.

One-Way ANOCOV Table Program NCS_ANOCOV1

Overview

The NCS_ANOCOV1 program calculates the elements of an analysis of covariance (ANOCOV) table and offers the following features:

1. Handles data for multiple data groups of (x, y) data. Each group has a set of replicated observations that are not replicated.
2. Adds observations to the current group.
3. Starts a new data group.
4. Edits observations for the currently edited group.
5. Calculates and displays the elements of the ANOCOV table.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

The sum and sum of squares for variable x:

$$Sx_j = \sum_{j=1}^{k} x_{ij}$$

$$TSSx = \sum\sum x_{ij}^2 - (\sum\sum x_{ij})^2 / \sum n_i$$

$$ASSx = \sum_i \frac{\left(\sum_j x_{ij}\right)^2}{n_i} - (\sum\sum x_{ij})^2 / \sum n_i$$

$$WSSx = TSSx - ASSx$$

The sum and sum of squares for variable y:

$$Sy_j = \sum_{j=1}^{k} y_{ij}$$

$$TSSy = \sum\sum y_{ij}^2 - (\sum\sum y_{ij})^2 / \sum n_i$$

$$ASSy = \sum_i \frac{\left(\sum_j y_{ij}\right)^2}{n_i} - (\sum\sum y_{ij})^2 / \sum n_i$$

$$WSSy = TSSy - ASSy$$

The degrees of freedom:

$$df_1 = k - 1$$

$$df_2 = \sum n_i - k$$

The mean sum of squares:

$$AMSx = ASSx / df_1$$

$$WMSx = WSSx / df_2$$

$$AMSy = ASSy / df_1$$

$$WMSy = WSSy / df_2$$

The F statistics:

$$Fx = AMSx / WMSx$$

$$Fy = AMSy / WMSy$$

The sum of products:

$$TSP = \sum\sum x_{ij}y_{ij} - (\sum x_{ij})(\sum y_{ij}) / \sum n_i$$

$$ASP = \sum_i \frac{\left(\sum_j x_{ij}\right)\left(\sum_j y_{ij}\right)}{n_i} - (\sum x_{ij})(\sum y_{ij}) / \sum n_i$$

$$WSP = TSP - ASP$$

The residual sum of squares:

$$TSS\hat{y} = TSSy - (TSP)^2 / TSSx$$

$$WSS\hat{y} = WSSy - (WSP)^2 / WSSx$$

$$ASS\hat{y} = TSS\hat{y} - WSS\hat{y}$$

The degrees of freedom:

$$df_3 = k - 1$$

$$df_4 = \sum n_i - k - 1$$

The residual mean squares:

$$AMS\hat{y} = ASS\hat{y} / df_3$$

$$WMS\hat{y} = WSS\hat{y} / df_4$$

The F statistic:

$$F = AMS\hat{y} / WMS\hat{y}$$

The one-way ANOCOV table is:

					Residuals			
	Degrees of freedom	SSx	SP	SSy	Degrees of freedom	SS$\hat{y}$	MS$\hat{y}$	F Statistic
Among means	df_1	ASSx	ASP	ASSy	df_3	ASS$\hat{y}$	AMS$\hat{y}$	F
Within groups	df_2	WSSx	WSP	WSSy	df_4	WSS$\hat{y}$	WMS$\hat{y}$	
Total		TSSx	TSP	TSSy		TSS$\hat{y}$		

If calculated F statistic does not exceed the value $F_{\alpha,df1,df2}$, we cannot reject, at a 100(1-α)% confidence level, the null hypothesis stating the various groups have the same statistical covariance.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	0	
2	Enter the values in the tabulated data. The program prompts you to input the tabulated values by rows.		XEQ A	X?	
	Enter the value for x.	x	R/S	Y?	
	Enter the value for y.	y	R/S	Data count	
	To enter the next pair of (x, y) data either go back to		R/S	X?	

Step #	Step/Substep	Input	Command	Output X	Output Y
	step 2, or press R/S, or resume with the above steps.				
3	To delete a data point in the current group.		XEQ D	X?	
	Enter the value for x.	x	R/S	Y?	
	Enter the value for y.	y	R/S	Data count	
	To delete another set of (x, y) data either go back to step 3, or press R/S, or resume with the above steps.		R/S	X?	
4	Begin entering the data of a new group.		XEQ B	X?	
	Enter the value for x.	x	R/S	Y?	
	Enter the value for y.	y	R/S	1	
	To delete another set of (x, y) data either go back to step 2, or press R/S, or resume with the above steps.		R/S	X?	
5	Calculate the elements of the ANOCOV table and view them. You first view the values of TSSx and ASSx. The steps executed in label C alter the values of some statistical summations that are initialized by ***other*** labels. Consequently you cannot repeatedly execute label C and obtain the same results. To re-view the results, execute the command XEQ V, as discussed in step 6.		XEQ C	ASSx	TSSx
	View the values of WSSx and TSSy.		R/S	TSSy	WSSx
	View the values of ASSy and WSSy.		R/S	WSSy	ASSy

Step #	Step/Substep	Input	Command	Output X	Output Y
	View the values for df1 and df2.		R/S	df2	df2
	View the values of Fx and Fy.		R/S	Fy	Fx
	View the values of TSP and ASP.		R/S	ASP	TSP
	View the values of WSP and TSS$\hat{y}$.		R/S	TSS$\hat{y}$	WSP
	View the values of WSS$\hat{y}$ and ASS$\hat{y}$.		R/S	ASS$\hat{y}$	WSS$\hat{y}$
	View the values of df3 and df4.		R/S	df4	df3
	View the values of AMS$\hat{y}$ and WMS$\hat{y}$.		R/S	WMS$\hat{y}$	AMS$\hat{y}$
	View the value of the F statistic.		R/S	F	
6	To view the ANOCOV table elements again.		XEQ V	ASSx	TSSx
	Perform the same substeps as in step 5 to view the other elements of the ANOCOV table.				
7	Display the program name. The program pauses to display the program name.		XEQ U	NCS-ANOCOV1	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A	ΣΣ x	1	Used
B	Begin a new group	B	ΣΣy	2	Used
C	Calculate and view ANOCOV table	C	$\Sigma\Sigma x^2$		
D	Delete a data point	D	$\Sigma\Sigma y^2$		
I	Initialize program	E	ΣΣxy		
U	Display program name	F	TSSx		
V	Review results	G	ASSx		
		H	WSSx		
Internally used labels		I	TSSy		
		J	ASSy		
		K	WSSy		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		L	df1, df3		
		M	df2		
		N	Fx		
		O	Fy		
		P	TSP		
		Q	ASP		
		R	WSP		
		S	TSS$\hat{y}$		
		T	WSS$\hat{y}$		
		U	ASS$\hat{y}$		
		V	k		
		W	∑n(i)		
		X	AMS$\hat{y}$		
		Y	WMS$\hat{y}$		
		Z	F statistic		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	CF 2	
I003	CF 1	
I004	0	
I005	STO A	Initialize statistical sums and counters
I006	STO B	
I007	STO C	
I008	STO D	
I009	STO E	
I010	STO V	
I011	STO W	
I012	STO G	
I013	STO J	
I014	STO Q	
I015	ClΣ	Clear statistical registers
I016	RTN	
A001	LBL A	Add a new data point
A002	INPUT X	Prompt for x and y
A003	INPUT Y	
A004	RCLY	
A005	RCL X	
A006	Σ+	
A007	CF 2	
A008	STOP	

Line	Command	Comments
A009	GTO A001	Resume to prompt for more data
D001	LBL D	Delete a data point
D002	INPUT X	Prompt for x and y
D003	INPUT Y	
D004	RCLY	
D005	RCL X	
D006	Σ-	
D007	CF 2	
D008	STOP	
D009	GTO D001	Resume to prompt for more data
B001	LBL B	Begin a new group
B002	1	
B003	STO+ V	
B004	n	
B005	STO+ W	Update grand sum of n
B006	Σx	
B007	STO+ A	Update grand sum of x
B008	Σy	
B009	STO+ B	Update grand sum of y
B010	Σx^2	
B011	STO+ C	Update grand sum of x squared
B012	Σy^2	
B013	STO+ D	Update grand sum of y squared
B014	Σxy	
B015	STO+ E	Update grand sum of x times y
B016	Σx	
B017	x^2	
B018	n	
B019	÷	
B020	STO+ G	Update Assx
B021	Σy	
B022	x^2	
B023	n	
B024	÷	
B025	STO+ J	Update Assy
B026	Σx	
B027	Σy	
B028	×	
B029	n	
B030	÷	
B031	STO+ Q	Update ASP
B032	FS? 1	
B033	RTN	

Line	Command	Comments
B034	ClΣ	
B035	GTO A001	
C001	LBL C	Calculate ANOCOV
C002	FS? 2	
C003	GTO V001	
C004	SF 1	
C005	XEQ B001	
C006	CF 1	
C007	RCL C	
C008	RCL A	
C009	x^2	
C010	RCL W	
C011	÷	
C012	STO- G	Update Assx
C013	-	
C014	STO F	Store TSSx
C015	RCL G	
C016	-	
C017	STO H	Store WSSx
C018	RCL D	
C019	RCL B	
C020	x^2	
C021	RCL W	
C022	÷	
C023	STO- J	Update Assy
C024	-	
C025	STO I	Store TSSy
C026	RCL J	
C027	-	
C028	STO K	Store WSSy
C029	RCL V	
C030	1	
C031	-	
C032	STO L	Store df1 (also =df3)
C033	RCL W	
C034	RCL V	
C035	-	
C036	STOM	Store df2
C037	RCL E	
C038	RCL A	
C039	RCL B	
C040	×	
C041	RCL W	

Line	Command	Comments
C042	÷	
C043	STO- Q	Update ASP
C044	-	
C045	STO P	Store TSP
C046	RCL Q	
C047	-	
C048	STO R	Store WSP
V001	LBL V	Review results
V002	SF 2	
V003	SF 10	
V004	TSSX.ASSX	This is a “tag” created using the EQN button
V005	PSE	Display output tag for TSSx and ASSx
V006	RCL F	
V007	RCL G	
V008	STOP	Display TSSx and ASSx
V009	WSSX,TSSY	This is a “tag” created using the EQN button
V010	PSE	Display output tag for WSSx and TSSy
V011	RCL H	
V012	RCL I	
V013	STOP	Display WSSx and TSSy
V014	ASSY,WSSY	This is a “tag” created using the EQN button
V015	PSE	Display output tag for ASSy and WSSy
V016	RCL J	
V017	RCL K	
V018	STOP	Display ASSy and WSSy
V019	DF1,DF2	This is a “tag” created using the EQN button
V020	PSE	Display output tag for df1 and df2
V021	RCL L	
V022	RCL M	
V023	STOP	Display df1 and df2
V024	RCL G	
V025	RCL L	
V026	÷	
V027	STO N	Store Fx
V028	RCL H	
V029	RCL M	
V030	÷	
V031	STO÷ N	
V032	RCL J	
V033	RCL L	
V034	÷	
V035	STO O	Store Fy
V036	RCL K	

Line	Command	Comments
V037	RCL M	
V038	÷	
V039	STO÷ O	
V040	FX,FY	This is a “tag” created using the EQN button
V041	PSE	Display output tag for Fx and Fy
V042	RCL N	
V043	RCL O	
V044	STOP	Display Fx and Fy
V045	TSP,ASP	This is a “tag” created using the EQN button
V046	PSE	Display output tag for TSP and ASP
V047	RCL P	
V048	RCL Q	
V049	STOP	Display TSP and ASP
V050	RCL I	
V051	RCL P	
V052	x^2	
V053	RCL F	
V054	÷	
V055	-	
V056	STO S	Store TSS$\hat{y}$
V057	RCL K	
V058	RCL R	
V059	x^2	
V060	RCL H	
V061	÷	
V062	-	
V063	STO T	Store WSS$\hat{y}$
V064	RCL S	
V065	x↔y	
V066	-	
V067	STO U	Store ASS$\hat{y}$
V068	WSP,TSSY^	This is a “tag” created using the EQN button
V069	PSE	Display output tag for WSP and TSS$\hat{y}$
V070	RCL R	
V071	RCL S	
V072	STOP	Display WSP and TSS$\hat{y}$
V073	WSSY^,ASSY^	This is a “tag” created using the EQN button
V074	PSE	Display output tag for WSS$\hat{y}$ and ASS$\hat{y}$
V075	RCL T	
V076	RCL U	
V077	STOP	Display WSS$\hat{y}$ and ASS$\hat{y}$
V078	RCL U	
V079	RCL L	

Line	Command	Comments
V080	÷	
V081	STO X	
V082	STO Z	Store F
V083	RCL T	
V084	RCL M	
V085	1	
V086	-	
V087	÷	
V088	STO Y	Store WMS$\hat{y}$
V089	STO÷ Z	Update F
V090	DF3,DF4	This is a "tag" created using the EQN button
V091	PSE	Display output tag for df3 and df4
V092	RCL L	
V093	RCL M	
V094	1	
V095	-	
V096	STOP	Display df3 and df4
V097	AMSY^,WMSY^	This is a "tag" created using the EQN button
V098	PSE	Display output tag for AMS$\hat{y}$ and WMS$\hat{y}$
V099	RCL X	
V100	RCL Y	
V101	STOP	Display AMS$\hat{y}$ and WMS$\hat{y}$
V102	F STAT	This is a "tag" created using the EQN button
V103	PSE	Display output tag for F statistic
V104	CF 10	
V105	RCL Z	
V106	RTN	Display F statistic
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-ANOCOV1	This is a "tag" created using the EQN button
U004	PSE	Display program name
U005	CF 10	
U006	RTN	

Example

Let's look at an example that uses the ANOCOV program. The next table contains a set of (x, y) observations. We want to test the hypotheses that the various groups have the same statistical covariance, at a 95% confidence level (that is, for $\alpha = 0.05$).

			Observations			
			1	2	3	4
Data Groups	1	x	3	2	1	2
		y	10	8	8	11
	2	x	4	3	3	5
		y	12	12	10	13
	3	x	1	2	3	1
		y	6	5	8	7

After you enter the data points, calculate and display the elements of the ANOCOV table. Armed with the results in the ANOCOV table, test the null hypothesis at a 95% confidence level.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program		XEQ I	0.00
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	3	R/S	Y?
	Enter a value for y.	10	R/S	1.00
	Proceed to enter the second data point.		R/S	X?
	Enter a value for x.	2	R/S	Y?
	Enter a value for y.	8	R/S	2.00
	Repeat the above steps for the remaining values in the same group.			
3	Begin entering data for the second group.		XEQ B	X?
	Enter the first value for variable x in the second group.	4	R/S	Y?
	Enter the value for y.	12	R/S	1.00
	Repeat the steps in steps 2 and 3 to enter the remaining data.			

Step #	Step/Substep	Input	Command	Output
4	Calculate and display the ANOCOV table. You first view the values of TSSx and ASSx.		XEQ C	17.00 9.50 Displays the values for TSSx and ASSx
	View the values of WSSx and TSSy.		R/S	7.50 71.67 Displays the values for WSSx and TSSy
	View the values of ASSy and WSSy.		R/S	55.17 16.50 Displays the values for ASSy and WSSy
	View the values for df1 and df2.		R/S	2.00 9.00 Displays the values df1 and df2
	To view the values of Fx and Fy.		R/S	5.70 15.05 Displays the value for the Fx and Fy statistics
	View the values of TSP and ASP.		R/S	27.00 20.75 Displays the values for the TSP and ASP
	View the values of WSP and TSS$\hat{y}$.		R/S	6.25 28.78 Displays the values for WSP and TSS$\hat{y}$
	View the values of WSS$\hat{y}$ and ASS$\hat{y}$.		R/S	11.29 17.49 Displays the values for WSS$\hat{y}$ and ASS$\hat{y}$
	View the values of df3 and df4.		R/S	2.00 8.00 Displays the values for df3 and df4

Step #	Step/Substep	Input	Command	Output
	View the values of AMS$\hat{y}$ and WMS$\hat{y}$.		R/S	8.75 1.41 Displays the values for AMS$\hat{y}$ and WMS$\hat{y}$
	View the value of the F statistic.		R/S	1.41 6.20 Displays the values for the F statistic

The One-way ANOCOV table is:

					Residuals			
	Degrees of freedom	SSx	SP	SSy	Degrees of freedom	SS$\hat{y}$	MS$\hat{y}$	F Statistic
Among means	2	9.5	20.75	55.17	2	17.49	8.75	6.20
Within groups	9	7.5	6.25	16.5	8	11.29	1.41	
Total		17.0	27.0	71.67		28.78		

Since the calculated F statistic is 6.20 and exceeds the value $F_{0.05,2,8}$ = 4.459, we cannot accept, at a 95% confidence level, the null hypotheses stating that the various groups have the same statistical covariance.

Appendix A contains a new approximation that I recently developed for inverse F function. The appendix also contains tables for selected values.

One-Way ANOVA Table Program NCS_ANOVA_ONE1

Overview

The NCS_ANOVA_ONE1 program calculates the elements of a one-way ANOVA table and offers the following features:

1. Process a set of variables that can have different number of observations.
2. Add and delete observations for each variable.
3. Recall the statistical summations of previously processed variables to add more data or delete data.
4. Display the basic statistics of the current variable.
5. Calculate and display the elements of the one-way ANOVA table.

While this program is based on similar versions found in the Stat Pacs of the HP-67 and HP-41C, feature number 3 is an important enhancement. The legacy versions of the ANOVA program allow you to delete invalid or erroneous observations of a variable **before** you move on to entering the data for subsequent variables. Once you proceed in entering values for subsequent variables, adding more observations or deleting values of previous variables is not possible! The program NCS_ANOVA_ONE1 uses the 3-D vector format to store and recall the values for the n, Σx, and Σx^2, for each variable, in unnamed memory registers using indirect addressing. This strategy allows you to recall the summation data for previously entered variable in order to perform appropriate updates. The program never stores all of raw data you enter, but instead stores the relevant statistical registers for each variable.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

The mean of the observations in the ith sample for i = 1, 2, 3, .., k (the number of variables):

$$\bar{x}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij}$$

The standard deviation of the observations in the ith sample:

$$s_i = \sqrt{[(\sum_{j=1}^{n_i} x_{ij}^2 - n_i \bar{x}_i^2) / (n_i - 1)]}$$

The sum of the observations in the i^{th} sample:

$$sum_i = \sum_{j=1}^{n_i} x_{ij}$$

The total sum of squares:

$$TSS = \sum_{i=1}^{k} \sum_{j=1}^{n_i} x_{ij}^2 - \left(\sum_{i=1}^{k} \sum_{j=1}^{n_i} x_{ij}\right)^2 / \left(\sum_{i=1}^{k} n_i\right)$$

The treatment sum of squares:

$$TrSS = \sum_{i=1}^{k} \frac{\left(\sum_{j=1}^{n_i} x_{ij}\right)^2}{n_i} - \left(\sum_{i=1}^{k} \sum_{j=1}^{n_i} x_{ij}\right)^2 / \left(\sum_{i=1}^{k} n_i\right)$$

The error sum of squares:

$$ESS = TSS - TrSS$$

The treatment degrees of freedom:

$$df_1 = k - 1$$

The error degrees of freedom:

$$df_2 = \sum_{i=1}^{k} n_i - k$$

The total degrees of freedom:

$$df_3 = \sum_{i=1}^{k} n_i - 1$$

The treatment mean square:

$$TrMS = TrSS / df_1$$

The error mean square:

$$EMS = ESS / df_2$$

The F statistic (with degrees of freedom df1 and df2):

$$F = TrMS / EMS$$

If the calculated F statistic does not exceed the value of $F_{\alpha,df1,df2}$ then we cannot reject, at a 100(1-α)% confidence level, the null hypothesis stating that the variables have the same statistical mean.

The one-way ANOVA table is:

	SS	Df	MS	F
Treatment	TrSS	df1	TrMS	F
Error	ESS	df2	EMS	
Total	TSS	df3		

Instructions

The following table shows the steps involved in using the various parts of ANOVA program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	0	
2	Add a new data point.		XEQ A	X?	
	Enter a value for x.	x	R/S	Number of data points	
	Enter the next value for x.		R/S	X?	
	Enter a value for x.	x	R/S	Number of data points	
3	Delete a data point.		XEQ D	X?	
	Enter a value for x to delete.	x	R/S	Number of data points	
	To delete another value of x.		R/S	X?	
4	To view the basic stat registers of the current variable.		XEQ E	sum x	n
			R/S	sdev	mean
5	To begin a data for the next variable.		XEQ B	X?	
	Enter the first value of x in the variable.	x	R/S	Number of data points	
	Resume at step 2 to add more data.				
6	To recall the summation registers of a previously entered variable.		XEQ R	J?	
	Enter the index of the variable. If your input exceeds the current number of variables, the program displays the error message INVALID DATA.	Index of variable	R/S		
	You can add data (see step 2), delete data (see step 3) from the variable you recalled. When you are done with editing the data of the variable you recalled, you can start a new variable (see step 5), recall another variable to edit its data, or proceed to calculate the ANOVA table (see step 7).				

Step #	Step/Substep	Input	Command	Output X	Output Y
7	To calculate and view the elements of the ANOVA table. The program first displays the values for TSS and TrSS.		XEQ C	TrSS	TSS
	View ESS and df1.		R/S	df1	ESS
	View df2and df3.		R/S	df3	df2
	View TrMS and EMS.		R/S	EMS	TrMS
	View the F statistic.		R/S	F	
8	To display the program name. The program pauses to display the program name.		XEQ U	NCS-ANOVA-ONE1	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A	Grand Sum of x		
B	Begin a new variable	B	Grand Sum of x^2		
C	Calculate ANOVA table elements	C	Grand Sum of n(i)		
D	Delete an existing data point	D	TSS		
E	View the basic statistics for the current variable	E	TrSS		
I	Initialize program	F	ESS = TSS - TrSS		
R	Recall summations for a variable	G			
U	Display program name	H	Store copy of register I		
Internally used labels		I	Current variable index		
S	Store summations of current variable	J	Index		
G	Recall a set of summation registers	K	Number of variables k		
		L	df1 = k - 1		
		M	df2 = sum of n(i) - k		
		N	df3 = sum of n(i) - 1		
		O	TrMS = TrSS / df1		
		P	EMS = ESS / df2		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		Q	F statistic		
		R			
		S			
		T			
		U			
		V			
		W			
		X	x		
		Y			
		Z			
		0			
		1	[n, Σx, Σx^2] for variable 1		
		2	[n, Σx, Σx^2] for variable 2		
		…			
		#k	[n, Σx, Σx^2] for variable k		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	ClΣ	Clear stat registers
I003	1	Initialize index to store stat registers
I004	STO I	
I005	STO K	Initialize the number of variables k
I006	RTN	
A001	LBL A	Add a new observation x to stat registers
A002	INPUT X	Prompt for x
A003	RCL X	
A004	ENTER	
A005	Σ+	Add to stat registers
A006	STOP	
A007	GTO A001	
D001	LBL D	Delete observation from stat registers
D002	INPUT X	
D003	RCL X	
D004	ENTER	
D005	Σ-	Remove from stat registers
D006	STOP	
D007	GTO D001	
S001	LBL S	Store summations of current variable
S002	[n, Σx, Σx^2]	Build stat registers to store into 3-D vector
S003	STO (I)	Store in an unnamed variable

Line	Command	Comments
S004	RTN	
B001	LBL B	Begin a new variable
B002	XEQ S	Store the current set of stat registers
B003	1	
B004	STO+ K	Increment the number of variables
B005	RCL K	
B006	STO I	Store current group number in register I
B007	ClΣ	Clear the stat registers
B008	GTO A001	Jump to label A to add the first value in the new variable
R001	LBL R	Recall summations of a variable
R002	XEQ S	Store the current set of stat registers
R003	INPUT J	Prompt for index of variable to recall
R004	RCL K	
R005	x<y?	Validate the input
R006	ACOS(2)	Create a runtime error that displays INVALID DATA
R007	RCL J	Copy validated input to register I. This step will prevent
R008	STO I	the corruption of the value in register I if input is incorrect.
R009	XEQ G	Recall stat registers
R010	RTN	
G001	LBL G	Recall summations registers
G002	-27	Set index to access stat register n
G003	STO J	
G004	RCL (I)	
G005	[1,0,0]	
G006	×	Get the value of stat register n
G007	STO (J)	Store it in stat register block
G008	-28	Set index to access stat register Σx
G009	STO J	
G010	RCL (I)	
G011	[0,1,0]	
G012	×	Get the value of stat register Σx
G013	STO (J)	Store it in stat register block
G014	-30	Set index to access stat register Σx^2
G015	STO J	
G016	RCL (I)	
G017	[0,0,1]	
G018	×	Get the value of stat register Σx^2
G019	STO (J)	Store it in stat register block
G020	RTN	
C001	LBL C	Calculate ANOVA table elements
C002	XEQ S	Save current stat variable
C003	RCL I	
C004	STO H	Copy the value of register I

Line	Command	Comments
C005	0	
C006	STO A	
C007	STO B	
C008	STO C	
C009	STO E	
C010	1	
C011	RCL K	
C012	1E3	
C013	÷	
C014	+	
C015	STO I	
C016	XEQ G	
C017	Σx^2	Update grand sum of x^2
C018	STO+ B	
C019	Σx	
C020	STO+ A	Update grand sum of x
C021	x^2	
C022	n	
C023	STO+ C	Update grand sum of n(i)
C024	÷	
C025	STO+ E	Update TrSS
C026	ISG I	
C027	GTO C016	
C028	RCL H	
C029	STO I	Restore I to keep track of current variable
C030	RCL B	
C031	RCL A	
C032	x^2	
C033	RCL C	
C034	÷	Calculate TSS
C035	STO- E	
C036	-	
C037	STO D	Calculate TrSS
C038	RCL E	
C039	-	
C040	STO F	Calculate ESS
C041	RCL K	
C042	1	
C043	-	
C044	STOL L	Calculate df1
C045	RCL C	
C046	RCL K	
C047	-	
C048	STO M	Calculate df2

Line	Command	Comments
C049	RCL C	
C050	1	
C051	-	
C052	STO N	Calculate df3
C053	RCL E	
C054	RCL L	
C055	÷	
C056	STO O	Calculate TrMS
C057	RCL F	
C058	RCL M	
C059	÷	
C060	STO P	Calculate EMS
C061	÷	
C062	STO Q	Calculate F statistic
C063	RCL D	
C064	RCL E	
C065	SF 10	
C066	TSS,TrSS	This is a “tag” created using EQN button
C067	PSE	Display output tag
C068	STOP	Display TSS and TrSS
C069	RCL F	
C070	RCL L	
C071	ESS,DF1	This is a “tag” created using EQN button
C072	PSE	Display output tag
C073	STOP	Display ESS and df1
C074	RCL M	
C075	RCL N	
C076	DF2,DF3	This is a “tag” created using EQN button
C077	PSE	Display output tag
C078	STOP	Display df2 and df3
C079	RCL O	
C080	RCL P	
C081	TrMS,EMS	This is a “tag” created using EQN button
C082	PSE	Display output tag
C083	STOP	Display TrMS and EMS
C084	RCL Q	Display F statistic
C085	F STAT	This is a “tag” created using EQN button
C086	PSE	Display output tag
C087	CF 10	
C088	RTN	
E001	LBL E	View statistic for current variable
E002	SF 10	
E003	n, Σx	This is a “tag” created using EQN button
E004	PSE	Display output tag

Line	Command	Comments
E005	n	
E006	Σx	
E007	STOP	Display n and Σx
E008	$\bar{x}$, sx	This is a "tag" created using EQN button
E009	PSE	Display output tag
E010	CF 10	
E011	$\bar{x}$	
E012	sx	Display mean and standard deviation
E013	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-ANOVA-ONE1	This is a "tag" created using EQN button
U004	PSE	Display program name
U005	CF 10	
U006	RTN	

Example

Let's look at an example that exercises various routines in the ANOVA program. The next table contains a set of observations. We want to test the hypothesis that the variables X1 through X4 have the same mean at a 95% confidence level (that is, at significance level α = 0.05).

	1	2	3	4	5	6	7
X1	88	99	96	68	85		
X2	78	62	98	83	61	88	
X3	80	61	74	92	78	54	77
X3	71	65	90	46			

After you enter the data point, calculate and display the elements of the ANOVA table.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	88	R/S	1.00
			R/S	X?
		99	R/S	2.00
			R/S	X?
		96	R/S	3.00

Step #	Step/Substep	Input	Command	Output
	Repeat the above steps for the remaining X1 values.			
3	Begin the second variable for variable X2.		XEQ B	X?
	Enter the first value for variable X2.	78	R/S	1.00
			R/S	X?
	Enter the second value for variable X2.	62	R/S	2.00
	Repeat the above substeps to finish enter the data for variable X2. When you are done repeat step 2 to enter the data for variables X3 and X3			
4	Calculate and display the ANOVA table.		XEQ C	4,530.00 930.44 Displays the values for TSS and TrSS
	View ESS and df1.		R/S	3,599.56 3.00 Displays the values for ESS and df1
	View df2 and df3.		R/S	18.00 21.00 Displays the values for df2 and df3
	View TrMS and EMS.		R/S	310.15 199.98 Displays the values for TrMS and EMS
	View the F statistic.		R/S	199.98 1.55 Displays the value for the F statistic

The One-way ANOVA table is:

	SS	df	MS	F
Treatment	930.44	3	310.15	1.55
Error	3599.56	18	199.98	
Total	4530.00	21		

Since the calculated F statistic is 1.55 and does not exceed the value $F_{0.05,3,18}$ = 3.16, we cannot reject, at a 95% confidence level, the null hypotheses stating that the means of variables X1 through X4 have the same statistical mean.

Appendix A contains a new approximation that I recently developed for inverse F function. The appendix also contains tables for selected values.

Two-Way ANOVA Table Program NCS_ANOVA_TWO1

Overview

The NCS_ANOVA_TWO1 program calculates the elements of a two-way ANOVA table with no replications. The program offers the following features:

1. Handle data for row and column treatments. Each type of treatment has one value (that is, there is no replication).
2. Add observations for each row and column treatment. The input proceeds row-wise.
3. Edit observations at specific row and column indices..
4. Calculate and display the elements of the two-way ANOVA table.

While this program is based on similar versions found in the Stat Pacs of the HP-67 and HP-41C, feature number 3 is an important enhancement. The legacy versions of the ANOVA program allow you to delete invalid or erroneous observations right after you made the error. Once you proceed to entering the next values, deleting old erroneous values is not possible! The program NCS_ANOVA_TWO1 stores the sum or rows and columns in un-named memory registers using indirect addressing. After you entered the table of data, you can go back and edit values located at a specific row and column. During the entry of the tabulated data, the program briefly pauses and shows you the row and column indices of the next tabulated value, before proceeding to prompt you to enter that value. Observing the row and column number is very useful if you want to catch an error and modify it later. Record on a piece of paper the erroneous value along with the row and column associated with that value. Later, after you are done entering the tabulated data, you can hunt for the offending values by specifying the row index, column index, and erroneous value. The program will also prompt you for the correct value. Armed with your input, the program then updates the appropriate row and column sums and other summations to reflect the correction you made.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

The row and column sums:

$$\text{Row } RS_i = \sum_{i=1}^{r} x_{ij}$$

$$\text{Column } CS_j = \sum_{j=1}^{c} x_{ij}$$

The sum of squares:

$$TSS = \sum\sum x_{ij}^2 - (\sum\sum x_{ij})^2 / rc$$

$$\text{Row } RSS = \sum_{i=1}^{r} \frac{\left(\sum_{j=1}^{c} x_{ij}\right)^2}{c} - (\sum\sum x_{ij})^2 / rc$$

$$\text{Column } CSS = \sum_{j=1}^{c} \frac{\left(\sum_{i=1}^{r} x_{ij}\right)^2}{r} - (\sum\sum x_{ij})^2 / rc$$

$$\text{Error } ESS = TSS - RSS - CSS$$

The degrees of freedom:

$$\text{Row } df_1 = r - 1$$

$$\text{Column } df_2 = c - 1$$

$$\text{Error } df_3 = (c - 1)(r - 1)$$

The mean sum of squares:

$$RMS = RSS / df_1$$

$$CMS = CSS / df_2$$

$$EMS = ESS / df_3$$

The F statistics:

$$\text{Row } F = RMS / EMS$$

$$\text{Column } F = CMS / EMS$$

The two-way ANOVA table is:

	SS	Df	MS	F
Row	RSS	df1	RMS	Row F
Column	CSS	df2	CMS	Column F
Error	ESS	df3	EMS	
Total	TSS			

You are given the 100(1-α)% confidence level to test the row and column treatment hypotheses. If the row F statistics exceeds the value of $F_{\alpha,df1,df3}$ we cannot reject the hypothesis that the row treatment is significant. Likewise, If the column F statistic exceeds the value of $F_{\alpha,df2,df3}$ we cannot reject the hypothesis that the column treatment is significant

Instructions

The following table shows the steps involved in using the various parts of ANOVA program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	R?	
	Enter the number of rows.	Rows	R/S	C?	
	Enter the number of columns.	Columns	R/S		
2	Enter the values in the tabulated data. The program prompts you to		XEQ A	Column	Row

Step #	Step/Substep	Input	Command	Output X	Output Y
	input the tabulated values across the rows.				
	After a brief pause the program prompts you to input a value.			X?	
	Enter the next value for x.	x	R/S	Next Column	Row
	After a brief pause the program prompts you to input a value.			X?	
	Enter the next value for x.	x	R/S	Next Column	Row
	Repeat the above substeps in this step until you have entered all of values in the table of data.				
3	Delete a data point.		XEQ D	I?	
	Enter the row index.	Row	R/S	J?	
	Enter the column index.	Column	R/S	X?	
	Enter the incorrect value to delete.	Incorrect x	R/S	X?	
	Enter the correct value.	Correct x	R/S		
4	To view the basic stat registers of the current variable.		XEQ E	sum x	n
			R/S	sdev	mean
5	To calculate and view the elements of the ANOVA table.		XEQ C	CSS	RSS
	View RSS and CSS.		R/S	ESS	TSS
	View TSS and ESS.		R/S	df1	df2
	View df1 and df2.		R/S	df3	
	View df3.		R/S	Column F	Row F
	View the row F and the column F statistics.				
6	To display the program name. The program pauses to display the program name.		XEQ U	NCS-ANOVA-TWO1	

Program Resources

Labels	Purpose	Memory Registers	Purpose
A	Enter the tabulated data	A	Grand sum of x
C	Calculate ANOVA	B	Grand sum of x^2
D	Delete erroneous x	C	c = the number of columns
F	Set/Restore display format	D	RSS
I	Initialize program	E	CSS
U	Display program name	F	TSS
		G	ESS = TSS - RSS - CSS
		H	
		I	Index for RSS
		J	Index for CSS
		K	Row df1 = r - 1
		L	Column df2 = c - 1
		M	Error df3 = (c-1)(r-1)
		N	Row F stat
		O	Column F stat
		P	
		Q	
		R	r = the number of rows
		S	
		T	
		U	
		V	
		W	
		X	x
		Y	
		Z	
		0	
		1	RS(1)
		2	RS(2)
		…	
		r	RS(r)
		r+1	CS(1)
		r+2	CS(2)
		..	
		r+c	CS(c)

Program Listing

Line	Command	Comments
F001	LBL F	Set default display format
F002	FIX 5	
F003	RTN	
I001	LBL I	Initialize program
I002	INPUT R	Prompt user for the number of rows
I003	INPUT C	Prompt user for the number of columns
I004	0	Initialize the grand/double summations
I005	STO A	
I006	STO B	
I007	1	
I008	RCL C	
I009	RCL R	
I010	×	
I011	1E3	
I012	÷	
I013	+	
I014	STO I	Prepare index register to loop in order to initialize the unnamed memory registers
I015	1E-499	Push a small value into the stack
I016	STO (I)	Store a very small non-zero value in the unnamed memory registers. These values should not significantly influence the data you enter.
I017	ISG I	
I018	GTO I016	End of loop
I019	RTN	
A001	LBL A	Enter the tabulated data
A002	1	
A003	RCL R	
A004	1E3	
A005	÷	
A006	+	
A007	STO I	Initialize index register I to loop and access memory registers that store row sums
A008	1	Start outer loop to process each row
A009	RCL R	
A010	+	
A011	LASTx	
A012	RCL C	
A013	+	
A014	1E3	
A015	÷	
A016	+	
A017	STO J	Initialize index register J to loop and access memory registers that store

Line	Command	Comments
		column sums
A018	RCL I	Start inner loop to process each column
A019	IP	
A020	RCL J	
A021	IP	
A022	RCL R	
A023	-	
A024	FIX 0	
A025	PSE	Pause to show the row index in Y and column index in X
A026	XEQ F	Restore display format
A027	INPUT X	
A028	STO+ (I)	Update row sum
A029	STO+ (J)	Update column sum
A030	STO+ A	Update grand sum of x
A031	x^2	
A032	STO+ B	Update grand sum of x^2
A033	ISG J	
A034	GTO A018	End of inner loop
A035	ISG I	
A036	GTO A008	End of outer loop
A037	RTN	
D001	LBL D	Delete erroneous x
D002	INPUT I	
D003	INPUT J	
D004	RCL C	
D005	STO+ J	
D006	INPUT X	Prompt user to enter erroneous x
D007	RCL X	
D008	STO- A	Decrement grand sum of x
D009	STO- (I)	Decrement row(I) sum
D010	STO- (J)	Decrement column(J) sum
D011	x^2	
D012	STO- B	Decrement grand sum of x^2
D013	INPUT X	Prompt user to enter correct x
D014	RCL X	
D015	STO+ A	Increment grand sum of x
D016	STO+ (I)	Increment row(I) sum
D017	STO+ (J)	Increment column(J) sum
D018	x^2	
D019	STO+ B	Increment grand sum of x^2
D020	RTN	
C001	LBL C	Calculate and display ANOVA table
C002	0	
C003	STO D	

Line	Command	Comments
C004	STO E	
C005	1	
C006	RCL R	
C007	1E3	
C008	÷	
C009	+	
C010	STO I	Initialize index register I to access the row sums
C011	1	Start outer loop to process each row
C012	RCL R	
C013	+	
C014	RCL R	
C015	RCL C	
C016	+	
C017	1E3	
C018	÷	
C019	+	
C020	STO J	Initialize index register J to loop and access memory registers that store column sums
C021	RCL (I)	
C022	x^2	
C023	STO+ D	
C024	ISG I	
C025	GTO C021	
C026	RCL C	
C027	STO÷ D	Calculate first term of RSS
C028	RCL (J)	
C029	x^2	
C030	STO+ E	
C031	ISG J	
C032	GTO C028	
C033	RCL R	
C034	STO÷ E	Calculate first term of CSS
C035	RCL B	
C036	RCL A	
C037	x^2	
C038	RCL C	
C039	RCL R	
C040	×	
C041	÷	
C042	STO- D	Calculate RSS
C043	STO- E	Calculate CSS
C044	-	
C045	STO F	Calculate TSS

Line	Command	Comments
C046	RCL D	
C047	-	
C048	RCL E	
C049	-	
C050	STO G	Calculate WSS
C051	RCL R	
C052	1	
C053	-	
C054	STO K	Calculate df1
C055	RCL C	
C056	1	
C057	-	
C058	STO L	Calculate df2
C059	×	
C060	STO M	Calculate df3
C061	RCL D	
C062	RCL K	
C063	÷	
C064	RCL G	
C065	RCL M	
C066	÷	
C067	÷	
C068	STO N	Calculate row F
C069	RCL E	
C070	RCL L	
C071	÷	
C072	RCL G	
C073	RCL M	
C074	÷	
C075	÷	
C076	STO O	Calculate column F
C077	RCL D	
C078	SF 10	
C079	RSS,CSS	This is a “tag” created using EQN button
C080	PSE	Display the output tag for RSS and CSS
C081	RCL E	Display RSS and CSS
C082	STOP	
C083	TSS,ESS	This is a “tag” created using EQN button
C084	PSE	Display the output tag for TSS and ESS
C085	RCL F	
C086	RCL G	
C087	STOP	Display TSS and ESS
C088	DF1,DF2	This is a “tag” created using EQN button

Line	Command	Comments
C089	PSE	Display the output tag for df1 and df2
C090	RCL K	
C091	RCL L	Display df1 and df2
C092	STOP	
C093	DF3	This is a “tag” created using EQN button
C094	PSE	Display the output tag for df3
C095	RCL M	Display df3
C096	STOP	
C097	F ROW,F COL	This is a “tag” created using EQN button
C098	PSE	Display the output tag for row F and column F
C099	CF 10	
C100	RCL N	
C101	RCL O	Display row F and column F
C102	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-ANOVA-TWO1	This is a “tag” created using EQN button
U004	PSE	Display program name
U005	CF 10	
U006	RTN	

☞

The label F has the command FIX 5 to reset the display format. If you wish to use a different display format, then edit the command in line F002. The program invokes label F after it briefly switches to a FIX 0 display, in order to show the row and column indices, before it prompts you to enter the value at that these coordinates in the data table.

Example

Let's look at an example that exercises the various routines in the ANOVA program. The next table contains a set of observations. Determine the relevance of the row and column treatments at a 95% confidence level (that is for a significance level $\alpha = 0.05$).

	Column 1	Column 2	Column 3	Column 4
Row 1	7	6	8	7
Row 2	2	4	4	4
Row 3	4	6	5	3

After you enter the data point, calculate and display the elements of the ANOVA table, and then examine the values of the row and column F statistics.

The following table shows the steps performed using the FIX 2 display mode. As mentioned earlier, the program briefly switches to the FIX 0 display when it displays the indices for the row and column entry.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	R?
	Enter the number of rows.	3	R/S	C?
	Enter the number of columns.	4	R/S	1E-499
2	Enter the tabulated data going across each row. The program pauses before each prompt to display the row and column indices for the next value to enter.		XEQ A	1. 1.
				X?
	Enter the first value in the data table, in row 1 and column 1.	7	R/S	1. 2.
	After a brief pause, the program proceeds with prompting you for an input.			X?
	Enter the next value in the data table, in row 1 and column 2.	6	R/S	1. 3.
	After a brief pause, the			X?

Step #	Step/Substep	Input	Command	Output
	program proceeds with prompting you for an input.			
	Enter the next value in the data table, in row 1 and column 3.	8	R/S	1. 4.
	After a brief pause, the program proceeds with prompting you for an input.			X?
	Enter the next value in the data table, in row 1 and column 4.	7	R/S	2. 1.
	After a brief pause, the program proceeds with prompting you for an input.			X?
	Enter the next value in the data table, in row 2 and column 1.	2	R/S	X?
	Repeat the above steps for the remaining tabulated values.			
3	Calculate and display the ANOVA table. The program first displays the values for RSS and CSS.		XEQ C	26.00 3.33 Displays the values for RSS and CSS
	View TSS and ESS.		R/S	36.00 6.67 Displays the values for TSS and ESS
	View df_1 and df_2.		R/S	2.00 3.00 Displays the values for df1 and df2
	View df_3.		R/S	3.00 6.00 Displays the value for df3

Step #	Step/Substep	Input	Command	Output
	View the row F and the column F statistics.		R/S	11.70 1.00 Displays the values for the row F and column F statistics

The two-way ANOVA table is (the values for column MS are calculated separately using the values from columns SS and df):

	SS	df	MS	F
Row	26.00	2	13.00	11.7
Column	3.33	3	1.11	1.00
Error	6.67	6	1.11	
Total	36.00			

The value for the row F statistic is 11.7. Comparing this value with $F_{0.05,2,6}$ which is equal to 5.14 we cannot reject the hypothesis that the row treatment is significant. The value for the column F statistic is 1.00. Compared this value with $F_{0.05,3,6}$ which is equal to 4.75 we cannot accept the hypothesis that the column treatment is significant.

Appendix A contains a new approximation that I recently developed for inverse F function. The appendix also contains tables for selected values.

Two-Way ANOVA Table with Replications Program NCS_ANOVA_TWO2

Overview

The NCS_ANOVA_TWO2 program calculates the elements of a two-way ANOVA table with replications. The program offers the following features:

1. Handle data for row and column treatments. Each type of treatment has multiple values.
2. Add observations for each cell, row, and column treatment. The input proceeds row-wise, cell by cell.
3. Edit observations at specific cell, row, and column indices..
4. View the data matrix.
5. Calculate and display the elements of the two-way ANOVA table.

The program stores the original data, using indirect addressing, in unnamed memory registers. This approach allows you to view and edit your original data.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

The row and column sums:

$$\text{Row } RS_i = \sum_{i=1}^{r} \sum_{k=1}^{n} x_{ijk}$$

$$\text{Column } CS_j = \sum_{j=1}^{c} \sum_{k=1}^{n} x_{ijk}$$

The sum of squares:

$$TSS = \sum_i \sum_j \sum_k x_{ijk}^2 - \left(\sum_i \sum_j \sum_k x_{ijk}\right)^2 / rcn$$

$$\text{Row } RSS = \sum_{i=1}^{r} \frac{\left(\sum_{j=1}^{c} \sum_{k=1}^{n} x_{ijk}\right)^2}{cn} - \left(\sum_i \sum_j \sum_k x_{ijk}\right)^2 / rcn$$

$$\text{Column } CSS = \sum_{j=1}^{c} \frac{\left(\sum_{i=1}^{r} \sum_{k=1}^{n} x_{ijk}\right)^2}{rn} - \left(\sum_i \sum_j \sum_k x_{ijk}\right)^2 / rcn$$

Interaction ISS = TSS – RSS – CSS – ESS

Error ESS = $\sum_i \sum_j \sum_k x_{ijk}^2 - \sum_i \sum_j (\sum_k x_{ijk})^2 / n$

The degrees of freedom:

Row $df_1 = r - 1$

Column $df_2 = c - 1$

Interaction $df_3 = (c - 1)(r - 1)$

Error $df_4 = rc(n - 1)$

The mean sum of squares:

RMS = RSS / df_1

CMS = CSS / df_2

IMS = ISS / df_3

EMS = ESS / df_4

The F statistics:

Fixed model F_1 = RMS / EMS

Mixed model F_{1m} = RMS / IMS

Column F_2 = CMS / EMS

Interactive F_3 = IMS / EMS

The two-way ANOVA table for the fixed row and fixed column effect is:

	SS	Df	MS	F
Row	RSS	df1	RMS	Fixed F_1
Column	CSS	df2	CMS	Column F_2
Interaction	ISS	df3	IMS	Interactive F_3
Error	ESS	df4	EMS	
Total	TSS			

The two-way ANOVA table for the fixed row and random column effect is:

	SS	Df	MS	F
Row	RSS	df1	RMS	Mixed F_{1m}
Column	CSS	df2	CMS	Column F_2
Interaction	ISS	df3	IMS	Interactive F_3
Error	ESS	df4	EMS	
Total	TSS			

The calculated F statistics are them compared with the F statistics in standard tables to determine if the set of null hypotheses that the column effect, row effect, and treatment effect are not significant.

Instructions

The following table shows the steps involved in using the various parts of ANOVA program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	R?	
	Enter the number of rows.	Rows	R/S	C?	
	Enter the number of columns.	Columns	R/S	N?	
	Enter the number of replications in each cell.	Number of replications	R.S		
2	Enter the values in the tabulated data. The program prompts you to input the tabulated values across the rows and cell by cell. The program pauses with the message NEXT CELL before prompting you to enter values for the next cell.		XEQ A	X?	
	Enter the next value for x.	x	R/S	X?	
	Enter the next value for x.	x	R/S	X?	
	Enter the next value for x.	x	R/S	X?	
	Repeat the above substeps in this step until you have entered all of values in the data table.				
3	Edit/View a data point.		XEQ D	I?	
	Enter the row index.	Row	R/S	J?	
	Enter the column index.	Column	R/S	K?	
	Enter the cell index.	Index	R/S	X?	
	Enter the correct value.	x	R/S		
4	To view the data in the matrix x while pausing at each entry.		XEQ V	x	
5	View the data in the matrix x while stopping at		XEQ W	x	

Step #	Step/Substep	Input	Command	Output X	Output Y
	each entry.				
	View the next value.		R/S	x	
	Repeat the above step to view the rest of the values.				
6	Calculate and display the cell sums, row sums, column sums, and the entries in the ANOVA table. The program displays the first cell sum.		XEQ C	cell sum	row + column/100
6.1	View the next cell sum.		R/S	cell sum	row + column/100
6.2	Repeat the above step to view the rest of the cell sums in the same row.		R/S	cell sum	row + column/100
6.3	View the row sum.		R/S	row sum	row index
6.4	View the cell sum in the next row, repeat steps 6.2. To view the row sum for that set of cell sums, perform step6.3. This alternative output pattern continues until you view the cell sums and the row sum of the last row. When you are done the program displays the sum of the first column.		R/S	column sum	column index
	View the sum of the second column.		R/S	column sum	column index
	View the rest of the column sums.		R/S	column sum	column index
	View the values of TSS and RSS.		R/S	RSS	TSS
	View CSS and ISS.		R/S	ISS	CSS
	View ESS and df_1.		R/S	df_1	ESS
	View df_2 and df_3.		R/S	df_3	df_2
	View df_4 and RMS		R/S	RMS	df_4
	View CMS and IMS.		R/S	IMS	CMS
	View EMS and F_1.		R/S	F_1.	EMS
	View F_{1m}.		R/S	F_{1m}	
	View F_2 and F_3.		R/S	F_3	F_2

Step #	Step/Substep	Input	Command	Output X	Output Y
	The program pauses to displays the message END.				
	To display the program name. The program pauses to display the program name.		XEQ U	NCS-ANOVA-TWO2	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add data points	A	TSS	1	Set = Pause, Clear = Stop
C	Calculate ANOVA table elements	B	RSS		
E	View/Edit an observations	C	Number of columns		
W	View all data points in matrix x	D	CSS		
V	View all data points in matrix x without stopping the program	E	ISS		
I	Initialize the program	F	ESS		
U	Display program anme	G	df1		
		H	df2		
		I	Used		
Z	Prepare loop variable to access all of the matrix element	J	Range of columns		
Y	Prepare loop variable to access the column summations	K	df3, Cell index		
		L	df4		
		M	RMS		
		N	Number of replications in each cell		
		O	CMS		
		P	IMS		
		Q	EMS		
		R	Number of rows		
		S	F1 stat		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		T	F1m stat		
		U	F2 stat		
		V	F3 stat, Row sum		
		W	Cell sum, $(\Sigma\Sigma\ x)^2$/rcn		
		X	x,I		
		Y	J		
		Z	K		
		0	r*c*n		
		1	x(1,1,1)		
		2	x(1,1,2)		
		…			
		n	x(1,1,n)		
		n+1	x(1,2,1)		
		n+2	x(1,2,2)		
		…			
		[(I-1)c + (J-1)]n + K	x(I,J,K)		
		…			
		rcn	x(r,c,n)		
		rcn+1	CS(1)		
		rcn+2	CS(2)		
		…			
		rcn+c	CS(c)		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	INPUT R	Prompt the user to enter the number of row
I003	INPUT C	Prompt the user to enter the number of columns
I004	INPUT N	Prompt the user to enter the number of replications
I005	0	
I006	STO I	Set index to store value of r*n*c
I007	RCL R	
I008	RCL C	
I009	×	
I010	RCL N	
I011	×	
I012	STO (I)	Store r*c*n in unnamed memory register 0
I013	RTN	
U001	LBL U	Display program name
U002	SF 10	

Line	Command	Comments
U003	NCS-ANOVA-TWO2	This is a "tag" created using the EQN button
U004	PSE	
U005	CF 10	
U006	RTN	
Y001	LBL Y	Prepare loop variable to access the column summations
Y002	1	The stat register n = r*c*n
Y003	n	
Y004	+	
Y005	LASTx	
Y006	RCL C	
Y007	+	
Y008	1E3	
Y009	÷	
Y010	+	
Y011	STO J	
Y012	RTN	
Z001	LBL Z	Prepare loop variable to access all of the matrix element
Z002	0	
Z003	STO I	Set index to recall value of r*n*c
Z004	1	
Z005	RCL (I)	
Z006	1E3	
Z007	÷	
Z008	+	
Z009	STO I	
Z010	RTN	
A001	LBL A	Add data points, row by row
A002	CF 1	
A003	SF 10	
A004	INPUT ROW WISE	This is a "tag" created using the EQN button
A005	PSE	Display the output tag
A006	PSE	
A007	CELL BY CELL	This is a "tag" created using the EQN button
A008	PSE	Display the output tag
A009	PSE	
A010	CF 10	
A011	XEQ Z	
A012	RCL N	
A013	STO Z	
A014	FS? 1	

Line	Command	Comments
A015	GTO A017	
A016	GTO A022	
A017	SF 10	
A018	NEXT CELL	
A019	PSE	
A020	CF 10	
A021	CF 1	
A022	INPUT X	
A023	RCL X	
A024	STO (I)	
A025	1	
A026	STO- Z	Decrement cell replication counter
A027	RCL Z	
A028	x>0?	
A029	GTO A033	
A030	RCL N	
A031	STO Z	
A032	SF 1	
A033	ISG I	
A034	GTO A014	
A035	CF 1	
A036	RTN	
E001	LBL E	View/Edit an observations
E002	INPUT I	Enter row index
E003	RCL R	
E004	x<y?	Validate index
E005	ACOS(2)	Generate INVALID DATA error message
E006	INPUT J	Enter column index
E007	RCL C	
E008	x<y?	Validate index
E009	ACOS(2)	Generate INVALID DATA error message
E010	INPUT K	Enter cell index
E011	RCL N	
E012	x<y?	Validate index
E013	ACOS(2)	Generate INVALID DATA error message
E014	RCL I	
E015	1	
E016	-	
E017	RCL C	
E018	×	
E019	RCL J	
E020	1	
E021	-	

Line	Command	Comments
E022	+	
E023	RCL N	
E024	×	
E025	RCL K	
E026	+	Calculate the index of unnamed register that stores x(I,J,K) = ((I-1) ×C + (J-1)) ×N + K
E027	STO I	
E028	RCL (I)	
E029	STO X	Get x(I,J,K)
E030	INPUT X	Show user the curent value and promptfor new value
E031	STO (I)	Store the new value of x(I,J,K)
E032	RTN	
V001	LBL V	View all data points in matrix x without stopping the program
V002	SF 1	
V003	XEQ W003	
V004	CF 1	
V005	RTN	
W001	LBL W	View the data matrix, row by row, cell by cell
W002	CF 1	
W003	XEQ Z	
W004	RCL (I)	Start the loop to view the matrix elements
W005	FS? 1	
W006	GTO W009	
W007	STOP	
W008	GTO W010	
W009	PSE	
W010	ISG I	
W011	GTO W004	End of the loop that views the matrix
W012	RTN	
C001	LBL C	Calculate ANOVA table elements
C002	ClΣ	
C003	XEQ Z	
C004	RCL (I)	Start a quick loop to calculate the grand sum of x and grand sum of x^2, and also set stat register n as rcn
C005	ENTER	
C006	Σ+	Add current x(I,J,K) to stat summations
C007	ISG I	
C008	GTO C004	End of loop
C009	XEQ Y	Initialize loop control variable to access column sums. Now stat register n = r×c×n
C010	1E-499	
C011	STO (J)	Start loop to initialize column sums
C012	ISG J	
C013	GTO C011	End of loop to initialize column sums

Line	Command	Comments
C014	SF 10	
C015	0	
C016	STO B	Initialize RSS
C017	STO D	Initialize CSS
C018	STO F	Initialize ESS
C019	1	
C020	STO I	Initialize index to matrix x
C021	STO X	Initialize row counter
C022	0	Start of outer loop
C023	STO V	Initialize row sum for current row
C024	XEQ Y	Initialize control variable J to access column sums
C025	1	Start of intermediate loop
C026	STO Y	Initialize column ccounter
C027	0	
C028	STO W	Initialize cell sum
C029	1	
C030	STO Z	Initialize cell counter
C031	RCL (I)	Start innermost loop
C032	STO+ W	Update cell sum
C033	1	
C034	STO+ I	
C035	STO+ Z	
C036	RCL Z	
C037	RCL N	
C038	x≥y?	
C039	GTO C031	End of innermost loop
C040	RCL X	
C041	RCL Y	
C042	100	
C043	÷	
C044	+	
C045	RCL W	
C046	CELL SUM	This is a "tag" created using the EQN button
C047	PSE	Display output tag
C048	STOP	Display cell sum in X and [row,col] in Y
C049	STO+ (J)	Update column sum
C050	STO+ V	Update row sum for current row
C051	x^2	
C052	STO+ F	Update ESS
C053	1	
C054	STO+ Y	Next column
C055	STO+ J	
C056	RCL Y	
C057	RCL C	

Line	Command	Comments
C058	x≥y?	More columns to process?
C059	GTO C027	End of intermediate loop
C060	ROW NUM,SUM	This is a "tag" created using the EQN button
C061	PSE	Display output tag
C062	RCLX	
C063	RCL V	
C064	STOP	Display row index and row sum
C065	x^2	
C066	STO+ B	Update RSS
C067	1	
C068	STO+ X	Next row
C069	RCL X	
C070	RCL R	
C071	x≥y?	More rows to process?
C072	GTO C022	End of outer loop
C073	XEQ Y	Initialize index to access column sums
C074	1	
C075	STO Y	Reset column index
C076	RCL Y	Start loop to access column sums
C077	RCL (J)	
C078	COL NUM, SUM	This is a "tag" created using the EQN button
C079	PSE	Display output tag
C080	STOP	Display column index and sum of column
C081	x^2	
C082	STO+ D	Update CSS
C083	1	
C084	STO+ Y	Next column
C085	ISG J	
C086	GTO C076	End of loop that accesses column sums
C087	Σx^2	
C088	Σx	
C089	x^2	
C090	n	
C091	÷	
C092	STO W	
C093	-	Calculate TSS
C094	STO A	
C095	STO E	ISS = TSS
C096	RCL B	
C097	RCL N	
C098	÷	

Line	Command	Comments
C099	RCL C	
C100	÷	
C101	RCL W	
C102	-	Calculate RSS
C103	STO B	
C104	STO- E	ISS = TSS - RSS
C105	RCL D	
C106	RCL N	
C107	÷	
C108	RCL R	
C109	÷	
C110	RCL W	
C111	-	Calculate CSS
C112	STO D	
C113	STO- E	ISS = TSS - RSS - CSS
C114	Σx^2	
C115	RCL F	
C116	RCL N	
C117	÷	
C118	-	
C119	STO F	Calculate ESS
C120	STO- E	ISS = TSS - RSS - CSS - ESS
C121	RCL R	
C122	1	
C123	-	Calculate df1
C124	STO G	
C125	RCL C	
C126	1	
C127	-	Calculate df2
C128	STO H	
C129	×	Calculate df3
C130	STO K	
C131	RCL R	
C132	RCL C	
C133	×	
C134	RCL N	
C135	1	
C136	-	
C137	×	Calculate df4
C138	STO L	
C139	RCL B	
C140	RCL G	
C141	÷	Calculate RMS

Line	Command	Comments
C142	STO M	
C143	RCL D	
C144	RCL H	
C145	÷	Calculate CMS
C146	STO O	
C147	RCL E	
C148	RCL K	Calculate IMS
C149	÷	
C150	STO P	
C151	RCL F	
C152	RCL L	
C153	÷	Calculate EMS
C154	STO Q	
C155	RCL M	
C156	RCL Q	
C157	÷	Calculate F1
C158	STO S	
C159	RCL M	
C160	RCL P	
C161	÷	Calculate F1m
C162	STO T	
C163	RCL O	
C164	RCL Q	
C165	÷	Calculate F2
C166	STO U	
C167	RCL P	
C168	RCL Q	
C169	÷	Calculate F3
C170	STO V	
C171	TSS,RSS	This is a "tag" created using the EQN button
C172	PSE	Display output tag
C173	RCL A	
C174	RCL B	
C175	STOP	Display the values of TSS and ESS
C176	CSS,ISS	This is a "tag" created using the EQN button
C177	PSE	Display output tag
C178	RCL D	
C179	RCL E	
C180	STOP	Display the values of CSS and ISS
C181	ESS,DF1	This is a "tag" created using the EQN button
C182	PSE	Display output tag
C183	RCL F	
C184	RCL G	

Line	Command	Comments
C185	STOP	Display the values of ESS and df1
C186	DF2,DF3	This is a "tag" created using the EQN button
C187	PSE	Display output tag
C188	RCL H	
C189	RCL K	
C190	STOP	Display the values of df2 and df3
C191	DF4,RMS	This is a "tag" created using the EQN button
C192	PSE	Display output tag
C193	RCL L	
C194	RCL M	
C195	STOP	Display the values of df4 and RMS
C196	CMS,IMS	This is a "tag" created using the EQN button
C197	PSE	Display output tag
C198	RCL O	
C199	RCL P	
C200	STOP	Display the values of CMS and IMS
C201	EMS,F1	This is a "tag" created using the EQN button
C202	PSE	Display output tag
C203	RCL Q	
C204	RCL S	
C205	STOP	Display the values of EMS and F1
C206	F1m STAT	This is a "tag" created using the EQN button
C207	PSE	Display output tag
C208	RCL T	
C209	STOP	Display the value of F1m
C210	F2,F3	This is a "tag" created using the EQN button
C211	PSE	Display output tag
C212	RCL U	
C213	RCL V	
C214	STOP	Display the values of F2 and F3
C215	END	This is a "tag" created using the EQN button
C216	PSE	Display output tag
C217	CF 10	
C218	RTN	

Example

Let's look at an example that exercises the various routines in the ANOVA program. The next table contains a set of observations. Determine the relevance of the row and column treatments at a 95% confidence level:

	Column 1	Column 2	Column 3
Row 1	4,7,5	2,3,2	5,6,4
Row 2	9,8,8	8,7,5	10,8,7

After you enter the data point, calculate and display the elements of the ANOVA table, and then examine the values of the row and column F statistics.

The following table shows the steps performed using the FIX 2 display mode. As mentioned earlier, the program briefly switches to the FIX 0 display when it displays the indices for the row and column entry.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	R?
	Enter the number of rows.	2	R/S	C?
	Enter the number of columns.	3	R/S	N?
	Enter the number of replications per cell.	3	R/S	
2	Enter the tabulated data going across each row. The program pauses before each prompt to display the row and column indices for the next value to enter.		XEQ A	X?
	Enter the first value in the data table, in row 1 column 1, and cell 1.	4	R/S	X?
	Enter the next value for x.	7		X?
	Enter the next value for x.	5	R/S	X?
	Repeat the above steps to enter the remaining values in the table. The program pauses to display the message			

Step #	Step/Substep	Input	Command	Output
	NEXT CELL before you start entering data for the next cell. This message helps you pace the input.			
3	Calculate and display the cell sums, row sums, column sums, and the entries in the ANOVA table. The program displays the first cell sum.		XEQ C	1.01 16.00 Displays the row/column index and cell sum for cell(1,1)
	View the next cell sum.		R/S	1.02 7.00 Displays the row/column index and cell sum for the cell(1,2)
	View the next cell sum.		R/S	1.03 15.00 Displays the row/column index and cell sum for the cell(1,3)
	View the first row sum.		R/S	1.00 38.00 Display the row index and row sum for row 1
	View the next cell sum.		R/S	2.01 25.00 Displays the row/column index and cell sum for the cell(2,1)
	View the next cell sum.		R/S	2.02 20.00 Displays the row/column index and cell sum for the cell(2,2)
	View the next cell sum.		R/S	2.03 25.00 Displays the row/column

Step #	Step/Substep	Input	Command	Output
				index and cell sum for the cell(2,3)
	View the second row sum		R/S	2.00 70.00 Display the row index and row sum for row 2
	View the sum of column 1.		R/S	1.00 41.00 Display the column index and sum for column 1
	View the sum of column 2.		R/S	2.00 27.00 Display the column index and sum for column 2
	View the sum of column 3.		R/S	3.00 40.00 Display the column index and sum for column 3
	View the values of TSS and RSS		R/S	96.00 56.89
	View CSS and ISS.		R/S	20.33 1.44
	View ESS and df_1.		R/S	17.33 1.00
	View df_2 and df_3.		R/S	2.00 2.00
	View df_4 and RMS		R/S	12.00 56.89
	View CMS and IMS.		R/S	10.17 0.72
	View EMS and F_1.		R/S	1.44 39.38

Step #	Step/Substep	Input	Command	Output
	View F_{1m}.		R/S	39.38 78.77
	View F_2 and F_3		R/S	7.04 0.50
	The program pauses to displays the message END.			

The Two-way ANOVA table for the fixed row and fixed column effects is:

	SS	df	MS	F
Row	56.89	1	56.89	39.38
Column	20.33	2	10.17	7.04
Interaction	1.44	2	0.72	0.50
Error	17.33	12	1.44	
Total	96.00			

The next table summarizes the calculated F statistics and compares them with values found in standard tables. The conclusion draw, at a 95% confidence level, is that we cannot accept the null hypotheses stating that the row and column effects are not significant. By contrast, we cannot reject the null hypothesis stating that the treatment effect is not significant.

	F Calculated	df1	df2	F Tabulated
Row	39.38	1	12	4.74
Column	7.04	2	12	3.88
Interaction	0.50	2	12	3.88

The Two-way ANOVA table for the fixed row and random column effects is:

	SS	df	MS	F
Row	56.89	1	56.89	78.77
Column	20.33	2	10.17	7.04
Interaction	1.44	2	0.72	0.50
Error	17.33	12	1.44	
Total	96.00			

The next table summarizes the calculated F statistics and compares them with values found in standard tables. The conclusion draw, at a 95% confidence level, is that we cannot accept the null hypotheses stating that the row and column effects are not significant. By contrast, we cannot reject the null hypothesis stating that the treatment effect is not significant.

	F Calculated	df1	df2	F Tabulated
Row	78.77	1	12	4.74
Column	7.04	2	12	3.88
Interaction	0.50	2	12	3.88

Appendix A contains a new approximation that I recently developed for inverse F function. The appendix also contains tables for selected values.

Bartlett Chi-Square Program NCS_BARTCHI1

Overview

The NCS_BARTCHI1 program calculates the Bartlett Chi-square statistic for an array of sample variances. The program offers the following operations:

1. Add sample variances.
2. Delete sample variances.
3. Edit sample variances.
4. Swap sample variances.
5. View specific sample variances.
6. View the values of the paired rank array.
7. Calculate the Bartlett Chi-square statistic and other related statistics.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Given an array of n sample variances v_i and their associated degrees of freedom f_i, the sum of degrees of freedom and weighted variance are:

$$f = \sum_{i=1}^{n} f_i$$

$$v = \frac{\sum_{i=1}^{n} f_i \, v_i}{f}$$

The Bartlett Chi-square statistic is obtained using:

$$\chi^2 = \frac{f \ln(v) - \sum_{i=1}^{n} f_i \ln(v_i)}{1 + \frac{\left(\sum_{i=1}^{n} \frac{1}{f_i} - \frac{1}{f}\right)}{3(n-1)}}$$

To test the null hypothesis:

H_0: The array of sample variances that are estimates for the population variance σ^2.

If the calculated Chi-square does not exceed $\chi 2_{\alpha,df}$ then we cannot reject the null hypothesis which states that the array of sample variances are estimates for the population variance σ^2.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	0	
2	Add a pair of rank values.		XEQ A	X?	
	Enter a value for variance.	v(i)	R/S	Y?	
	Enter a value for degrees of freedom.	f(i)	R/S	Number of data points	
	Enter the next sample variances. Repeat this step until you have entered all of the sample variances and their degrees of freedom.	v(i)	R/S	X?	
3	Edit a sample's variance and frequency.		XEQ E	I?	
	Enter the index for the value.	Index	R/S	X?	
	View the value and optionally enter a replacement value for the variance.	v(i)	R/S	Y?	
	View the value and optionally enter a replacement value for the degrees of freedom.	f(i)	R/S	[v(i),f(i)]	
4	View the data while pausing at each value. The program displays the values in the table by columns.		XEQ V	I?	
	Enter the index for the	Index	R/S	[v(i),f(i)]	

Step #	Step/Substep	Input	Command	Output X	Output Y
	value.				
5	View the data while pausing at each value. The program displays the values in the table by columns.		XEQ W	[v(i),f(i)]	
6	Calculate the Bartlett Chi-square statistic and related statistics. The program first displays the value for f and v.		XEQ C	v	f
	View the degrees of freedom and the Bartlett Chi-square statistic.		R/S	Degrees of freedom	Chi-square
7	Display the program name. The program pauses to display the program name.		XEQ U	NCS-BARTCHI1	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	C	Count for number of data points	1	Index invalid when set
D	Delete an existing data point	F	f = sum of degrees of freedom	2	Project a new x onto y when set
E	Edit a data point	V	v = sum of variances		
V	View a data point	T	Sum of 1/f(i)		
W	View all data points	U	Sum of f(i) * ln(v(i))		
I	Initialize	I	Index, used		
S	Swap two data points	J	Index, used		
C	Calculate statistics	X	v(i)		
U	Display program name	Y	F(i)		
Labels for internal use		Z	Chi-square		
H	Validate indices and display error message if an index is invalid	1	Data point #1		
G	Prompt for x and y values	2	Data point #2		
R	Display the message INVALID DATA	3	Data point #3		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		…			
		#C	Data point #C		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	

Line	Command	Comments
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO– C	
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	Validate index
S006	RCL (I)	Recall data points and place them on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x⇆y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	FS? 1	Index out or range?
V005	GTO R001	
V006	RCL (I)	Recall data point
V007	STOP	
V008	1	
V009	RCL I	
V010	+	
V011	STO I	
V012	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	

Line	Command	Comments
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data pointd
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate the rank statistics
C002	0	
C003	STO F	
C004	STO V	
C005	STO T	
C006	STO U	
C007	1	
C008	RCL C	
C009	1E3	
C010	÷	
C011	+	
C012	STO I	Initialize loop variable
C013	RCL (I)	
C014	[1,0]	
C015	×	
C016	STO X	Get the variance
C017	RCL (I)	
C018	[0,1]	
C019	×	
C020	STO Y	Get the degrees of freedom
C021	STO+ F	
C022	×	
C023	STO+ V	
C024	RCL Y	
C025	1/x	
C026	STO+ T	
C027	RCL X	
C028	LN	
C029	RCL Y	
C030	×	
C031	STO+ U	
C032	ISG I	
C033	GTO C013	
C034	RCL T	
C035	RCL F	

Line	Command	Comments
C036	STO÷ V	Calculate variance v
C037	1/x	
C038	-	
C039	3	
C040	÷	
C041	RCL C	
C042	1	
C043	-	
C044	÷	
C045	1	
C046	+	
C047	1/x	
C048	RCL V	
C049	LN	
C050	RCL F	
C051	×	
C052	RCL U	
C053	-	
C054	×	
C055	STO Z	
C056	SF 10	
C057	F,V	This is a “tag” created using EQN button
C058	PSE	Display output tag
C059	RCL F	
C060	RCL V	
C061	STOP	Display f and v
C062	DF,CHI	This is a “tag” created using EQN button
C063	PSE	Display output tag
C064	CF 10	
C065	RCL C	
C066	1	
C067	-	
C068	RCL Z	
C069	RTN	Display the degrees of freedom and Chi-square
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT Y	Prompt user for y value
G004	[X,Y]	Create data point. Enter this statement using the EQN key.
G005	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	Enter this statement using the EQN key.
R004	RTN	
U001	LBL U	Display program name

Line	Command	Comments
U002	SF 10	
U003	NCS-BARTCHI1	This is a “tag” created using EQN button
U004	PSE	Display the program name
U005	CF 10	
U006	RTN	

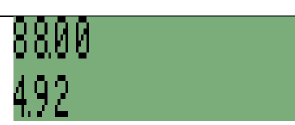
Example

Let’s look at an example that exercises various routines in the NCS_BARTCHI1 program. Here is a table that shows the variances on the density of a polymer, taken from different batches. The table also shows the associated degrees of freedom for each batch:

Batch Number	Sample Variance	Degrees of Fredom
1	5.5	10
2	5.1	20
3	5.2	17
4	4.7	18
5	4.8	8
6	4.3	15

After you enter the above data, calculate the f and v statistics, degrees of freedom, and the Bartlett Chi-square statistic. Use the latter statistic to test the null hypothesis at a 95% confidence level.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Enter the data.		XEQ A	X?
	Enter the first value of the sample variance.	5.5	R/S	Y?
	Enter the first value of associated degrees of freedom.	10	R/S	1.00
	Repeat the above steps to enter the rest of the tabulated values for the sample variances and degrees of freedom.			
3	Calculate f and v statistics, degrees of freedom, and the Bartlett Chi-square statistic. The program		XEQ C	 Displays the f and v

Step #	Step/Substep	Input	Command	Output
	displays the values for the f and v statistics.			statistics
	To view the degrees of freedom, and the Bartlett Chi-square statistic.		R/S	5.00 0.25 Displays the degrees of freedom and the Bartlett Chi-square statistic

Since the calculated χ^2 value of 0.25 does not exceed the value of $\chi^2_{5,0.05}$ (equal to 11.070) we cannot reject the null hypothesis, stating that the sample variances are estimates for the population variance σ^2.

Appendix A contains a new approximation that I recently developed for inverse χ^2 function. The appendix also contains tables for selected values. Using the approximation in Appendix A, the estimated value of $\chi^2_{5,0.05}$ comes out to be 11.35, which has a 2.52% error.

Behrens-Fisher Test Program NCS_BEHRENS1

Overview

The NCS_BEHRENS1 program implements the Behrens-Fisher test for paired and unpaired arrays. The program allows you to enter multiple groups of data and then select any two groups to test whether or not they have the same statistical means. This feature allows you to compare the means of multiple data groups without rekeying in the data for each comparison.

The program does not store the raw data for each set. Doing so and offering the expected features of adding, deleting, and deleting data points can be taxing on the calculator resources. Instead, the program supports the following operations:

- Adding and deleting data within the currently edited group.
- Ending the work on a data group and starting a new data group. The program can support as many groups as allowed by the number of unnamed memory registers. The program stores the number of data points, the sum of observations, and the sum of observations squared for each data group using a 3-D vector.
- Recalling a previously stored data group to add or delete observations. This feature gives flexibility to the user by not *sealing* one data group once you move on to another data group.
- Selecting any two data groups to calculate Behrens-Fisher statistic that is used to test the null hypothesis.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Given data groups x and y of equal or unequal number of data elements. These data groups may or may not have the same average and standard deviation. It is the latter difference that sets the Behrens-Fisher test apart from the paired and unpaired Student-t tests. These tests assume that the standard deviations of the tested data are statistically the same. The null hypothesis states that the difference between the means of the two data groups is the value D:

$H_0: \mu_x - \mu_y - D = 0$

The program calculates the Behrens-Fisher statistics using the following equation:

$$d = (\bar{x} - \bar{y} - D) / \sqrt{(s_x^2/n_x + s_y^2/n_y)}$$

$$\theta = \tan^{-1}[(s_x / s_y) \sqrt{(n_y / n_x)}]$$

Given the values for d, θ, n_x, n_y, and the 100(1- α)% confidence level, you need to look in special tables for the Behrens-Fisher statistics to determine if the values allow to either not reject or not accept the null hypothesis.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	0	
2	Add a new data point.		XEQ A	X?	
	Enter a value for x.	x	R/S	Number of data points	
	Enter another value.		R/S	X?	
	Resume with the steps above.				
3	Delete a data point.		XEQ D	X?	
	Enter a value for x.	x	R/S	Number of data points	
4	End the data entry for the current group.		XEQ E	[n, Σx, Σx^2]	
	Resume with step 2 to enter new data.				
5	Recall summations of previous data group.		XEQ R	I?	
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S		
	You can resume at step 2 to add more data and/or step 3 to delete data. When you are done, execute step				

Step #	Step/Substep	Input	Command	Output X	Output Y
	4.				
6	To calculate the Student-t and the degrees of freedom.		XEQ C	I?	
	Enter the index of the one group. If you enter an invalid index (greater than the number of actual sets), the program displays the INVALID DATA error message. Entering an index of 0 or less has unpredictable effects!	Index	R/S	J?	
	Enter the index of the other group. If you enter an invalid index (equal to the index for the other group, or greater than the number of actual sets), the program displays the INAVLID DATA error message. Entering an index of 0 or less has unpredictable effects!	Index	R/S	D?	
	Enter the value for the tested mean difference D.	D	R/S	d statistic	Angle theta
	View the mean and standard deviation of the first group.		R/S	mean value	sdev value
	View the mean and standard deviation of the second group.		R/S	mean value	sdev value
7	Display the program name. The program pauses to display the program name.		XEQ U	NCS-BEHRENS1	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a data point to the current group	A	n1	1	Set = recall group mode
C	Calculate Student-t and degrees of freedom for 2 groups	B	Mean of x		
D	Delete a data point from current group	C	Sdev of x		
E	End the data entry for the current group	D	difference between mean D		
I	Initialize the program	E	n2		
R	Recall summations of previous data group	F	Mean of y		
U	Display the program name	G	Sdev of y		
		H			
Internally used labels		I	Index of stat data for x		
P	Parse data recalled from unnamed variables	J	Index of stat data for y		
		K			
		L			
		M			
		N	Number of data groups		
		O			
		P			
		Q			
		R			
		S			
		T	Angle		
		U			
		V			
		W			
		X			
		Y	Used		
		Z	Used		
		1	[n, Σx, Σx^2] for group 1		
		2	[n, Σx, Σx^2] for group 2		
		…			
		n	[n, Σx, Σx^2] for group n		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	ClΣ	
I003	0	
I004	STO N	Initialize data group counter
I005	CF 1	Clear recall data group mode
I006	RTN	
A001	LBL A	Add data point to the current data group
A002	INPUT X	
A003	RCL X	
A004	Σ+	
A005	STOP	
A006	GTO A001	
D001	LBL D	Delete a data point from the current data group
D002	INPUT X	
D003	RCL X	
D004	Σ-	
D005	RTN	
E001	LBL E	End the data entry for the current data group
E002	FS? 1	Is recalled data group mode on?
E003	GTO E008	
E004	1	
E005	STO+ N	
E006	RCL N	
E007	STO I	
E008	[n, Σx, Σx^2]	
E009	STO (I)	
E010	ClΣ	Clear summation registers block
E011	CF 1	Clear recalled data group mode
E012	RTN	
P001	LBL P	Parse data recalled from unnamed variables
P002	STO Z	
P003	[0,0,1]	
P004	×	
P005	RCL Z	
P006	[0,1,0]	
P007	×	
P008	RCL Z	
P009	[1,0,0]	
P010	×	
P011	RTN	

Line	Command	Comments
R001	LBL R	Recall an existing data group
R002	FS? 1	Already editing another recalled group?
R003	GTO R006	
R004	XEQ E	
R005	GTO R009	
R006	[n, Σx, Σ x^2]	
R007	STO (I)	
R008	ClΣ	
R009	INPUT I	
R010	SF 1	Set group recall mode
R011	RCL (I)	
R012	XEQ P	Parse recalled data group
R013	-27	
R014	STO J	
R015	R↓	
R016	STO (J)	Store n
R017	R↓	
R018	1	
R019	STO- J	
R020	R↓	
R021	STO (J)	Store Σx
R022	R↓	
R023	2	
R024	STO- J	
R025	R↓	
R026	STO (J)	Store Σx^2
R027	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-BEHRENS1	This is a "tag" created using the EQN button
U004	PSE	Display tag
U005	CF 10	
U006	RTN	
C001	LBL C	Calculate statistics
C002	INPUT I	Prompt user for the index of the first group
C003	RCL I	
C004	RCL N	
C005	x<y?	Is the index I out of range?
C006	ASIN(2)	Use EQN button to enter this expression
C007	INPUT J	Prompt user for the index of the second group
C008	RCL J	
C009	RCL N	
C010	x<y?	Is the index J out of range?

Line	Command	Comments
C011	ASIN(2)	Use EQN button to enter this expression
C012	RCL I	
C013	RCL J	
C014	x=y?	Indices I and J equal?
C015	ASIN(2)	Use EQN button to enter this expression
C016	INPUT D	Prompt user to enter the value for difference D
C017	RCL (I)	Recall the summation data of the first group
C018	XEQ P	Parse the value into separate sums
C019	STO A	Store nx
C020	÷	
C021	STO B	Store mean of x
C022	x^2	
C023	RCL A	
C024	×	
C025	-	
C026	RCL A	
C027	1	
C028	-	
C029	÷	
C030	√x	
C031	STO C	Store sdev of x
C032	RCL (J)	Recall the summation data of the second group
C033	XEQ P	Parse the value into separate sums
C034	STO E	Store ny
C035	÷	
C036	STO F	Store mean of y
C037	x^2	
C038	RCL E	
C039	×	
C040	-	
C041	RCL E	
C042	1	
C043	-	
C044	÷	
C045	√x	
C046	STO G	Store sdev of y
C047	RCL B	
C048	RCL F	
C049	-	
C050	RCL D	
C051	-	
C052	RCL C	
C053	x^2	
C054	RCL A	

Line	Command	Comments
C055	÷	
C056	RCL G	
C057	x^2	
C058	RCL E	
C059	÷	
C060	+	
C061	√x	
C062	÷	Calculate d statistic
C063	STO Z	
C064	RCL C	
C065	RCL G	
C066	÷	
C067	RCL E	
C068	RCL A	
C069	÷	
C070	√x	
C071	×	
C072	ATAN	Calculate the angle theta
C073	STO T	
C074	SF 10	
C075	ANGLE,D	This is a “tag” created using the EQN button
C076	PSE	Display output tag
C077	RCL Z	
C078	STOP	
C079	X DEV,MEAN	This is a “tag” created using the EQN button
C080	PSE	Display output tag
C081	RCL C	
C082	RCL B	
C083	STOP	
C084	Y DEV,MEAN	This is a “tag” created using the EQN button
C085	PSE	Display output tag
C086	CF 10	
C087	RCL G	
C088	RCL F	
C089	RTN	

Example

Let’s look at an example that exercises various routines in the NCS_BEHRENS1 program. The next table contains data points that we want to test for equal statistical means (that is, d = 0).

x	y
79	91
84	103

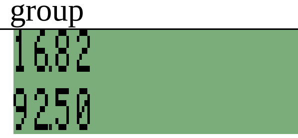
108	90
114	113
120	108
103	87
122	100
120	80
	99
	84

After you enter the above data, perform the following steps:

1. Calculate degrees of freedom and Student-t statistic.
2. Test the null hypothesis at a 95% confidence level.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Add the first data point in the first data group.		XEQ A	X?
	Enter a value for x.	79	R/S	1.00
	Repeat the above steps for the rest of the values in the first data groups.			
3	End the first data group.		XEQ E	
4	Add the first data point in the second data group.		XEQ A	X?
	Enter a value for x.	91	R/S	1.00
	Repeat the above steps for the rest of the values in the second data groups.			

Step #	Step/Substep	Input	Command	Output
5	End the second data group.		XEQ E	
6	Calculate statistical results.		XEQ C	I?
	Enter the index for group 1.	1	R/S	J?
	Enter the index for group 2.	2	R/S	D?
	Enter the difference of 0.	0	R/S	 Displays the values for the Student-t and the degrees of freedom
	View the standard deviation and the mean for the first data group.		R/S	 Displays the standard deviation and the mean for the first data group
	View the standard deviation and the mean for the first data group.		R/S	 Displays the standard deviation and the mean for the first data group

Best Linear Regression Program NCS_BEST_LR1

Overview

This section introduces you to the best fit program and looks at program rationale and features.

Quick Introduction

The NCS_BEST_LR1 program implements automatic best linear regression and offers the following features:

1. Storing data points once and reusing them for different regression scenarios. The program stores the (x, y) data points as 2-D vectors, a new feature of the HP-35s. This storage scheme reduces the number of registers used by half.
2. The ability to delete, edit, view, and swap data points. Thus the regression program incorporates basic operations to manage the data points.
3. The ability to select separate sets of transformation for the x and for the y observations.

HP's Legacy with Automatic Best Fit Programs

The HP-35s, like many HP calculators, has built-in linear regression. This built-in feature makes it very easy to calculate the slope and intercept for the best straight line that passes though a set of (x, y) data points. A linear regression is the simplest case of regression analysis since it assumes a linear relationship between the independent variable x and the dependent variable y. Not all variables in science, finance, and other disciplines relate to each other in a linear manner. However, by applying mathematical transformations on either or both x and y variables it is possible to linearize the fit. HP has implemented on other calculator models automatic best fit by using the logarithm function to transform data, Along with linear values, HP has offers fitting the models shown in the next table.

Model Name	Equation
Linear	y = a + b x
Logarithmic	y = a + b ln(x)
Exponential	ln(y) = a + b x
Power	ln(y) = a + b ln(x)

By applying the combination of two transformation (linear and logarithmic), you get a set of four models. The NCS_BEST_LR1 program offers up to ten transformations for each of the variables x and y. These transformations give you up to a hundred different curves to test with your data.

Design Features

The design of an automatic best fit program involves several choices:

1. The set of mathematical transformations

2. Whether to hard code the set of mathematical transformations into the program, or offer the user a choice. This aspect is crucial because the presence of zeros and negative data eliminates the use of mathematical transformations like square roots, logarithms, and reciprocals.
3. Whether or not to allow the dynamic shifting and scaling of values during the regression phase.
4. The number of best curve fits to display.

Transformations Table

Looking at design choices number 1 and 2, in the last subsection, I present the following table that contains the set of mathematical transformations:

Transformation Index	Transformation Function f(x)
0	ln(x)
1	x
2	x^2
3	x^3
4	1/x
5	√x
6	1/x^2
7	1/√x
8	Custom 1: current default is $\sqrt[3]{x}$
9	Custom 2: current default is $1/\sqrt[3]{x}$

The above table shows a set of transformation and their associated indices. The table shows an expanded set of mathematical transformations, which the program can apply to determine the best curve fit. Using the indices, the program user can select which set of transformations to apply to each variable. The program implements the set of transformations as an integer made up of the single digits in the above table. I will call this number the *transformation group value*. To mimic HP's classical automatic best fit we use the transformation group value of 10 with each variable. Here the number 10 is made up of the digit 1 and 0 which represent the indices for linear and logarithm in the above table, respectively. The program initializes the transformation group for x and y using the value 4210 for each variable. This value is made up of the digits 4, 2, 1, and 0 which represent the reciprocal, square, linear, and logarithmic transformations, respectively. You can enter these digits in any order, but you must observe the following little rule. The digit 0 must appear to the right of any other digit. So the transformation group values of 4210, 2104, 4120, 4021, 4012 all perform the same set of transformations, although in different order. The order of applying a transformation is from least significant digit to most significant digit. Notice that the number 0421 will not perform the logarithmic transformation, because the digit 0 in this case is not significant. Why not? The program, working with copies of the transformation group values, performs the following steps to manage the transformations:

1. Extracts the least significant digit and uses that digit to determine the next mathematical transformation
2. Reduces the transformation group values by 10 and keeps the integer part

3. Repeats steps 1 and 2 until the transformation group value is zero. This is why the digit 0 must be significant. Putting it at the leftmost place will cause the program to deduce that there are no more transformation indices to process.

The program design allows you to use separate transformation group values for the x and y variables. For example you can use the transformation group value of 3210 for x and just the value of 10 for y. This flexibility allows you to avoid certain transformations for either variable but not the other.

The program handles transformations number 8 and 9 in a bit different manner. The program design uses the labels X and Y as a special placeholder where you can code in your particular transformations and apply them to the variables x and/or y. The label T is the routine that performs the mathematical transformations and calls the code in labels X and Y when the transformation index is 8 or 9, respectively. By setting two separate labels, the novice and average users should find it easier to edit code in isolated and dedicated labels. A good programmer can certainly edit the code in label T itself. However, she or he should observe how the code flows and how it goes through the sequence of comparing the transformation index with different integers.

Shifting and Scaling

Regarding the subject of dynamic shifting and scaling of variables, I decided that such operations may work very well with specific transformations while at the same time might backfire with other transformations. The suitability of scaling and shifting greatly depends on the nature of the data entering the regression. The labels X and Y should be suitable venues that allow you to implement custom scaling and shifting and incorporate them with specific mathematical transformation. For example, you can code label X to get $\ln(x_0 - x)$ where x_0 is a fixed value, either hard coded or recall from a memory register. Another example is to code label Y to apply the transformation $\sqrt{(x + x_0)}$ which can deal with all negative values of x, if x_0 is large enough. This approach allows you to implement scaling and shifting for specific transformation and avoid applying them across the board with all of the transformation. You can certainly apply scaling and shifting to you raw data points separately *before* you key them in the program, if you deem such mathematical operations as appropriate for all the transformations. One example is shifting your data points to eliminate negative values and zeros. Such an operation allows you to apply any or all of the mathematical transformations supported by the automatic best fit program.

And the Winner is?

What about the results of the best curve fit? Simply presenting the very best curve fit for a large combination of curves, which can reach a hundred, may seem too limited. On the other extreme, displaying the results for all the curves, sorted using the values of the coefficient of determination, r^2, is perhaps too demanding for the resources of the HP-35s. The current program design displays the top three best fits.

Instructions

The following table shows the steps involved in using the various parts of the linear regression program. The commands show the label truncated to the first letter when step 001 is implied.

Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the transformation group values. YOU MUST PERFORM THIS STEP AT LEAST ONCE.		XEQ K		
2	Initialize the program.		XEQ I	0	
3	Add a new data point.		XEQ A	X?	
	Enter a value for x.	x	R/S	Y?	
	Enter a value for y.	y	R/S	Number of data points	
4	Delete a data point.		XEQ D	I?	
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message INVALID DATA.	Index of data point to delete	R/S	Number of data points	
5	Edit a data point.		XEQ E	X?	
	Enter a new value for x.	x	R/S	Y?	
	Enter a new value for y.	y	R/S	I?	
	Enter the index of the data point to edit. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S		
6	View a data point.		XEQ V	I?	
	Enter the index of the data point to view. If you enter an invalid	Index	R/S	Data point	Index

Step #	Step/Substep	Input	Command	Output X	Output Y
	index the program displays the message INVALID DATA.				
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	Data point	Index
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1	
7	View all data points (Note the program displays the results using the PSE statement.) If you spot an error in the data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	Data point	Index
8	Swap data points.		XEQ S	I?	
	Enter the index of the first data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the first data point	R/S	J?	
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S		
9	To view or assign new transformation group values.		XEQ J	X? <Tx group value>	
	Optionally enter a new	Optional	R/S	Y?	

Step #	Step/Substep	Input	Command	Output X	Output Y
	value for transforming x	new value		<Ty group value>	
	Optionally enter a new value for transforming y	Optional new value	R/S		
10	Calculate the best regression fits.[1]		XEQ C		
	View the transformation indices for the best regression fit. Tx is the transformation index for x. Ty is the transformation index for y.[2]			[r^2, Tx, Ty]	
	View best regression fit.		R/S	[slope, intercept]	[r^2, Tx, Ty]
	View the transformation indices for the 2nd best regression fit.[2]			[r^2, Tx, Ty]	
	View the 2nd best regression fit.		R/S	[slope, intercept]	[r^2, Tx, Ty]
	View the transformation indices for the 2nd best regression fit.[2]			[r^2, Tx, Ty]	
	View the 3rd best regression fit.		R/S	[slope, intercept]	[r^2, Tx, Ty]
11	To re-examine the best fit results.		XEQ F		
	View the transformation indices for the best regression fit. [2]			[r^2, Tx, Ty]	
	View best regression fit.		R/S	[slope, intercept]	[r^2, Tx, Ty]
	View the transformation indices for the 2nd best regression fit. [2]			[r^2, Tx, Ty]	
	View the 2nd best regression fit.		R/S	[slope, intercept]	[r^2, Tx, Ty]
	View the transformation indices for the 3rd best regression fit. [2]			[r^2, Tx, Ty]	
	View the 3rd best regression fit.		R/S	[slope, intercept]	[r^2, Tx, Ty]
12	To display the program		XEQ U	NCS-BEST-	

Step #	Step/Substep	Input	Command	Output X	Output Y
	name. The program pauses to display the program name.			LR1	

[1] While executing label C, the program switches to the FIX 0 display mode and briefly pauses to show the transformation indices. The program performs this kind of pause twice. The first pause occurs during the model iterations, so you can follow the progress of the program. The second pause occurs when the program displays the final results.

[2] The display is in the FIX 0 mode which will round the value of r^2 to the nearest integer (1 or 0). The second and third digits in the 3-D vector are the relevant part of the output since they represent the transformation indices. Using FIX 0 reduces the distraction from all the superfluous zeros and makes the integer values of the transformation indices stand out and easy to read.

Comments on Runtime Errors

The program validates the indices that you enter to manage data points to prevent the corruption of the data points. Aside from this feature, you are responsible to selecting transformations that that will avoid runtime errors, if you have negative values and zeros. The best fit program is a convenient tool that cycles through different regression curves and find the best three. The application is not a magical data mining tool. The best way to use the program is to know your data!

Program Resources

Labels	Purpose	Memory Registers	Purpose
A	Add a new data point	A	
D	Delete an existing data point	B	
E	Edit a data point	C	Data counter
V	View a data point	D	Ty Group
W	View all data points	E	Tx Group Initial
I	Initialize data points	F	Tx Group
J	View/Set transformation groups	G	Tx
K	Initialize/Reset transformation groups	H	Ty
S	Swap two data points	I	Index
C	Calculate best regression fit	J	Index
F	View best three curve fits	K	[r^2,Tx,Ty] for best fit
Z	Set/reset display format	L	[m, b] for best
X	Custom Transform #1	M	[r^2,Tx,Ty] for 2nd best fit
Y	Custom Transform #2	N	[m, b] for 2nd best fit
U	Display program name	O	[r^2,Tx,Ty] for 3rd best fit
Labels for internal use		P	[m, b] for 3rd best fit

Labels	Purpose	Memory Registers	Purpose
H	Validate indices and set flag 1 if index if invalid	Q	
G	Prompt for X and Y	R	r^2
T	Transform data	S	
P	Parse the value of the transformation index group	T	
B	Find the best three curve fits	U	
		V	
Flag 1	Index invalid when set	W	
		X	Used
		Y	Used
		Z	
		0	[Tx Group, Ty Group]
		1	Data point 1
		2	Data point 2
		…	
		#Count	Data point #Count

Customizing the Program

The program lends itself to several aspects of customization. The next sections discuss these labels.

Labels X and Y

Labels X and Y hold the code for the two custom mathematical transformations. The code in label T invokes labels X and Y. By using two dedicated labels, the user can easily edit the code in these labels to customize the transformations. Each label should be coded such that it gets a value from stack register X and returned the transformed value in that same register. You can use the memory registers that are not used by the program to store intermediate results.

The default coding for labels X and Y is:

```
X001    LBL X
X002    3
X003    1/x
X004    y^x
X005    RTN
Y001    LBL Y
Y002    3
Y003    1/x
Y004    y^x
Y005    1/x
```

Y006 RTN

The above labels support the transformations that return the cube root and its reciprocal.

Label Z

Label Z contains a single command—one that sets the “default” display format. Currently, label Z sets the display format to FIX 5 (in line Z002 of the listing). You can edit this command to offer your preferred display format. The program uses label Z to switch between FIX 0, needed to efficiently display the transformation indices, and the “default” display format. Here is the current implementation of label Z:

Z001 LBL Z
Z002 FIX 5
Z003 RTN

Label K

Label K sets the default transformation group numbers. This value is currently set to 4210 (in line K004 of the listing). You can easily edit this value to set different initial default transformation group numbers. You can also invoke XEQ J to dynamically change the transformation group numbers. These numbers will remain in effect until you clear the machine, clear the memory, or execute either command XEQ K or XEQ J. Here is the current implementation of label K:

K001 LBL K
K002 0
K003 STO I
K004 4210
K005 [REGX,REGX]
K006 STO (I)
K007 RTN

Program Listing

Line	Command	Comment
Z001	LBL Z	Set/restore display mode
Z002	FIX 5	
Z003	RTN	
X001	LBL X	Label to contain the custom transformation #1
X002	3	
X003	1/x	
X004	y^x	
X005	RTN	
Y001	LBL Y	Label to contain the custom transformation #2
Y002	3	
Y003	1/x	

Line	Command	Comment
Y004	y^x	
Y005	1/x	
Y006	RTN	
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO- C	

Line	Command	Comment
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	
S006	RCL (I)	Recall data points on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x↔y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	RCL (I)	Recall data point
V005	STOP	
V006	1	
V007	RCL I	
V008	+	
V009	STO I	
V010	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points
W009	RCL (I)	
W010	PSE	

Line	Command	Comment
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT Y	Prompt user for y value
G004	[X,Y]	Create data point
G005	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	
R004	RTN	
T001	LBL T	Transform data. Works for x and y data.
T002	x≠0?	
T003	GTO T007	
T004	x↔y	
T005	LN	ln(x) transformation
T006	RTN	
T007	1	
T008	x≠y?	Transformation index is not 1?
T009	GTO T012	
T010	REGZ	No transformation
T011	RTN	
T012	1	
T013	+	
T014	x≠y?	Transformation index is not 2?
T015	GTO T019	
T016	REGZ	
T017	x^2	x^2 transformation
T018	RTN	
T019	1	
T020	+	
T021	x≠y?	Transformation index is not 3?
T022	GTO T027	
T023	REGZ	
T024	3	
T025	y^x	x^3 transformation
T026	RTN	
T027	1	
T028	+	
T029	x≠y?	Transformation index is not 4?
T030	GTO T034	
T031	REGZ	

Line	Command	Comment
T032	1/x	1/x transformation
T033	RTN	
T034	1	
T035	+	
T036	x≠y?	Transformation index is not 5?
T037	GTO T041	
T038	REGZ	
T039	√x	√x transformation
T040	RTN	
T041	1	
T042	+	
T043	x≠y?	Transformation index is not 6?
T044	GTO T049	
T045	REGZ	
T046	x^2	
T047	1/x	1/ x^2 transformation
T048	RTN	
T049	1	
T050	+	
T051	x≠y?	Transformation index is not 7?
T052	GTO T057	
T053	REGZ	
T054	√x	
T055	1/x	1/√x transformation
T056	RTN	
T057	1	
T058	+	
T059	x≠y?	Transformation index is not 8?
T060	GTO T064	
T061	REGZ	
T062	XEQ X	Invoke first custom transformation
T063	RTN	
T064	REGZ	
T065	XEQ Y	Invoke second custom transformation
T066	RTN	
P001	LBL P	Parse the transformation index group
P002	10	
P003	÷	
P004	ENTER	
P005	IP	
P006	x↔y	
P007	FP	
P008	10	

Line	Command	Comment
P009	×	
P010	IP	Update the transformation index group in Y stack register
P011	RTN	Index for next transformation in X stack register
B001	LBL B	Find Best fit
B002	RCL O	
B003	[1,0,0]	
B004	×	Get 3rd best r^2
B005	RCL R	Get current r^2
B006	x<x?	Current r^2 is less than 3rd best r^2?
B007	RTN	Exit
B008	RCL M	
B009	[1,0,0]	
B010	×	
B011	RCL R	Get 2nd best r^2
B012	x>y?	Current r^2 is better than 2nd best r^2?
B013	GTO B019	
B014	[R,G,H]	Found new 3rd best r^2
B015	STO O	
B016	[m,b]	
B017	STO P	
B018	RTN	Exit
B019	RCL K	
B020	[1,0,0]	
B021	×	Get best r^2
B022	RCL R	
B023	x>y?	Is the current r^2 better than best r^2?
B024	GTO B032	
B025	[R,G,H]	Found new 2nd best r^2
B026	x↔ M	Copy old 2nd best to 3rd best
B027	x↔ O	
B028	[m,b]	
B029	x↔ N	
B030	x↔ P	
B031	RTN	Exit
B032	[R,G,H]	Found new best r^2
B033	x↔ K	Move old 2nd best to 3rd best fit
B034	x↔ M	
B035	x↔ O	
B036	[m,b]	Move old best to 2nd best fit
B037	x↔ L	
B038	x↔ N	
B039	x↔ P	Store new best fit
B040	RTN	Exit

Line	Command	Comment
K001	LBL K	Initialize/Reset transformation groups
K002	0	
K003	STO I	
K004	4210	
K005	[REGX,REGX]	
K006	STO (I)	
K007	RTN	
J001	LBL J	View/Select transformation groups
J002	0	
J003	STO I	
J004	RCL (I)	
J005	[1,0]	
J006	×	
J007	STO X	
J008	RCL (I)	
J009	[0,1]	
J010	×	
J011	STO Y	
J012	INPUT X	Prompt for transformation group for x
J013	INPUT Y	Prompt for transformation group for y
J014	[X,Y]	
J015	STO (I)	Update transformation groups
J016	RTN	
C001	LBL C	Perform the linearized regression and find the best fit
C002	3	
C003	RCL C	Not enough data?
C004	x<y?	
C005	GTO R001	
C006	[0,0]	Initialize best r^2 registers
C007	STO L	
C008	STO N	
C009	STO P	
C010	2	
C011	[-1,0,0]	
C012	STO K	Store -1 in best r^2
C013	×	
C014	STO M	Store -2 in best r^2
C015	LASTx	
C016	+	
C017	STO O	Store -3 in best r^2
C018	0	
C019	STO I	
C020	RCL (I)	Recall master value for transformation groups
C021	[1,0]	

Line	Command	Comment
C022	×	
C023	STO E	Store Tx group value
C024	RCL (I)	
C025	[0,1]	
C026	×	
C027	STO D	Store Ty group value
C028	RCL D	Outer loop to transform y -------------------
C029	XEQ P	Parse the Ty group value
C030	STO H	Store the next Ty value
C031	x↔y	
C032	STO D	Store the reduced value of the Ty group
C033	RCL E	
C034	STO F	Copy initial Tx group value
C035	RCL F	Intermediate loop to transform x ----------------
C036	XEQ P	Parse the Tx group value
C037	STO G	Store the next Tx value
C038	x↔y	
C039	STO F	Store the reduced value of the Tx group
C040	1	
C041	RCL C	
C042	1E3	
C043	÷	
C044	+	
C045	STO I	Initialize loop to process data points
C046	ClΣ	
C047	FIX 0	
C048	[G,H]	
C049	PSE	
C050	XEQ Z	Restore display format
C051	RCL (I)	Innermost loop --------------------------------------
C052	[1,0]	
C053	×	
C054	RCL G	Get current Tx value
C055	XEQ T	Transform x
C056	STO X	
C057	RCL (I)	
C058	[0,1]	
C059	×	
C060	RCL H	Get current Ty value
C061	XEQ T	Transform y
C062	STO Y	
C063	RCL X	
C064	Σ+	

Line	Command	Comment
C065	ISG I	
C066	GTO C051	End of innermost loop - - - - - - - - - - - - - - - -
C067	r	
C068	x^2	
C069	STO R	Store r^2. You can insert a R/S or PSE command before the next line to stop or pause the program in order to examine the value of r^2.
C070	XEQ B	Rank r^2 and regression results among the best
C071	RCL F	Next Tx?
C072	x≠0?	
C073	GTO C035	End of intermediate loop - - - - - - - - - - - - - - - -
C074	RCL D	
C075	x≠0?	
C076	GTO C028	End of outer loop -
F001	LBL F	View best three fits
F002	SF 10	
F003	BEST FIT	This is a “tag” created using EQN button
F004	PSE	Display output tag
F005	FIX 0	
F006	RCL K	Show transformation indices
F007	PSE	
F008	XEQ Z	Restore display format
F009	RCL L	Best fit
F010	STOP	
F011	2 ND BEST	This is a “tag” created using EQN button
F012	PSE	Display output tag
F013	FIX 0	
F014	RCL M	Show transformation indices
F015	PSE	
F016	XEQ Z	Restore display format
F017	RCL N	Best 2nd fit
F018	STOP	
F019	3RD BEST	This is a “tag” created using EQN button
F020	PSE	Display output tag
F021	CF 10	
F022	FIX 0	
F023	RCL O	Show transformation indices
F024	PSE	
F025	XEQ Z	Restore display format
F026	RCL P	Best 3rd fit
F027	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-BEST-LR1	This is a “tag” created using EQN button

Line	Command	Comment
U004	PSE	Display program name
U005	CF 10	
U006	RTN	

Example

Let's look at an example that exercises various routines in the regression program. The next Table contains (x, y) data points that we want to fit with the best model.

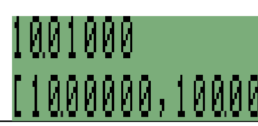
x	y
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

The above data represents points for $y=x^2$. This is an example where two models will provide the best fit—the quadratic fit (y vs. x^2) and the power fit (ln(y) vs. ln(x)).

Perform the following steps:

1. Initialize the transformation groups numbers
2. Use the transformation group values of 210 for both variables to determine the best fit.
3. Reset the data point counters
4. Enter the data points shown in the above table.
5. Calculate and display the best fits.

The following table shows the steps using the FIX 5 display mode as set by label Z.

Step #	Step/Substep	Input	Command	Output
1	Initialize the transformation group numbers.		XEQ K	
2	Set the transformation groups.		XEQ J	X?
	Enter the new transformation group for variable x.	210	R/S	Y?
	Enter the new transformation group for variable y.	210	R/S	
3	Initialize the program.		XEQ I	0
4	Add the first data point.		XEQ A	X?
	Enter a value for x.	1	R/S	Y?
	Enter a value for y.	1	R/S	1
5	Add the second data point.		XEQ A	X?
	Enter a value for x.	2	R/S	Y?
	Enter a value for y.	4	R/S	2
6	Repeat step 3 for all other data point.			10.00000
7	View all data points (Note the program displays the results using the PSE statement). If you spot an error use the command XEQ E to overwrite		XEQ W	

Step #	Step/Substep	Input	Command	Output
	the erroneous data point with the correct one.			
8	Calculate the regression coefficients.		XEQ C	[1., 2., 1.]
				[1.00000,20000… [1.00000,0.00000
	Best fit is y= 0 + 1 * x^2 with r^2 = 1.		R/S	[1., 0., 0.]
				[1.00000,0.0000… [2.00000,9.37100
	Second best curve fit is: ln(y) = 9.371E-11 + 2 ln(x) with r^2 = 1.		R/S	[1., 1., 1.]
				[0.94976,1.0000… [11.00000,⁻22.00
	Third best curve fit is: y = -22 + 11 x with r^2 = 0.94876.			

The above example provides the following best curve fits:

- The best curve fit is y = 0 + x^2 with r^2 = 1
- The second best fit is ln(y) = 9.371E-11 + 2 ln(x) with r^2 = 1. This is considered an alternate best fit. This case is rather unique.
- The third best fit is y = -22 + 11 x with r^2 =0.94876.

Contingency Table Program NCS_CONTABLE1

Overview

The NCS_CONTABLE1 program implements n × k contingency table calculations. The program stores the input data, column sums, row sums in the unnamed memory registers. Since there are quite a few unnamed registers, you can store contingency table with a good number rows and columns.

The program offers the following operations:

1. Initializing the program. This step includes specifying the number of rows and column for the contingency table.
2. Adding entries for the contingency table.
3. Editing entries in the contingency table by specifying the column and row indices for the element to edit.
4. Calculating the row sums, column sums, grand sum, chi-Square and the Pearson contingency coefficient.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Given an n × k table:

		j				Totals
		1	2	…	k	
i						
	1	x_{11}	x_{12}	…	x_{1k}	R_1
	2	x_{21}	x_{22}	…	x_{2k}	R_2
	…	…	…	…	…	
	n	x_{n1}	x_{n2}	…	x_{nk}	R_n
Totals		C_1	C_2	…	C_k	T

The row, column, and grand sums are given by:

Row sum $R_i = \sum_{j=1}^{k} x_{ij}$ for i = 1 to n

Column sum $C_j = \sum_{i=1}^{n} x_{ij}$ for j = 1 to k

Total sum $T = \sum_{i=1}^{n} \sum_{j=1}^{k} x_{ij}$ for i = 1 to n and j = 1 to k

The chi-square and the Pearson contingency coefficient are calculated using the following equations:

$$\chi^2 = T\left(\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{x_{ij}}{R_j\,C_i}\right) - T$$

$$C_c = \sqrt{\frac{\chi 2}{\chi 2\ +\ T}}$$

If the calculated chi-Square does not exceeds the chi-Square at (n-1)(k-1) degrees of freedom and α significance level, then we cannot reject the null hypothesis that says there is no difference in row and column treatments.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	K?	
	Enter the number of columns.	k	R/S	N?	
	Enter the number of rows.	n	R/S	1E-499	
2	Add data points.		XEQ A	X?	
	Enter a value for x.	x	R/S	X?	
	Enter the next value. The sequence of input is by columns. Repeat this step until you have entered all of the elements in the data table.	x	R/S	X?	

Step #	Step/Substep	Input	Command	Output X	Output Y
3	View/Edit a data point.		XEQ E	I?	
	Enter the row index.	Index	R/S	J?	
	Enter the column index.	Index	R/S	X?	
	View the value and optionally enter a replacement value for x.	x	R/S	x	
4	View the data while pausing at each value. The program displays the values in the table by columns.		XEQ V	x	
5	View the data while stopping at each value. The program displays the values in the table by columns.		XEQ W	x(1,1)	
	To view the next value in the matrix x.		R/S	X(2,1)	
	Repeat this step for the rest of the elements in matrix x.		R/S	X(I,J)	
6	Calculate the chi-square and the contingency coefficient		XEQ C	Cc	chi-square
	To view the row sums. Repeat this step for each row sum. The program pauses to display a ROW SUM tag before showing each row sum.		R/S	Row sum	
	To view the columns sums. Repeat this step for each row sum. The program pauses to display a COL SUM tag before showing each column sum.		R/S	Column sum	
	To view the grand sum		R/S	T	
7	Display the program name. The program pauses to display the program name.		XEQ U	NCS-CONTABLE1	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add data points	A	1+k×n	1	Set = Pause, Clear = Stop
C	Calculate chi square and contingency coefficient	B	n+k×n		
E	View/Edit an observations	C	Index of column sums		
W	View all data points in matrix x	D			
V	View all data points in matrix x without stopping the program	E			
I	Initialize the program	F			
U	Display program name	G			
		H			
Z	Display INVALID DATA	I	Index to access x(I,J)		
S	Set index to access column sums	J	Index to access row or column sums		
R	Set index to access row sums	K	Number of columns		
X	Set index to access x(I,J)	L	Column counter		
		M			
		N	Number of rows		
		O	Row counter		
		P			
		Q			
		R	Index of row sums		
		S	chi-Square		
		T	Grand sum of T		
		U			
		V			
		W			
		X	x(I,J)		
		Y			
		Z			
		1	x(1,1)		
		2	x(2,1)		
		…			
		n	x(n,1)		
		n+1	x(1,2)		
		n+2	x(2,2)		
		…			

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		2n	x(n,2)		
		…			
		I + (J-1) ×n	x(I,J)		
		…			
		k×n	x(n,k)		
		1+k×n	Row Sum(1)		
		2+k×n	Row Sum(2)		
		…			
		n+k×n	Row Sum(n)		
		1+n+k×n	Column Sum (1)		
		2+n+k×n	Column Sum (2)		
		…			
		k+n+k×n	Column Sum (k)		

The above table shows that the program stores the matrix x in the unnamed registers as a series of chained columns. That is why the input of the elements in matrix x also follows a column-wise order.

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	INPUT K	Prompt for the number of columns
I003	INPUT N	Prompt for the number of rows
I004	RCL K	
I005	RCL N	
I006	×	
I007	STO A	Store n×k
I008	RCL N	
I009	+	
I010	STOB	Store n×k+n
I011	RCL K	
I012	+	
I013	STO I	Store k+n+k×n which is the index of the last unnamed register to be used
I014	1E-499	Push a small number in the stack
I015	STO (I)	
I016	ISG I	
I017	GTO I015	End of loop to initialize the unnamed registers
I018	RTN	
Z001	LBL Z	Display error message
Z002	ACOS(2)	
Z003	RTN	

Line	Command	Comments
X001	LBL X	Set index to access x(I,J)
X002	1	
X003	RCL A	
X004	1E3	
X005	÷	
X006	+	
X007	STO I	
X008	RTN	
R001	LBL R	Set index to access row sums
R002	RCL A	
R003	1	
R004	+	
R005	RCL B	
R006	1E3	
R007	÷	
R008	+	
R009	STO R	Store index to access row sums
R010	RTN	
S001	LBL S	Set index to access column sums
S002	RCL B	
S003	1	
S004	+	
S005	RCL B	
S006	RCL K	
S007	+	
S008	1E3	
S009	÷	
S010	+	
S011	STO C	Store index to access column sums
S012	RTN	
A001	LBL A	Add data points
A002	XEQ X	Set index to access x(I,J) by column
A003	INPUT X	Start input loop
A004	STO(I)	
A005	ISG I	
A006	GTO A003	End of input loop
A007	RTN	
E001	LBL E	View/Edit an observations
E002	INPUT I	Enter index I
E003	RCL I	Validate index I
E004	RCL N	
E005	x<y?	
E006	GTO Z001	Display error message
E007	INPUT J	Enter index J

Line	Command	Comments
E008	RCL J	
E009	RCL K	
E010	x<y?	
E011	GTO Z001	Display error message
E012	RCL J	Calculate index for x(I,J)
E013	1	
E014	-	
E015	RCL N	
E016	×	
E017	RCL I	
E018	+	
E019	STO I	
E020	RCL (I)	Recall current x(I,J) value
E021	STO X	
E022	INPUT X	Prompt user for new value
E023	STO (I)	Store new value for x(I,J)
E024	RTN	
V001	LBL V	View all data points in matrix x without stopping the program
V002	SF 1	Set flag 2 to make the program pause when showing x(I,J)
V003	XEQ W	
V004	CF 1	
V005	RTN	
W001	LBL W	View all data points in matrix x
W002	XEQ X	Set index to access data points
W003	RCL (I)	Start loop to display data
W004	FS? 1	
W005	GTO W008	
W006	STOP	
W007	GTO W009	
W008	PSE	
W009	ISG I	
W010	GTO W003	
W011	RTN	
C001	LBL C	Calculate chi square and contingency coefficient
C002	XEQ X	Set index to access x(I,J) elements
C003	XEQ S	Set index to access column sums
C004	XEQ R	Set index to access row sums
C005	RCL B	
C006	RCL K	
C007	+	
C008	1E3	
C009	÷	
C010	RCL A	
C011	1	

Line	Command	Comments
C012	+	
C013	+	
C014	STO J	
C015	1E-499	
C016	STO (J)	
C017	ISG J	
C018	GTO C016	
C019	1	
C020	STO L	Initialize column counter
C021	STO O	Initialize row counter
C022	0	
C023	STO T	Initialize the grand sum of x
C024	RCL(I)	Start main loop to iterate over each x(I,J)
C025	STO+ T	
C026	RCL C	Store column sum index in J
C027	STO J	
C028	R↓	
C029	STO+ (J)	Update column sum
C030	RCL R	
C031	STO J	Store row sum in J
C032	R↓	
C033	STO+ (J)	Update row sum
C034	1	
C035	STO+ R	Increment row index
C036	STO+ O	
C037	RCL N	
C038	RCL O	
C039	x<=y?	
C040	GTO C045	
C041	1	
C042	STO+ C	
C043	STO O	Reset row counter
C044	XEQ R	Set index for row sums
C045	ISG I	
C046	GTO C024	End of main loop
C047	0	
C048	STO S	Initialize the double summation used to calculate chi-Square
C049	XEQ X	Set index to access x(I,J)
C050	XEQ R	Set index to access row sums
C051	XEQ S	Set index to access column sums
C052	1	
C053	STO L	Initialize column counter
C054	STO O	Initialize row counter
C055	RCL (I)	Start main loop

Line	Command	Comments
C056	x^2	
C057	RCL R	Get row sum
C058	STO J	
C059	R↓	
C060	RCL (J)	
C061	÷	
C062	RCL C	Get column sum
C063	STO J	
C064	R↓	
C065	RCL (J)	
C066	÷	
C067	STO+ S	Update double summation
C068	1	
C069	STO+ R	Update index for column sum
C070	STO+ O	
C071	RCL N	
C072	RCL O	
C073	x<=y?	
C074	GTO C079	
C075	1	
C076	STO+ C	Update index for row sums
C077	STO O	Reset column counter
C078	XEQ R	Reset index for column sum
C079	ISG I	
C080	GTO C055	End of main loop
C081	RCL S	
C082	1	
C083	-	
C084	RCL T	
C085	×	
C086	STO S	Calculate and store chi-Square
C087	RCL S	
C088	RCL T	
C089	RCL S	
C090	+	
C091	÷	
C092	√x	Calculate contingency coefficient
C093	SF 10	
C094	CHI,CC	This is a "tag" created using the EQN button
C095	PSE	Display output tag
C096	STOP	Display Cc and chi-Square
C097	XEQ R	Set index for row sums
C098	STO I	
C099	RCL(I)	Start loop to display row sums

Line	Command	Comments
C100	ROW SUM	This is a "tag" created using the EQN button
C101	PSE	Display output tag
C102	STOP	Display row sum
C103	ISG I	
C104	GTO C099	End of loop to view row sums
C105	XEQ S	Set index for column sums
C106	STO I	
C107	RCL(I)	Start loop to display column sums
C108	COL SUM	This is a "tag" created using the EQN button
C109	PSE	Display output tag
C110	STOP	Display column sum
C111	ISG I	
C112	GTO C107	End of loop to view column sums
C113	RCL T	Recall grand sum of X
C114	GRAND SUM	This is a "tag" created using the EQN button
C115	PSE	Display output tag
C116	CF 10	
C117	RTN	Display grand sum of X
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-CONTABLE1	This is a "tag" created using the EQN button
U004	PSE	
U005	CF 10	
U006	RTN	

Example

Let's look at an example that exercises various routines in the NCS_CONTABLE1 program. The next table contains the data.

	1	2	3	4
1	36	67	49	56
2	31	60	49	54
3	58	87	80	68

After you enter the above data, perform the following steps:

1. Calculate and display the chi-square and contingency coefficient.
2. Display the row sums.
3. Display the column sums.
4. Display the grand sum.
5. Determine if the row and column treatment are significant at a 95% confidence level.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	K?
	Enter the number of columns.	4	R/S	N?
	Enter the number of rows.	3	R/S	1E-499
2	Enter the data.		XEQ A	X?
	Enter x(1,1)	36	R/S	X?
	Enter x(2,1)	31	R/S	X?
	Enter x(3,1)	58	R/S	X?
	Enter x(1,2)	67	R/S	X?
	Enter x(2,2)	60	R/S	X?
	Enter x(3,2)	87	R/S	X?
	Enter x(1,3)	49	R/S	X?
	Enter x(2,3)	49	R/S	X?
	Enter x(3,3)	80	R/S	X?
	Enter x(1,4)	56	R/S	X?
	Enter x(2,4)	54	R/S	X?
	Enter x(3,4)	68	R/S	
3	Calculate chi-square, and summations. The program first displays the chi-square and the contingency coefficient.		XEQ C	 Displays the chi-square and the contingency coefficient

Step #	Step/Substep	Input	Command	Output
	View the row sums.		R/S	13.02 210.00 Displays row sum 1
			R/S	210.00 194.00 Displays row sum 2
			R/S	194.00 293.00 Displays row sum 3
	View the column sums.		R/S	16.02 125.00 Displays column sum 1
			R/S	125.00 214.00 Displays column sum 2
			R/S	214.00 178.00 Displays column sum 3
			R/S	178.00 180.00 Displays column sum 4
	View the grand sum.		R/S	180.00 697.00 Displays the grand sum

Since the calculated chi-square value of 3.37 does not exceed the value of $\chi^2_{6,0.05}$ (equal to 12.59) we cannot reject the null hypothesis stating that there is no difference in row and column treatments.

Appendix A contains a new approximation that I recently developed for inverse χ^2 function. The appendix also contains tables for selected values.

Nonlinear Regression Program NCS_CRESFIT1

Overview

The NCS_CRESFIT1 program implements nonlinear regression and offers the following features:

1. Storing data points once and reusing them for different regression scenarios. The program stores the (x, y) data points as 2-D vectors, a new feature of the HP-35s, thus reducing the number of unnamed memory registers used by half.
2. The ability to delete, edit, view, and swap data points. Thus the regression program incorporates basic operations to manage the data points.
3. Transforming the x and y observations. You can edit the program to implement your custom transformations and then recalculate the regression statistics without having to reenter the data points.
4. Calculating the nonlinear parameter, as well as the regression intercept, slope, and coefficient of determination.
5. Calculating projected value of y for existing or new values of x.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

The program performs nonlinear regression on the following asymptotic exponential model:

$$y = A + B\ e^{-kx}$$

The above model has the coefficients A and B that can be calculated using linear regression for a given value of the nonlinear parameter k.

The crescent model is a special case of the asymptotic exponential model and has the following form:

$$y = y_0\ (1 - e^{-kx})$$

Comparing the two models we find $A = y_0$ and $B = -y_0$. Since the slope and intercept of the linearized model are related to y0, you can calculate y_0 as an average:

$$y_0 = (A - B) / 2$$

The less error you have in the data the closer are the values of A and B in magnitude.
The program linearizes the asymptotic exponential model by supplying values for the nonlinear parameter k and then calculating the coefficients A and B using the calculator's built-in linear regression.

Working with the nonlinear parameter k involves supplying an initial guess for k, a step search value for changing k, and a minimum step size. The program uses the minimum step size as a criterion to stop the iteration. The iterative process aims to find the value of k that maximizes the value of the coefficient of determination, r^2, for the linearized model.

Here is the algorithm used to refine the guess for the nonlinear parameter k. The linear search algorithm assumes that you start with the values for K, S, and M that represent the nonlinear parameter k, search step size, and minimum step size, respectively.

1. Let the search direction variable C = 1
2. Let RSqrBest = RSqr for K
3. Let Kbest = K
4. Let K0 = K + C*S
5. Calculate RSqr for K0
6. If RSqr > RSqrBest then
    a. Let RSqrBest = RSqr
    b. Let Kbest = K0
    c. Let K = K0
    d. Let InterceptBest = linear regression intercept
    e. Let SlopeBest = linear regression slope
7. Else
    a. Let C = -1*C to reverse the search direction
    b. Let K0 = K + C*S
    c. Calculate RSqr for K0
    d. If RSqr > RSqrBest then
        i. Let RSqrBest = RSqr
        ii. Let Kbest = K0
        iii. Let K = K0
        iv. Let InterceptBest = linear regression intercept
        v. Let SlopeBest = linear regression slope
    e. Else
        i. Let S = S / 10 to reduce the search step size
        ii. If S < M then go to step 9
8. Go to step 4
9. Results are RSqrBest, Kbest, InterceptBest, and SlopeBest.

There is a special code customization section in this document that presents examples of three-parameter nonlinear models that can be adapted to work with this program.

Instructions

The following table shows the steps involved in using the various parts of the linear regression program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	0	
2	Add a new data point.		XEQ A	X?	
	Enter a value for x.	x	R/S	Y?	
	Enter a value for y.	y	R/S	Number of data points	
3	Delete a data point.		XEQ D	I?	
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message INVALID DATA.	Index of data point to delete	R/S	Number of data points	
4	Edit a data point.		XEQ E	I?	
	Enter the index of the data point to edit. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	X?	
	Enter a new value for x.	x	R/S	Y?	
	Enter a new value for y.	y	R/S		
5	View a data point.		XEQ V	I?	
	Enter the index of the data point to	Index	R/S	Data point	Index

Step #	Step/Substep	Input	Command	Output X	Output Y
	view. If you enter an invalid index the program displays the message INVALID DATA.				
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	Data point	Index
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1	
6	View all data points (Note the program displays the results using the PSE statement.) Note: If you spot an error in the data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	Data point	Index
7	Swap data points.		XEQ S	I?	
	Enter the index of the first data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the first data point	R/S	J?	
	Enter the index of	Index of the	R/S		

Step #	Step/Substep	Input	Command	Output X	Output Y
	the second data point. If you enter an invalid index the program displays the message INVALID DATA.	second data point			
8	Calculate the regression coefficients.		XEQ C	K?	
	Enter the initial guess for the nonlinear coefficient k.	K	R/S	S?	
	Enter the initial search step size.	Step size	R/S	M?	
	Enter the minimum search step size. Note: The program pauses to display all values of K used in label B (by viewing variable T) and the pauses to display the best guesses for the nonlinear parameter k (by viewing the variable K).	Minimum step size.	R/S	k	r^2
	View the intercept and slope.		R/S	Slope	Intercept
9	Project x onto y.		XEQ P	X?	
	Enter the value for x.	x	R/S	$\hat{y}$	x
10	To display the program name. The program pauses to display the program name.		XEQ U	NCS-CRESFIT1	

Comments on Runtime Errors

The program validates the indices that you enter to manage data points to prevent the corruption of the data points. Aside from this feature, you are responsible to enter data that will avoid runtime errors. For example, if you are using the square root or logarithms to transform the x and/or y data, then entering data points with negative values will raise runtime errors. Likewise, using reciprocals and logarithms will generate runtime errors for data that are zeros. Entering redundant data points also has its peril of giving you meaningless results. Remember the old saying about program input and output data "Junk in, junk out."

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A	Best Intercept	1	Used
D	Delete an existing data point	B	Best Slope		
E	Edit a data point	C	Data count, n		
V	View a data point	D			
W	View all data points	E			
I	Initialize	F			
S	Swap two data points	G			
C	Calculate regression coefficients	H			
U	Display name of program	I			
Z	Review regression results	J			
	Internally Used Labels	K	K		
B	Calculate R2 for a given value of K	L	K0		
		M	Minimum step size		
		N			
		O			
		P			
		Q	Kbest		
		R	RSqrBest		
		S	Step size for K		
		T	K used to calculate r^2		
		U			
		V			
		W			
		X	x		
		Y	y		
		Z	Step sign		
		1	Data point 1		
		2	Data point 2		

		…			
		n	Data point n		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration

Line	Command	Comments
D020	1	Decrement data point counter
D021	STO– C	
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	Validate index
S006	RCL (I)	Recall data points and place them on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x↔y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	FS? 1	Index out or range?
V005	GTO R001	
V006	RCL (I)	Recall data point
V007	STOP	
V008	1	
V009	RCL I	
V010	+	
V011	STO I	
V012	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	

Line	Command	Comments
W007	STO I	Store loop index
W008	RCL I	Start loop to view the data points
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate coefficients for nonlinear regression
C002	1	
C003	STO Z	Store sign of search
C004	INPUT K	Input non-linear coefficient
C005	INPUT S	Input step size
C006	INPUT M	Input minimum step size
C007	RCL K	
C008	XEQ B	Calculate R2
C009	STO R	Store as R2Best
C010	RCL K	
C011	STO Q	Kbest = K
C012	RCL K	
C013	RCL S	
C014	RCL Z	Step = sign * Step
C015	×	
C016	+	
C017	STO L	K0 = K + Step
C018	XEQ B	Calculate R2
C019	RCL R	Get R2Best
C020	x↔y	
C021	x<y?	R2 not better than R2Best?
C022	GTO C034	
C023	STO R	R2Best = R2
C024	RCL L	
C025	STO K	K = K0
C026	VIEW K	
C027	PSE	Pause to view the new Kbest
C028	STO Q	Kbest = K0
C029	b	
C030	STO A	Store best intercept
C031	m	
C032	STO B	Store best slope
C033	GTO C012	Resume to the top of the loop
C034	-1	
C035	STO× Z	Change sign for step
C036	RCL K	
C037	RCL S	

Line	Command	Comments
C038	RCL Z	
C039	×	Step = sign * Step
C040	+	
C041	STO L	K0 = K + Step
C042	XEQ B	Calculate R2
C043	RCL R	Get R2Best
C044	x↔y	
C045	x<y?	R2 not better than R2Best?
C046	GTO C023	
C047	10	Step = Step/ 10
C048	STO÷ S	
C049	RCL S	
C050	RCL M	
C051	x>y?	
C052	GTO Z001	View results.
C053	GTO C012	Resume at the top of the loop
Z001	LBL Z	
Z002	SF 10	
Z003	R2 Best, K	This is an output tag created using the EQB button
Z004	PSE	Display output tag
Z005	RCL R	
Z006	RCL Q	
Z007	STOP	Display R2best and Kbest
Z008	INTRC,SLOPE	This is an output tag created using the EQB button
Z009	PSE	
Z010	CF 10	
Z011	RCL A	Display output tag
Z012	RCL B	
Z013	RTN	Display intercept and slope
B001	LBL B	Calculate R2 for a given value of K
B002	STO T	Store copy of working K value
B003	VIEW T	
B004	PSE	Pause to view K
B005	ClΣ	
B006	1	
B007	RCL C	
B008	1E3	
B009	÷	
B010	+	
B011	STO I	
B012	RCL (I)	Start the loop
B013	[0,1]	
B014	×	

Line	Command	Comments
B015	STO Y	Get y
B016	RCL (I)	
B017	[1,0]	
B018	×	
B019	STO X	Get x
B020	RCL T	Start calculating exponential term
B021	×	
B022	+/-	
B023	e^x	
B024	Σ+	
B025	ISG I	
B026	GTO B012	End the loop
B027	r	
B028	x^2	
B029	RTN	
P001	LBL P	Project x onto y
P002	INPUT X	
P003	RCL X	
P004	RCL Q	
P005	×	
P006	e^x	
P007	1/x	
P008	RCL B	
P009	×	
P010	RCL A	
P011	+	
P012	STOP	
P013	GTO P001	
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT Y	Prompt user for y value
G004	[X,Y]	Create data point. Enter this statement using the EQN key.
G005	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	Enter this statement using the EQN key.
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-CRESFIT1	This is a “tag” created using EQN button
U004	PSE	Display program name
U005	CF 10	
U006	RTN	

Code Customization

It is fairly easy to customize the code to make the program work for other three-parameter models with one nonlinear parameter.

The Extended Crescent Model

The extended crescent model has the following form:

$$y = A + B\,(1 - e^{-kt})$$

Working with this model is easy. The coefficient k is the nonlinear parameter and the coefficients A and B are the linear parameters.

Line	Command	Comments
B001	LBL B	Calculate R2 for a given value of K
B002	STO T	Store copy of working K value
B003	VIEW T	
B004	PSE	Pause to view K
B005	ClΣ	
B006	1	
B007	RCL C	
B008	1E3	
B009	÷	
B010	+	
B011	STO I	
B012	RCL (I)	Start the loop
B013	[0,1]	
B014	×	
B015	STO Y	Get y
B016	RCL (I)	
B017	[1,0]	
B018	×	
B019	STO X	Get x
B020	RCL T	
B021	×	
B022	e^x	
B023	1/x	
B024	+/-	
B025	1	
B026	+	
B027	Σ+	
B028	ISG I	
B029	GTO B012	End loop
B030	r	
B031	x^2	

B032	RTN	

The Offset Power Model

The offset power model has the following form:

$$y = A + B\ x^C$$

Working with this model is easy. The coefficient C is the nonlinear parameter and the coefficients A and B are the linear parameters.

Line	Command	Comments
B001	LBL B	Calculate R2 for a given value of K
B002	STO T	Store copy of working K value
B003	VIEW T	
B004	PSE	Pause to view K
B005	ClΣ	
B006	1	
B007	RCL C	
B008	1E3	
B009	÷	
B010	+	
B011	STO I	
B012	RCL (I)	Start the loop
B013	[0,1]	
B014	×	
B015	STO Y	Get y
B016	RCL (I)	
B017	[1,0]	
B018	×	
B019	STO X	Get x
B020	RCL T	
B021	y^x	
B022	×	
B023	Σ+	
B024	ISG I	
B025	GTO B012	End loop
B026	r	
B027	x^2	
B028	RTN	

The Extended Reciprocal Hyperbola Model

The extended reciprocal hyperbola model has the following form:

$$y = A + B \left(\frac{C\,x}{1+C\,x}\right)$$

Working with this model is easy. The coefficient A is the nonlinear parameter and the coefficients B and C are the linear parameters. The above model is transformed into:

$$\frac{1}{y-A} = \frac{1}{B} + \frac{1}{BC}\frac{1}{x}$$

If we let Y = 1/(y-A) and X = 1/x we have the following linearized model:

$Y = \alpha + \beta X$

Comparing the last two equations we deduce the following relation between the coefficients B and C and the intercept and slope of the linearized model:

$B = 1 / \alpha$
$C = \alpha / \beta$

Line	Command	Comments
B001	LBL B	Calculate R2 for a given value of K
B002	STO T	Store copy of working K value
B003	VIEW T	
B004	PSE	Pause to view K
B005	ClΣ	
B006	1	
B007	RCL C	
B008	1E3	
B009	÷	
B010	+	
B011	STO I	
B012	RCL (I)	Start the loop
B013	[0,1]	
B014	×	
B015	STO Y	Get y
B016	RCL T	
B017	-	
B018	1/x	
B019	RCL (I)	
B020	[1,0]	
B021	×	
B022	1/x	
B023	Σ+	

B024	ISG I	
B025	GTO B012	End loop
B026	R	
B027	x^2	
B028	RTN	

Two-Term Fourier Models

The two-term Fourier models have one of the following forms:

y = A + B sin(Cx)
y = A + B cos(Cx)

Working with these models is also easy. The coefficient C is the nonlinear parameter and the coefficients A and B are the linear parameters.

Line	Command	Comments
B001	LBL B	Calculate R2 for a given value of K
B002	STO T	Store copy of working K value
B003	VIEW T	
B004	PSE	Pause to view K
B005	ClΣ	
B006	1	
B007	RCL C	
B008	1E3	
B009	÷	
B010	+	
B011	STO I	
B012	RCL (I)	Start the loop
B013	[0,1]	
B014	×	
B015	STO Y	Get y
B016	RCL (I)	
B017	[1,0]	
B018	×	
B019	STO X	Get x
B020	RCL T	
B021	×	
B022	SIN	Or COS
B023	Σ+	
B024	ISG I	
B025	GTO B012	End loop
B026	r	
B027	x^2	
B028	RTN	

Sinusoid and Cosinusoid Models

The two-term Fourier models have one of the following forms:

$$y = A \sin(B + Cx)$$
$$y = A \cos(B + Cx)$$

Working with these models is also easy. The coefficient A is the nonlinear parameter and the coefficients B and C are the linear parameters. The iterations use the following forms:

$$\sin^{-1}(y/A) = B + Cx$$
$$\cos^{-1}(y/A) = B + Cx$$

Line	Command	Comments
B001	LBL B	Calculate R2 for a given value of K
B002	STO T	Store copy of working K value
B003	VIEW T	
B004	PSE	Pause to view K
B005	ClΣ	
B006	1	
B007	RCL C	
B008	1E3	
B009	÷	
B010	+	
B011	STO I	
B012	RCL (I)	Start the loop
B013	[0,1]	
B014	×	
B015	STO Y	Get y
B016	RCL T	
B017	÷	
B018	ASIN	Or ACOS
B019	RCL (I)	
B020	[1,0]	
B021	×	
B022	STO X	Get x
B023	Σ+	
B024	ISG I	
B025	GTO B012	
B026	r	
B027	x^2	
B028	RTN	

Hyperbolic Sine and Cosine Models

The hyperbolic sine and cosine models have one of the following forms:

$$y = A + \sinh(B + Cx)$$
$$y = A + \cosh(B + Cx)$$

Working with these models is also easy. The coefficient A is the nonlinear parameter and the coefficients B and C are the linear parameters. The iterations use the following forms:

$$\sinh^{-1}(y - A) = B + Cx$$

$$\cosh^{-1}(y - A) = B + Cx$$

Line	Command	Comments
B001	LBL B	Calculate R2 for a given value of K
B002	STO T	Store copy of working K value
B003	VIEW T	
B004	PSE	Pause to view K
B005	ClΣ	
B006	1	
B007	RCL C	
B008	1E3	
B009	÷	
B010	+	
B011	STO I	
B012	RCL (I)	Start the loop
B013	[0,1]	
B014	×	
B015	STO Y	Get y
B016	RCL T	
B017	-	
B018	ASINH	Or ACOSH
B019	RCL (I)	
B020	[1,0]	
B021	×	
B022	STO X	Get x
B023	Σ+	
B024	ISG I	
B025	GTO B012	
B026	r	
B027	x^2	
B028	RTN	

The above examples of code customization should cover other popular models and give you a general idea of how to deal with other three-parameter models with one nonlinear parameter

Example

Let's look at an example that exercises various routines in the regression program. The next Table contains (x, y) data points that we want to fit in a nonlinear regression model. The data represent time (variable x) and concentration of a gas in an organic liquid (variable y):

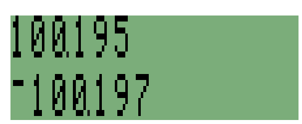
x	Y
0	0.00
1	13.9
2	25.9
3	36.2
4	45.1
5	52.8
6	59.3
7	65.0
8	69.9
9	74.1

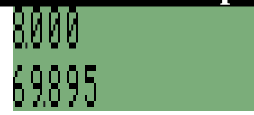
After you enter the above data, perform the following steps:

6. View all of the data points.
7. Calculate the nonlinear coefficient as well as the regression slope, intercept, and r^2.
8. Calculate the projected value of y for x = 8.

The following table shows the steps using the FIX 3 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0.000
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	0	R/S	Y?
	Enter a value for y.	0	R/S	1.000
3	Add the second data point.		XEQ A	X?
	Enter a value for x.	1	R/S	Y?
	Enter a value for y.	13.9	R/S	2.000
4	Repeat step 3 for all other data point.			10.000
5	View all data points (Note the program displays the results using the PSE		XEQ W	10.010 [9.000,74.100]

Step #	Step/Substep	Input	Command	Output
	statement). If you spot an error use XEQ E to overwrite the erroneous data point with the correct one.			
6	Calculate the regression coefficients.		XEQ C	K?
	Enter a guess for the nonlinear coefficient k.	0.1	R/S	S?
	Enter the initial search step.	0.01	R/S	M?
	Enter the minimum search step size. The program first displays the values for r^2 and nonlinear coefficient k. Note: The program pauses to display all values of K used in label B (by viewing variable T) and the pauses to display the best guesses for the nonlinear parameter k (by viewing the variable K).	0.0001	R/S	 Displays the values for r^2 and nonlinear coefficient k
	View the values for the intercept and slope. The fitted line is: $y = 100.195 – 100.97 e^{-0.15x}$ with an r^2 of 0.9999.		R/S	 Displays the values for the intercept and slope
7	Project an existing x onto y.		XEQ P	X?
	Enter the value for x.	8	R/S	X?

Step #	Step/Substep	Input	Command	Output
	View the value of x and $\hat{y}$.			 Shows the values of x and $\hat{y}$.

Difference Among Proportions Program NCS_DAP1

Overview

The NCS_DAP1 program calculates the statistics related to differences among proportions. The program offers the following features:

1. Storing the (x, n) data points as 2-D vectors (a new feature of the HP-35s) and thus reducing the number of registers used by half.
2. The ability to delete, edit, view, and swap data points. Thus the program incorporates basic operations to manage the data points.
3. Calculating the mean, standard deviation, and Student-t statistic.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Give an array of observed values $x_1, x_2, \ldots, x_k$ based on random variable with a binomial distribution. This distribution has the parameters n_i and θ_i for i = 1 to k. You can use a χ^2 statistic to test the null hypothesis stating that all of θ_i values are equal. The degrees of freedom for the χ^2 statistic equal k – 1.

$H_0: \theta_1 = \theta_2 = \theta_3 \ldots = \theta_k$

The program calculates the χ^2 statistic using the following equation:

$$\chi^2 = \Sigma\, [(x_i - n_i\,\hat{\theta})^2]/[\, n_i\,\hat{\theta}\,(1 - \hat{\theta})]$$

$$= \Sigma n_i\, [\Sigma(x_i^2 / n_i) / \Sigma x_i + \Sigma\, [(n_i - x_i)^2 / n_i] / \Sigma\, (n_i - x_i) - 1]$$

The summations are all taken for i = 1 to k.

The program calculates an estimate for the value of $\hat{\theta}$ using the following equation:

$$\hat{\theta} = \Sigma x_i / \Sigma n_i$$

If the calculated χ^2 statistic does not exceed the value of $\chi^2_{\alpha,k-1}$ then we cannot reject the null hypothesis stating that all the θ_i values are equal,

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value for x.	x	R/S	N?		
	Enter a value for n.	n	R/S	Number of data points		
3	Delete a data point.		XEQ D	I?		
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message INVALID DATA.	Index of data point to delete	R/S	Number of data points		
4	Edit a data point.		XEQ E	I?		
	Enter the index for the data point you want to edit.	Index	R/S	X?		
	Enter a new value for x.	x	R/S	N?		
	Enter a new value for n.	n	R/S			
5	View a data point.		XEQ V	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	Data point	Index	
	To view the next data point. You can repeat this substep to view a		R/S	Data point	Index	

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	succession of data points.					
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1		
6	View all data points (Note the program displays the results using the PSE statement.) Note: If you spot an error in the data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	Data point	Index	
7	Swap data points.		XEQ S	I?		
	Enter the index of the first data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the first data point	R/S	J?		
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S			
8	Calculate the results (using arrays) and view the difference mean, and the difference standard deviation.		XEQ C	$\hat{\theta}$	χ^2	k-1
	To view the degrees of freedom.		R↓	χ^2	k-1	
9	Display the program		XEQ U	NCS-DAP1		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	name. The program pauses to display the program name.					

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a data point	A		1	Set = Pause, Clear = Stop
C	Calculate statistics for difference among proportions	B			
D	Delete a data point	C	Number of data points, k		
E	View/Edit a data point	D	$\Sigma x(i)^2/n(i)$		
W	View all data points	E	Σ (n(i) - x(i))		
V	View all data points without stopping the program	F	$\Sigma (n(i) - x(i))^2 /n(i)$		
I	Initialize the program	G			
U	Display program name	H			
		I	Index		
H	Validate index	J	Index		
G	Prompt data from use	K			
R	Generate INVALID Data message	L			
		M			
		N	n(i)		
		O			
		P			
		Q			
		R			
		S			
		T	$\hat{\theta}$		
		U			
		V			
		W			
		X	x(i)		
		Y			
		Z	chi-square		
		1	[x,n] for data point 1		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		2	[x,n] for data point 2		
		…			
		C	[x,n] for data point C		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	

Line	Command	Comments
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO- C	
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	FS? 1	
S005	RTN	
S006	INPUT J	
S007	XEQ H	
S008	FS? 1	
S009	RTN	
S010	RCL (I)	Recall data points on the stack
S011	RCL (J)	
S012	STO (I)	Store data point in swapped indices
S013	x↔y	
S014	STO (J)	
S015	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	FS? 1	
V005	RTN	
V006	RCL (I)	Recall data point
V007	STOP	
V008	1	
V009	RCL I	
V010	+	
V011	STO I	

Line	Command	Comments
V012	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	
R004	RTN	
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT N	Prompt user for n value
G004	[X,N]	Create data point
G005	RTN	
U001	LBL U	
U002	SF 10	
U003	NCS-DAP1	This is a "tag" created using the EQN button
U004	PSE	
U005	CF 10	
U006	RTN	
C001	LBL C	Calculate statistics for difference among proportions
C002	ClΣ	Clear stat variables
C003	0	Clear special summations registers
C004	STO D	
C005	STO E	
C006	STO F	
C007	1	
C008	RCL C	
C009	1E3	
C010	÷	
C011	+	
C012	STO I	Initialize the loop variable
C013	RCL (I)	Start the loop to update the sat summations
C014	[0,1]	
C015	×	

Line	Command	Comments
C016	STO N	Get n(i)
C017	RCL (I)	
C018	[1,0]	
C019	×	
C020	STO X	Get x(i)
C021	Σ+	
C022	RCL X	
C023	x^2	
C024	RCL N	
C025	÷	
C026	STO+ D	
C027	RCL N	
C028	RCL X	
C029	-	
C030	STO+ E	
C031	x^2	
C032	RCL N	
C033	÷	
C034	STO+ F	
C035	ISG I	
C036	GTO C013	End of loop that updates the stat summation
C037	RCL D	Start evaluating chi-square
C038	Σx	
C039	÷	
C040	RCL F	
C041	RCL E	
C042	÷	
C043	+	
C044	1	
C045	-	
C046	Σy	
C047	×	
C048	STO Z	Calculate and store chi-square
C049	Σx	
C050	Σy	
C051	÷	
C052	STO T	Calculate and store Theta hat
C053	SF 10	
C054	DF,CHI,THETA	This is a "tag" created using the EQN button
C055	PSE	Display the output tag
C056	CF 10	
C057	RCL C	
C058	1	

Line	Command	Comments
C059	-	Calculate degrees of freedom
C060	RCL Z	
C061	RCL T	
C062	RTN	Display the degrees of freedom, chi-square, and Theta hat

Example

Let's look at an example that exercises various routines in the NCS_DAPT1 program. The next table contains (x, n) data points that we want to use to test the null hypothesis.

x	n
232	400
260	500
197	400

After you enter the above data, calculate the chi-square statistic and theta hat, and the test the null hypothesis at a 95% confidence level.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	232	R/S	N?
	Enter a value for y.	400	R/S	1.00
	Repeat the above steps to enter the other data points.			3.00
3	Calculate the statistical results. The program displays the values of the chi-square and theta hat		XEQ C	 Displays the chi-square and theta hat.
	Display the value of the degrees of freedom.		R↓	 Displays the value for the degrees of freedom and chi-square

Since the calculated χ^2 statistic of 6.47 exceeds the value of $\chi^2_{2,0.05}$ of 5.991 we cannot accept the null hypothesis stating that all of values θ_i for are equal.

Appendix A contains a new approximation that I recently developed for inverse χ^2 function. The appendix also contains tables for selected values. Using the approximation for χ^2 presented in the appendix, you get a value of 6.49, which has an 8.5% error. This error is too significant for our example. Keep in mind that the approximation presented in the appendix is for degrees of freedom of 5 and greater.

Intraclass Correlation Coefficient Program NCS_INTRACLASS1

Overview

The NCS_INTRACLASS1 program calculates the intraclass correlation coefficient for a table of k × n observations. The program stores the input data in the unnamed memory registers. Since there are quite a few unnamed registers, you can store a data table with a good number rows and columns.

The program offers the following operations:

1. Initializing the program.
2. Adding entries for the data table.
3. Editing entries in the data table by specifying the column and row indices for the element to edit.
4. Calculating the row sums, grand total, ANOVA table elements, and the intraclass correlation coefficient.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Given an n × k data table:

		Observation j				Totals
		1	2	…	n	
Group i	1	x_{11}	x_{12}	…	x_{1n}	T_1
	2	x_{21}	x_{22}	…	x_{2n}	T_2
	…	…	…	…	…	
	k	x_{k1}	x_{k2}	…	x_{kn}	T_k
						T

The rows and grand sums are given by:

Sum group $T_i = \sum_{j=1}^{n} x_{ij}$ for i = 1 to k

Grand total = $\sum_{i=1}^{n} T_i$ for j = 1 to k

Sum of squares mean:

$$MSS = T^2/kn$$

Sum of squares among groups:

$$ASS= \sum_{i=1}^{n} T_i^2 / n - MSS$$

Sum of squares within groups:

$$WSS = \sum_{i=1}^{k} \sum_{j=1}^{n} x_{ij} - MSS - ASS$$

The degrees of freedom:

$$df1 = k - 1$$

$$df2 = k (n-1)$$

The F statistic:

$$F = \frac{ASS}{k-1} / \frac{WSS}{k(n-1)}$$

The intraclass correlation coefficient:

$$r_I = \left(\frac{ASS}{k-1} - \frac{WSS}{k(n-1)}\right) / \left(\frac{ASS}{k-1} - \frac{WSS}{k}\right)$$

The value of r_I is an estimate for the population intraclass correlation coefficient, ρ_I.

If the calculated F statistic does not exceeds the F at k-1 and k(n-1) degrees of freedom, and 100(1-α)% confidence level, then we cannot reject the null hypothesis that says there $\rho_I = 0$. This means that the data has no correlation among groups, and are random samples from normal population with the same variance.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	K?	
	Enter the number of rows.	k	R/S	N?	
	Enter the number of column.	n	R/S	1E-499	
2	Add data points.		XEQ A	X?	
	Enter a value for x.	x	R/S	X?	
	Enter the next value. The sequence of input is by rows. Repeat this step until you have entered all of the elements in the data table.	x	R/S	X?	
3	View/Edit a data point.		XEQ E	I?	
	Enter the row index.	Index	R/S	J?	
	Enter the column index.	Index	R/S	X?	
	View the value and optionally enter a replacement value for x.	x	R/S	x	
4	View the data while pausing at each value. The program displays the values in the table by rows.		XEQ V	x	
5	View the data while stopping at each value. The program displays the values in the table by rows.		XEQ W	x(1,1)	
	To view the next value in the matrix x.		R/S	x(2,1)	
	Repeat this step for the rest of the elements in matrix x.		R/S	X(I,J)	
6	Calculate the ANOVA table and intraclass correlation coefficient.		XEQ C	Row sum	
	View the next row sum.		R/S	Row sum	
	Repeat the above step for the rest of the rows. The program displays the values for the grand total and MSS.		R/S	MSS	Total
	The program displays the		R/S	WSS	ASS

Step #	Step/Substep	Input	Command	Output X	Output Y
	values for ASS and WSS.				
	The program displays the values for df1 and df2.		R/S	df2	df1
	The program displays the values for rI and F.		R/S	F	rI
7	Display the program name. The program pauses to display the program name.		XEQ U	NCS-INTRACLASS1	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add data points	A	n*k	1	Set = Pause, Clear = Stop
C	Calculate intraclass correlation coefficient	B	Row sum		
E	View/Edit an observations	C	MSS		
W	View all data points in matrix x	D	ASS		
V	View all data points in matrix x without stopping the program	E	WSS		
I	Initialize the program	F	F		
U	Display program name	G	df1		
	Labels for internal use	H	df2		
X	Set index to access x(I,J)	I	Index to access x(I,J)		
		J	Index to access row or column sums		
		K	Number of rows		
		L	Column counter		
		M			
		N	Number of columns		
		O			
		P			
		Q			
		R	Intraclass correlation coefficient, rI		
		S			
		T	Grand sum of x(I,J)		
		U			
		V			
		W			

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		X	x(I,J)		
		Y			
		Z			
		1	x(1,1)		
		2	x(1,2)		
		…			
		n	x(1,n)		
		n+1	x(2,1)		
		n+2	x(2,2)		
		…			
		(I-1)n + J	x(I,J)		
		…			
		k*n	x(k,n)		

The above table shows that the program stores the matrix x in the unnamed registers as a series of chained columns. That is why the input of the elements in matrix x also follows a column-wise order.

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	INPUT K	Prompt for the number of columns
I003	INPUT N	Prompt for the number of rows
I004	RCL K	
I005	RCL N	
I006	×	
I007	STO A	Store n×k
I008	RTN	
X001	LBL X	Set index to access x(I,J)
X002	1	
X003	RCL A	
X004	1E3	
X005	÷	
X006	+	
X007	STO I	
X008	RTN	
A001	LBL A	Add data points
A002	XEQ X	Set index to access x(I,J) by column
A003	INPUT X	Start input loop
A004	STO(I)	
A005	ISG I	
A006	GTO A003	End of input loop

Line	Command	Comments
A007	RTN	
E001	LBL E	View/Edit an observations
E002	INPUT I	Enter index I
E003	RCL K	Validate index I
E004	x<y?	
E005	ACOS(2)	Display error message
E006	INPUT J	Enter index J
E007	RCL J	
E008	RCL N	
E009	x<y?	
E010	ACOS(2)	Display error message
E011	RCL I	Calculate index for x(I,J)
E012	1	
E013	-	
E014	RCL N	
E015	×	
E016	RCL J	
E017	+	
E018	STO I	
E019	RCL (I)	Recall current x(I,J) value
E020	STO X	
E021	INPUT X	Prompt user for new value
E022	STO (I)	Store new value for x(I,J)
E023	RTN	
V001	LBL V	View all data points in matrix x without stopping the program
V002	SF 1	Set flag 2 to make the program pause when showing x(I,J)
V003	XEQ W	
V004	CF 1	
V005	RTN	
W001	LBL W	View all data points in matrix x
W002	XEQ X	Set index to access data points
W003	RCL (I)	Start loop to display data
W004	FS? 1	
W005	GTO W008	
W006	STOP	
W007	GTO W009	
W008	PSE	
W009	ISG I	
W010	GTO W003	
W011	RTN	
C001	LBL C	Calculate intraclass correlation coefficient
C002	XEQ X	Set index to access x(I,J) elements
C003	0	Initialize summations
C004	STO T	

Line	Command	Comments
C005	STO D	
C006	STO E	
C007	0	Start outer loop to handle various rows
C008	STO B	Initialize the sum of the current row
C009	RCL N	
C010	STO L	Initialize the column counter
C011	RCL (I)	Start the inner loop
C012	STO+ B	Update the sum of the current row
C013	STO+ T	Update the grand sum of x(I,J)
C014	x^2	
C015	STO+ E	Update the summation part of WSS
C016	1	
C017	STO- L	Decrement column counter
C018	RCL L	Still processing the current row?
C019	x>0?	
C020	GTO C032	
C021	SF 10	
C022	ROW SUM	This is a "tag" created using the EQN button
C023	PSE	Display output tag
C024	CF 10	
C025	RCL B	
C026	STOP	Display row sum
C027	x^2	
C028	STO+ D	
C029	ISG I	
C030	GTO C007	End of outer loop
C031	GTO C034	
C032	ISG I	
C033	GTO C011	End of inner loop
C034	SF 10	
C035	RCL T	
C036	x^2	
C037	RCL A	
C038	÷	
C039	STO C	Calculate MSS
C040	TOTAL, MSS	This is a "tag" created using the EQN button
C041	PSE	Display output tag
C042	RCL T	
C043	x↔y	
C044	STOP	Display sum of x(I,J) and MSS
C045	RCL D	
C046	RCL N	
C047	÷	
C048	RCL C	

Line	Command	Comments
C049	-	
C050	STO D	Calculate ASS
C051	RCL E	
C052	RCL C	
C053	-	
C054	RCL D	
C055	-	
C056	STO E	Calculate WSS
C057	ASS, WSS	This is a "tag" created using the EQN button
C058	PSE	Display output tag
C059	RCL D	
C060	x↔y	
C061	STOP	Display ASS and WSS
C062	RCL K	
C063	1	
C064	-	Calculate df1
C065	STO G	
C066	RCL A	
C067	RCL K	
C068	-	Calculate df2
C069	STO H	
C070	DF1,DF2	This is a "tag" created using the EQN button
C071	PSE	Display output tag
C072	STOP	Display df1 and df2
C073	RCL D	Start calculating rI
C074	RCL K	
C075	1	
C076	-	
C077	÷	
C078	STO F	
C079	RCLE	
C080	RCL A	
C081	RCL K	
C082	-	
C083	÷	
C084	STO÷ F	Calculate F
C085	-	
C086	RCL D	
C087	RCL K	
C088	1	
C089	-	
C090	÷	
C091	RCL E	
C092	RCL K	

Line	Command	Comments
C093	÷	
C094	+	
C095	÷	Calculate rI
C096	STO R	
C097	RCL F	
C098	RI,F STAT	This is a "tag" created using the EQN button
C099	PSE	Display output tag
C100	CF 10	
C101	RTN	Display rI and F
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-INTRACLASS1	This is a "tag" created using the EQN button
U004	PSE	
U005	CF 10	
U006	RTN	

Example

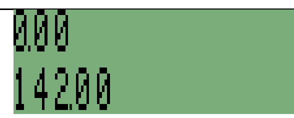
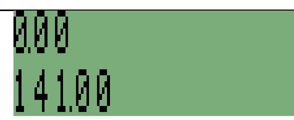
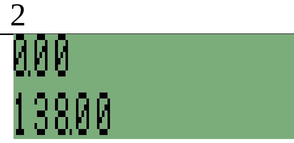
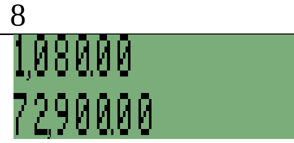
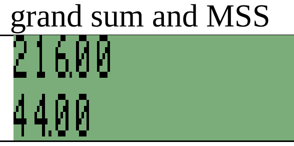
Let's look at an example that exercises various routines in the NCS_INTRACLASS1 program. The next table contains the data.

	1	2
1	71	71
2	69	72
3	59	65
4	65	64
5	66	60
6	73	72
7	68	67
8	70	68

After you enter the above data, perform the following steps:

1. Calculate and display the chi-square and contingency coefficient.
2. Display the row sums.
3. Display the column sums.
4. Display the grand sum.
5. Determine if the row and column treatment are significant at a 95% confidence level.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	K?
	Enter the number of row.	8	R/S	N?
	Enter the number of column.	2	R/S	
2	Enter the data.		XEQ A	X?
	Enter x(1,1)	71	R/S	X?
	Enter x(1,2)	71	R/S	X?
	Enter x(2,1)	69	R/S	X?
	Enter x(2,2)	72	R/S	X?
	Enter x(3,1)	59	R/S	X?
	Enter x(3,2)	65	R/S	X?
	Enter x(4,1)	65	R/S	X?
	Enter x(4,2)	64	R/S	X?
	Enter x(5,1)	66	R/S	X?
	Enter x(5,2)	60	R/S	X?
	Enter x(6,1)	73	R/S	X?
	Enter x(6,2)	72	R/S	X?
	Enter x(7,1)	68	R/S	X?
	Enter x(7,2)	67	R/S	X?
	Enter x(8,1)	70	R/S	X?
	Enter x(8,2)	68	R/S	
3	Calculate the ANOVA table components and the intraclass correlation coefficient. The first output is the sum of the first row.		XEQ C	 Displays the sum for row 1
	The program displays the sum for the next row.		R/S	 Displays the sum for row 2
	Repeat the above steps to view the sums of the remaining row		R/S	 Displays the sum for row 8
	The program displays the values for the grand total and MSS.		R/S	 Displays the values for the grand sum and MSS
	The program displays the values for ASS and WSS.		R/S	

Step #	Step/Substep	Input	Command	Output
				Displays the values for ASS and WSS
	The program displays the values for df_1 and df_2.		R/S	 Displays the values for df_1 and df_2
	The program displays the values for r_I and F.		R/S	 Displays the values for r_I and the F statistic

Since the calculated F of 5.61 exceeds the value of $F_{7,8,0.05}$ (equal to 3.5005) we cannot accept the null hypothesis stating that $\rho_I = 0$ (there is no correlation among groups). The value of r_I seems high enough above 0 and close to 1, which enforces the conclusion about the null hypothesis.

Appendix A contains a new approximation that I recently developed for inverse F function. The appendix also contains tables for selected values.

Kendall Test Program NCS_KENDALL1

Overview

The NCS_KENDALL1 program calculates the Kendall concordance coefficient for a matrix of rank values.

The program offers the following operations:

1. Entering the matrix of ranks.
2. Editing a rank value.
3. Viewing specific rank value.
4. Viewing the values of the array of ranks.
5. Calculating the Kendall concordance coefficient and related statistics.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

The Kendall concordance coefficient is calculated using a matrix of ranks. Each column in the matrix represents the rank values supplied by an observer. The rows of the matrix represent the ranks for a specific item.

The Kendall concordance coefficient is obtained using:

$$W = \frac{12\ \sum_{i=1}^{n}\left(\sum_{j=1}^{k} R_{ij}\right)^2}{k^2\ n\ (n^2-1)} - \frac{3(n+1)}{n+1}$$

Here n and k are the number of rows and column in the matrix, respectively. The values of W range from 0 to 1. The value of zero indicates that observers are in total disagreement on how they ranked the items. By contrast, a value of one indicates that observers are in total agreement.

To test the null hypothesis:

H_0: The observers have no community of preference.

When the value of n exceeds 7, we calculate an approximation to the chi-square statistic using the following equation:

$$\chi^2 = k\ (n - 1)\ W$$

If the calculated $\chi 2$ does not exceed $\chi 2_{\alpha,n-1}$ then we cannot reject the null hypothesis that the observers have no community of preference.

For small values of n consult your favorite statistics book to find special tables for the hypothesis test.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	K?	
	Enter the number of columns in the rank matrix.	k	R/S	N?	
	Enter the number of rows in the rank matrix.	n	R/S	Value of 1 + nk/1000	
2	Enter the elements of the rank matrix. Enter the values row by row,		XEQ A	R?	
	Enter a value for Rank(1,1).	Rank	R/S	R?	
	Enter a value for Rank(1,2).	Rank	R/S	R?	
	Enter the next rank value. Repeat this step until you have entered all of the rank value.	Rank	R/S	R?	
3	Edit a rank value.		XEQ E	I?	
	Enter the row index for the sought rank.	Row index	R/S	J?	
	Enter the column index for the sought rank.	Column Index	R/S	R?	
	View the value and optionally enter a replacement value for the rank.	Rank	R/S		
4	View the data while		XEQ V	rank	

Step #	Step/Substep	Input	Command	Output X	Output Y
	pausing at each value. The program displays the values in the table by columns.				
5	View the data while stopping at each value.		XEQ W	rank	
	To view the next rank value.		R/S	rank	
	Repeat the above steps to view the rest of the matrix rank, row by row.		R/S	rank	
6	Calculate the degrees of freedom, Kendall test statistic, and chi-square value. The program displays the latter two statistics.		XEQ C	chi-square	W
	View the degrees of freedom.		R↓	W	Degrees of freedom
7	Display the program name. The program pauses to display the program name.		XEQ U	NCS-KENDALL1	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Enter the data in the matrix R	A		1	Set = Pause, Clear = Stop
C	Calculate Kendall's Coefficient of Concordance	B			
E	View/Edit an observation	C			
W	View all data points in matrix R	D	Degrees of freedom = N-1		
V	View all data points in matrix R without stopping the program	E			
I	Initialize the program	F			
U	Display program name	G			
		H			
		I	Index		
		J	Index		
		K	The number of		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
			columns K		
		L	value of 1 + NK/1000		
		M			
		N	The number of rows N		
		O			
		P			
		Q			
		R	Rank(I,J)		
		S	Sum of Rank for a row		
		T	Sum of Squared sums of row ranks		
		U			
		V			
		W	Kendall's Coefficient of Concordance, W		
		X			
		Y			
		Z			
		1	R(1,1)		
		2	R(1,2)		
		…			
		k	R(1,k)		
		k+1	R(2,1)		
		k+2	R(2,2)		
		…			
		2k	R(2,k)		
		2k+1	R(3,1)		
		2k+2	R(3,2)		
		…			
		(I-1)k+J	R(I,J)		
		…			
		nk	R(n,k)		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	INPUT K	Prompt the user to enter the value for k
I003	INPUT N	Prompt the user to enter the value for n
I004	RCL N	
I005	RCL K	
I006	×	
I007	XEQ L	
I008	STO L	Store loop control variable to sequentially access the entire elements of matrix R, row by row
I009	RTN	
L001	LBL L	
L002	1E3	
L003	÷	
L004	1	
L005	+	
L006	RTN	
U001	LBL U	
U002	SF 10	
U003	NCS-KENDALL1	This is an "output tag" created using the EQN button
U004	PSE	
U005	CF 10	
U006	RTN	
A001	LBL A	Enter the data in the matrix R
A002	RCL L	
A003	STO I	Set the loop control variable
A004	INPUT R	Start the loop to input the elements of matrix R, row by row
A005	STO (I)	
A006	ISG I	
A007	GTO A004	End of input loop
A008	RTN	
E001	LBL E	View/Edit an observation
E002	INPUT I	Prompt the user to enter the row index
E003	RCL N	Validate the row index
E004	x<y?	
E005	ACOS(2)	
E006	INPUT J	Prompt the user to enter the column index
E007	RCL K	Validate the column index
E008	x<y?	
E009	ACOS(2)	
E010	RCL I	
E011	1	

Line	Command	Comments
E012	-	
E013	RCL K	
E014	×	
E015	RCL J	
E016	+	
E017	STO I	Calculate and store index to acccess R(I,J)
E018	RCL (I)	Get the current value of R(I,J)
E019	STO R	
E020	INPUT R	Display current value and prompt for new value
E021	STO (I)	Store new value of R(I,J)
E022	RTN	
E023	LBLV	View all data points in matrix R without stopping the program
E024	SF 1	
E025	XEQ W003	
E026	CF 1	
E027	RTN	
W001	LBL W	View all data points in matrix R
W002	CF 1	
W003	RCL L	
W004	STO I	Initialize the loop variable
W005	RCL (I)	Start the loop to view the matrix R, row by row
W006	FS? 1	
W007	GTO W010	
W008	STOP	
W009	GTO W011	
W010	PSE	
W011	ISG I	
W012	GTO W005	End of loop
W013	RTN	
C001	LBL C	Calculate Kendall's Coefficient of Concordance
C002	0	Initialize the double summation
C003	STO T	
C004	RCL L	
C005	STO I	Initialize the main loop variable
C006	0	Start outer loop
C007	STO S	Initialize the sum of row ranks
C008	1	
C009	STO J	Set the column index
C010	RCL (I)	Start the inner loop
C011	STO+ S	Update sum of ranks
C012	1	
C013	STO+ J	
C014	RCL J	
C015	RCL K	

Line	Command	Comments
C016	x≥y?	
C017	GTO C024	
C018	RCL S	
C019	x^2	
C020	STO+ T	
C021	ISG I	
C022	GTO C006	End of inner loop
C023	GTO C026	
C024	ISG I	
C025	GTO C010	End of outer loop
C026	RCL T	Start to calculate W
C027	12	
C028	×	
C029	RCL K	
C030	x^2	
C031	÷	
C032	RCL N	
C033	/	
C034	LASTx	
C035	x^2	
C036	1	
C037	-	
C038	÷	
C039	RCL N	
C040	1	
C041	+	
C042	3	
C043	×	
C044	RCL N	
C045	1	
C046	-	
C047	STO D	Calculate and store the degrees of freedom
C048	÷	
C049	-	
C050	STO W	Calculate and store W
C051	RCL D	
C052	×	
C053	RCL K	
C054	×	
C055	STO Z	Calculate and store chi-square
C056	SF 10	
C057	W,CHI	This is an "output tag" created using the EQN button
C058	PSE	Display output tag

Line	Command	Comments
C059	CF 10	
C060	RCL D	
C061	RCL W	
C062	RCL Z	
C063	RTN	Display W and chi-square

Example

Let's look at an example that exercises various routines in the NCS_KENDALL1 program. Here is a table containing rank data:

Item #	Observer 1	Observer 2	Observer 3
1	6	7	3
2	1	4	2
3	9	3	5
4	2	6	1
5	10	8	9
6	3	2	6
7	5	9	8
8	4	1	4
9	8	10	10
10	7	5	7

The program requires that you only enter the ranks row by row. After you enter the above data, calculate the degrees of freedom, Kendall statistic, and chi-square value. Use the chi-square to check the null hypothesis at a 95% confidence level. If you enter erroneous values, execute the XEQ E command to edit these values. You can also view the data using the command XEQ W or XEQ V before (or even after) you calculate the statistical results.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	K?
	Enter the number of columns in the rank matrix.	3	R/S	N?
	Enter the number of rows in the rank matrix.	10	R/S	1.03
2	Enter the data.		XEQ A	R?
	Enter the value of Rank(1,1).	6	R/S	R?
	Enter the value of Rank(1,2).	7	R/S	R?

Step #	Step/Substep	Input	Command	Output
	Repeat the above steps to enter the rest of the rank values, row by row.			
3	Calculate degrees of freedom, Kendall statistic, and chi-square value.		XEQ C	0.69 18.64 Displays the Kendall test statistic and chi-square value
	View the degrees of freedom.		R↓	9.00 0.69 Displays the degrees of freedom and the Kendall statistic.

Since the calculated χ^2 value of 18.64 exceeds the value of $\chi^2_{0.05,9}$ (equal to 16.91) we cannot accept the null hypothesis, stating that the observers have no community of preference.

Using the approximation for χ^2, shown in the appendix, you get $\chi^2 = 17.13$ for a significance level of 0.05, with a 1.35% error.

Appendix A contains a new approximation that I recently developed for inverse χ^2 function. The appendix also contains tables for selected values.

Kruskal-Wallis Test Program NCS_KRUSKAL1

Overview

The NCS_KRUSKAL1 program implements the Kruskal-Wallis test to determine if several groups of ranked data belong to a single population. The program does not store the raw data for each data group. Doing so and offering the expected features of adding, deleting, and deleting data points can be taxing on the calculator resources. Instead, the program stores the summation data and supports the following operations:

1. Adding and deleting data within the currently edited group.
2. Ending the work on a data group and starting a new data group. The program can support as many groups as allowed by the number of unnamed memory registers. The program stores the number of data points and the sum of rank values for each data group using a 2-D vector.
3. Recalling a previously stored data group to add or delete observations. This feature gives flexibility to the user by not *sealing* the data once you move on to another data group.
4. Calculating Kruskal-Wallis statistic and other related statistics.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

The program handles a set of k data groups whose values are ranked as a single group. Each group has n_i number of rank values. The null hypothesis states that samples of size n_1, n_2, …, n_k, belong to same population.

The program calculates the Kruskal-Wallis statistic, H, which is used to test the null hypothesis. When each of the sample sizes exceeds 5, the Kruskal-Wallis statistic is distributed approximately as chi-square with k – 1 degrees of freedom. When some or all of the samples have 5 or smaller values, you need to consult special tables to use the Kruskal-Wallis statistic.

H_0 : All k samples belong to same population.

The program calculates the Kruskal-Wallis statistic using the following equation:

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{\left(\sum_{j=1}^{n_i} R_{ij}\right)^2}{n_i} - 3(N+1)$$

$$N = \sum_{i=1}^{k} n_i$$

If the value of H does not exceed the value of $\chi^2_{\alpha,k-1}$ we cannot reject the null hypothesis which states that the ranked samples come from the same population.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	R?		
	Enter a value for the rank.	Rank	R/S	Number of data points		
	Enter another value.		R/S	R?		
	Resume with the steps above.					
3	Delete a data point.		XEQ D	R?		
	Enter a value for the rank.	Rank	R/S	Number of data points		
4	End the data entry for the current group		XEQ E	[n, Σx]		
	Resume with step 2 to enter new data.					
5	Recall summations of previous data group.		XEQ R	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S			
	You can resume at step 2 to add more data and/or step 3 to delete data. When you are done, execute step					

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	4.					
6	To calculate the degrees of freedom, the Kruskal-Wallis statistic, and the total number of observations		XEQ C	H	N	k-1
	Enter the index of the one group. If you enter an invalid index (greater than the number of actual sets), the program displays the INVALID DATA error message. . Entering an index of 0 or less has unpredictable effects!		R↓	N	k-1	
7	Display the program name. The program pauses to display the program name.		XEQ U	NCS-KRUSKAL1		

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a data point to the current group	A		1	Set = recall data group mode
C	Calculate the Kruskal-Wallis statistic.	B		10	Used to display equations as output tags
D	Delete a data point from current group	C			
E	End the data entry for the current group	D			
I	Initialize the program	E			
R	Recall summations of previous data group	F			
U	Display the program	G			

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
	name				
		H	Kruskal-Wallis statistic H		
Internally used labels		I	Index		
P	Parse data recalled from unnamed variables	J	Index		
		K			
		L			
		M			
		N	Number of data groups, k		
		O			
		P			
		Q			
		R	Rank		
		S	ΣRank(i) for a group		
		T	Σ [ΣRank(i,j)^2] /n(i)		
		U	N = Σn(i)		
		V			
		W			
		X			
		Y			
		Z	Used		
		1	[n, ΣRank] for group 1		
		2	[n, ΣRank] for group 2		
		…			
		k	[n, ΣRank] for group k		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	ClΣ	
I003	0	
I004	STO N	Initialize data group counter
I005	CF 1	Clear recall data group mode
I006	RTN	
A001	LBL A	Add data point to the current data group
A002	INPUT R	
A003	RCL R	
A004	Σ+	
A005	STOP	

Line	Command	Comments
A006	GTO A001	
D001	LBL D	Delete a data point from the current data group
D002	INPUT R	
D003	RCL R	
D004	Σ-	
D005	RTN	
E001	LBL E	End the data entry for the current data group
E002	FS? 1	Is recalled data group mode on?
E003	GTO E008	
E004	1	
E005	STO+ N	
E006	RCL N	
E007	STO I	
E008	[n, Σx]	
E009	STO (I)	
E010	ClΣ	Clear summation registers block
E011	CF 1	Clear recalled data group mode
E012	RTN	
P001	LBL P	
P002	STO Z	
P003	[0,1]	
P004	×	
P005	RCL Z	
P006	[1,0]	
P007	×	
P008	RTN	
R001	LBL R	Recall an existing data group
R002	FS? 1	Already editing another recalled group?
R003	GTO R006	
R004	XEQ E	
R005	GTO R009	
R006	[n, Σx]	
R007	STO (I)	
R008	ClΣ	
R009	INPUT I	
R010	SF 1	Set group recall mode
R011	RCL (I)	
R012	XEQ P	Parse recalled data group
R013	-27	
R014	STO J	
R015	R↓	
R016	STO (J)	Store n
R017	R↓	

Line	Command	Comments
R018	1	
R019	STO- J	
R020	R↓	
R021	STO (J)	Store sum of x
R022	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-KRUSKAL1	This is a "tag" created using the EQN button
U004	PSE	
U005	CF 10	
U006	RTN	
C001	LBL C	Calculate Kruskal-Wallis statistic
C002	0	Initialize grand sum and N
C003	STO T	
C004	STO U	
C005	1	
C006	RCL N	
C007	1E3	
C008	/	
C009	+	
C010	STO I	Set value for loop variable
C011	RCL (I)	Start of loop
C012	XEQ P	
C013	STO+ U	Update N
C014	x↔y	
C015	x^2	
C016	x↔y	
C017	÷	
C018	STO+ T	Update grand sum
C019	ISG I	
C020	GTO C011	End of loop
C021	RCL T	Start calculating H
C022	12	
C023	×	
C024	RCL U	
C025	÷	
C026	RCL U	
C027	1	
C028	+	
C029	÷	
C030	LASTx	
C031	3	

Line	Command	Comments
C032	×	
C033	-	Calculate H
C034	STO H	
C035	SF 10	
C036	DF,N,H	This is a "tag" created using the EQN button
C037	PSE	Display output tag
C038	CF 10	
C039	RCL N	
C040	1	
C041	-	Calculate degrees of freedom
C042	RCL U	
C043	RCL H	Display df, N, and H
C044	RTN	

Example

Let's look at an example that exercises various routines in the NCS_KRUSKAL1 program. The next table contains ranked data that we want to use to test the null hypothesis.

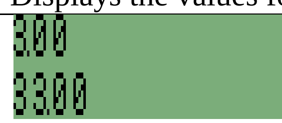
	j									
i	1	2	3	4	5	6	7	8	9	10
1	29	5	26	10	33	30				
2	11	12	9	7	20	18	19	21		
3	14	28	8	25	17	15	32	4	2	
4	6	27	3	16	24	13	1	31	22	23

After you enter the above data, perform the following steps:

6. Calculate Kruskal-Wallis statistic H and the total number of observations N.
7. Test the null hypothesis at a 95% confidence level.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Add the first data point in the first data group.		XEQ A	R?
	Enter a value for Rank(1,1).	29	R/S	1.00
	Repeat the above steps for the rest of			

Step #	Step/Substep	Input	Command	Output
	the values in the first data groups.			
3	End the first data group		XEQ E	
4	Add the first data point in the second data group.		XEQ A	R?
	Enter a value for Rank(2,1).	11	R/S	1.00
	Repeat the above steps for the rest of the values in the second data groups.			
5	End the second data group		XEQ E	
6	Repeat the steps 2 to 5 to enter the rank values in third and fourth data groups.			
7	Calculate the statistical results. The program displays the values for N and the Kruskal-Wallis statistic H.		XEQ C	 Displays the values for N and H
	View the value for the degrees of freedom.		R↓	 Displays the values for the degrees of freedom and N

Since the calculated Kruskal-Wallis statistic has a value of 2.29 which does not exceed the value of $\chi^2_{3,0.05}$ of 7.815 we cannot reject the null hypothesis stating that samples come from the same population.

Appendix A contains a new approximation that I recently developed for inverse χ^2 function. The appendix also contains tables for selected values.

Latin Square ANOVA Table Program NCS_LATIN_SQR1

Overview

The NCS_LATIN_SQR1 program calculates the elements of a Latin square ANOVA table. The program offers the following features:

4. Handling Latin square designs of different sizes and is not limited to one size.
5. Adding observations for each cell, row, and column treatment. The input proceeds row-wise, cell by cell.
6. Editing observations at specific cell, row, and column indices..
7. Viewing the data matrix.
8. Calculating and display the elements of the Latin square ANOVA table.

The program stores the original data, using indirect addressing, in unnamed memory registers. This approach allows you to view and edit your original data.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

A Latin square ANOVA table processes data that has a special layout aimed at examining the effects or three factors. Here is an example of a 3 × 3 Latin square data layout:

	Columns		
Rows	1	2	3
1	A x_{11}	B x_{12}	C x_{13}
2	C x_{21}	A x_{22}	B x_{23}
3	B x_{31}	C x_{32}	A x_{33}

The above table is an example of a Latin square layout that allows you to example the effect of columns, rows, and the *treatments*. Each cell has a treatment designation. The Latin square data layout must conform to the following rules:

1. The layout is square and has an equal number of rows and columns.
2. The layout has an equal number of rows, columns, and treatments.
3. Each row or column must have all of the treatments appearing once and only once.
4. Not two or more rows and columns can have the same treatment sequence.
5. You can choose the combinations for the treatments. Random configurations are recommended especially for larger Latin square data layouts.

The program stores the raw data and a numeric code for the treatment in the unnamed data registers, using 2-D vectors. The program also uses the unnamed registers to store the row sums, column sums, and treatment sums.

To use the program you must replace the treatment codes A, B, C, and so forth with the integers 1, 2, 3, and so on, respectively. Thus using the integer treatment codes, the above 3 × 3 Latin square data layout becomes:

	Columns		
Rows	1	2	3
1	Treatment 1 x_{11}	Treatment 2 x_{12}	Treatment 3 x_{13}
2	Treatment 3 x_{21}	Treatment 1 x_{22}	Treatment 2 x_{23}
3	Treatment 2 x_{31}	Treatment 3 x_{32}	Treatment 1 x_{33}

The row, column, and treatment sums:

$$\text{Row } RS_i = \sum_{i=1}^{n} x_{ij}$$

$$\text{Column } CS_j = \sum_{j=1}^{n} x_{ij}$$

$$\text{Treatment } TrS_k = \sum_{k=1}^{n} x_{ij}$$

The sum of squares:

$$TSS = \sum\sum x_{ij}^2 - (\sum\sum x_{ij})^2 / n^2$$

$$\text{Row } RSS = \frac{1}{n} \sum_i (\sum_j x_{ij})^2 - (\sum\sum x_{ij})^2 / n^2$$

$$\text{Column } CSS = \frac{1}{n} \sum_j (\sum_i x_{ij})^2 - (\sum\sum x_{ij})^2 / n^2$$

$$\text{Treatment } TrSS = \frac{1}{n} \sum_j (\sum_k x_{ij})^2 - (\sum\sum x_{ij})^2 / n^2$$

Residual ReSS = TSS – RSS – CSS – TrSS

The degrees of freedom:

Row $df_1 = n - 1$

Column $df_1 = n - 1$

Treatment $df_1 = n - 1$

Error $df_2 = n^2 - 3n + 2$

The mean sum of squares:

$RMS = RSS / df_1$

$CMS = CSS / df_1$

$TrMS = TrSS / df_1$

$ReMS = ReSS / df_2$

The F statistics:

Row effect $F_1 = RMS / ReMS$

Column effect $F_2 = CMS / ReMS$

Treatment effect $F_3 = TrMS / ReMS$

The Latin square ANOVA table is:

	SS	Df	MS	F
Row	RSS	df1	RMS	Row F_1
Column	CSS	df1	CMS	Column F_2
Treatment	TrSS	df1	TrMS	Treatment F_3
Residual	ReSS	df2	ReMS	
Total	TSS			

The calculated F statistics are them compared with the F statistics in standard tables to determine if the set of null hypotheses that the column effect, row effect, and treatment effect are not signficant.

Instructions

The following table shows the steps involved in using the various parts of Latin Square program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	N?	
	Enter the size of the Latin square layout.	Size	R/S	Size squared	
2	Enter the values in the tabulated data, row by row.		XEQ A	X?	
	Enter the value for x.	x	R/S	K?	
	Enter the integer value for the treatment k.	k	R/S	X?	

Step #	Step/Substep	Input	Command	Output X	Output Y
	Enter the next value for x.	x	R/S	K?	
	Enter the next integer value for the treatment k.	K	R/S	X?	
	Repeat the above substeps in this step until you have entered all of values and treatment codes in the data table.				
3	Edit a data point.		XEQ D	I?	
	Enter the row index.	Row	R/S	J?	
	Enter the column index.	Column	R/S	X?	
	Enter the correct value.	x	R/S	K?	
	Enter the treatment code.	k	R/S		
4	To view the data in the matrix x while pausing at each entry.		XEQ V	[x,k]	
5	View the data in the matrix x while stopping at each entry.		XEQ W	[x,k]	
	View the next value.		R/S	[x,k]	
	Repeat the above step to view the rest of the values.				
6	Calculate and display the row sums, column sums, treatment sum, and the entries in the ANOVA table. The program displays the sum of the first row.		XEQ C	Row sum	
	View the next row sum. Repeat this step to view the rest of the row sums. When the program has shown you all of the row sums it displays the message COL SUMS to signal that you are done with the row sums.		R/S	Row sum	
	View the first column sum. Repeat this step to view the rest of the column sums. When the program has shown you all of the column sums it		R/S	Column sum	

Step #	Step/Substep	Input	Command	Output X	Output Y
	displays the message TRT SUMS to signal that you are done with the column sums.				
	View the first column sum. Repeat this step to view the rest of the column sums. When the program has shown you all of the column sums it displays the message TrT SUMS to signal that you are done with the column sums.		R/S	row sum	row index
	View the first treatment sum. Repeat this step to view the rest of the treatment sums. When the program has shown you all of the treatment sums it displays the message TSS, RSS to signal that you are done with the treatment sums.		R/S	Treatment sum	
	View the values of TSS and RSS		R/S	RSS	TSS
	View CSS and TrSS.		R/S	TrSS	CSS
	View RESS and df_1.		R/S	df_1	ReSS
	View df_2 and RMS.		R/S	RMS	df_2
	View df_4 and RMS		R/S	RMS	df_4
	View CMS and TrMS.		R/S	TrMS	CMS
	View ReMS and F_1.		R/S	F_1	ReMS
	View F_2 and F_3.		R/S	F_3	F_2
	The program pauses to display the message END.				
7	To display the program name. The program pauses to display the program name.		XEQ U	NCS-LATIN-SQR1	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add data points	A	TSS	1	Set = Pause, Clear = Stop
C	Calculate ANOVA table elements	B	RSS		
E	View/Edit an observation	C	CSS		
W	View all data points in matrix x	D	TrSS		
V	View all data points in matrix x without stopping the program	E	ReSS, Row index I		
I	Initialize the program	F	df1, Column index J		
U	Display program name	G	df2, Treatment index K		
		H	RMS		
		I	Index to access [x,k] data matrix		
Z	Prepare loop variable to access all of the matrix element	J	Index to access row, column, and treatment summations		
Y	Prepare loop variable to access the row, column, and treatment summations	K	Map key value		
X	Prepare the indices to access each of the row, column, and treatment summations	L			
R	Add a value to a summation	M	n^2		
D	Calculate summation for RSS, CSS, or TrSS	N	N		
G	Get x and map key from 2D value	O	CMS, Row sum access to unnamed memory		
S	Set 2D value from x and map key	P	TrMS, Column sum access to unnamed memory		
		Q	ReMS, Treatment sum access to unnamed memory		
		R	F1		
		S	F2		
		T	F3		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		U			
		V			
		W			
		X	X		
		Y	Used		
		Z	Used		
		0			
		1	[x(1,1),map_key(1,1)]		
		2	[x(1,2),map_key(1,2)]		
		…			
		n	[x(1,n),map_key(1,n)]		
		n+1	[x(2,1),map_key(2,1)]		
		n+2	[x(2,2),map_key(2,2)]		
		…			
		(I-1)n + J	[x(I,J),map_key(I,J)]		
		…			
		n^2	[x(n,n),map_key(n,n)]		
		n^2+1	RowSum(1)		
		n^2+2	RowSum(2)		
		…			
		n^2+n	RowSum(n)		
		n^2+n+1	ColSum(1)		
		n^2+n+2	ColSum(2)		
		…			
		n^2+2n	ColSum(n)		
		n^2+2n+1	TrtSum(1)		
		n^2+2n+2	TrtSum(2)		
		…			
		n^2+3n	TrtSum(n)		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	INPUT N	Enter the size of the design
I003	RCL N	
I004	x^2	
I005	STO M	Store n^2 in M
I006	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-LATIN-	This is a "tag" created using the EQN button

Line	Command	Comments
	SQRS1	
U004	PSE	
U005	CF 10	
U006	RTN	
S001	LBL S	Set 2D value from x and map key
S002	[X,K]	
S003	STO (I)	
S004	RTN	
G001	LBL G	Get x and map key from 2D value
G002	RCL (I)	Get the 2D value in unnamed register I
G003	[1,0]	
G004	×	Get the x value
G005	STO X	
G006	RCL (I)	
G007	[0,1]	
G008	×	Get the key map value
G009	STO K	
G010	RTN	
Z001	LBL Z	Prepare loop variable to access all of the matrix element
Z002	1	
Z003	RCL M	
Z004	1E3	
Z005	÷	
Z006	+	
Z007	STO I	
Z008	RTN	
Y001	LBL Y	Prepare loop variable to access the row, column, and treatment summations
Y002	RCL M	
Y003	1	
Y004	+	
Y005	RCL M	
Y006	RCL N	
Y007	3	
Y008	×	
Y009	+	
Y010	1E3	
Y011	÷	
Y012	+	
Y013	STO J	
Y014	RTN	
X001	LBL X	Prepare the indices to access each of the row, column, and treatment summations
X002	RCL M	

Line	Command	Comments
X003	STO O	Store index to access row sums
X004	RCL N	
X005	+	
X006	STO P	Store index to access column sums
X007	RCL N	
X008	+	
X009	STO Q	Store index to access treatment sums
X010	RTN	
A001	LBL A	Add data points
A002	XEQ Z	Initialize the index to access all of the matrix elements
A003	INPUT X	Start of input loop
A004	INPUT K	
A005	XEQ S	Convert input into a 2D vector and store it in the unnamed memory registers
A006	ISG I	
A007	GTO A003	End of input loop
A008	RTN	
E001	LBL E	View/Edit an observation
E002	INPUT I	
E003	RCL I	
E004	RCL N	
E005	x<y?	Validate input
E006	ACOS(2)	Generate INVALID DATA error message
E007	INPUT J	
E008	RCL J	
E009	RCL N	
E010	x<y?	Validate input
E011	ACOS(2)	Generate INVALID DATA error message
E012	RCL I	
E013	1	
E014	-	
E015	RCL N	
E016	×	
E017	RCL J	
E018	+	
E019	STO I	
E020	XEQ G	Get the data from the unnamed memory registers
E021	INPUT X	
E022	INPUT K	
E023	XEQ S	Store the updated data into the unnamed memory registers
E024	RTN	
V001	LBL V	View all data points in matrix x without stopping the program
V002	SF 1	
V003	XEQ W003	

Line	Command	Comments
V004	CF 1	
V005	RTN	
W001	LBL W	View all data points in matrix x
W002	CF 1	
W003	XEQ Z	Initialize the index to visit all of the matrix elements
W004	XEQ G	Get the data from the unnamed memory registers
W005	RCL K	
W006	RCL X	
W007	FS? 1	
W008	GTO W011	
W009	STOP	
W010	GTO W012	
W011	PSE	
W012	ISG I	
W013	GTO W004	
W014	RTN	
R001	LBL R	Add a value to a summation
R002	+	Add index and offset already in the stack
R003	STO J	
R004	x↔y	Swap index with the value to be added
R005	STO+ (J)	Add value to summation in an unnamed memory register
R006	RTN	
C001	LBL C	Calculate ANOVA table elements
C002	ClΣ	
C003	XEQ Z	Initialize the index to access all of the matrix elements
C004	XEQ G	Get the data from the unnamed memory registers
C005	RCL X	
C006	RCL X	
C007	Σ+	
C008	ISG I	
C009	GTO C004	End of loop
C010	XEQ Y	Initialize index to access all row, column, and treatment summations
C011	1E-499	
C012	STO (J)	Start loop to initialize the summations
C013	ISG J	
C014	GTO C012	End of loop that initializes the summations
C015	XEQ X	Prepare the indices to access each of the row, column, and treatment summations
C016	1	
C017	STO I	Initialize index to access elements of matrix x
C018	STO E	Initialize row counter
C019	1	
C020	STO F	Initialize column counter
C021	XEQ G	Get the next value of x and key map k

Line	Command	Comments
C022	RCL X	
C023	RCL E	
C024	RCL O	
C025	XEQ R	Add x to row summation(I)
C026	RCL X	
C027	RCL F	
C028	RCL P	
C029	XEQ R	Add x to column summation(J)
C030	RCL X	
C031	RCL K	
C032	RCL Q	
C033	XEQ R	Add x to treat÷ment summation(K)
C034	1	
C035	STO+ I	Increment index to access x
C036	STO+ F	Increment row index
C037	RCL F	
C038	RCL N	
C039	x≥y?	
C040	GTO C021	
C041	1	
C042	STO+ E	
C043	RCL E	
C044	RCL N	
C045	x≥y?	
C046	GTO C019	End of outer loop
C047	Σx^2	
C048	Σx	
C049	x^2	
C050	n	
C051	÷	
C052	STO Z	
C053	-	Calculate TSS
C054	STO A	
C055	STO E	ReSS = TSS
C056	XEQ Y	Prepare loop variable to access the row, column, and treatment summations
C057	SF 10	
C058	ROW SUMS	This is a "tag" created using the EQN button
C059	PSE	Display output tag
C060	XEQ D	Calculate RSS
C061	STO B	Store RSS
C062	STO H	RMS = RSS
C063	STO- E	ReSS = TSS - RSS

Line	Command	Comments
C064	COL SUMS	This is a "tag" created using the EQN button
C065	PSE	Display output tag
C066	XEQ D	Calculate CSS
C067	STO C	Store CSS
C068	STO O	CMS = CSS
C069	STO- E	ReSS = TSS - RSS - CSS
C070	TRT SUMS	This is a "tag" created using the EQN button
C071	PSE	Display output tag
C072	XEQ D	Calculate TrSS
C073	STO D	Store TrSS
C074	STO P	TrMS = TrSS
C075	STO- E	ReSS = TSS - RSS - CSS - TrSS
C076	RCL N	
C077	1	Calculate df1
C078	-	
C079	STO F	
C080	STO÷ H	Calculate RMS
C081	STO÷ O	Calculate CMS
C082	STO÷ P	Calculate TrMS
C083	RCL M	
C084	RCL N	
C085	3	
C086	×	
C087	-	
C088	2	
C089	+	Calculate df2
C090	STO G	
C091	1/x	
C092	RCL E	
C093	×	Calculate ReMS
C094	STO Q	
C095	1/x	
C096	STO R	
C097	STO S	
C098	STO T	
C099	RCL H	
C100	STO× R	Calculate F1
C101	RCL O	
C102	STO× S	Calculate F2
C103	RCL P	
C104	STO× T	Calculate F3
C105	TSS,RSS	This is a "tag" created using the EQN button
C106	PSE	Display output tag

Line	Command	Comments
C107	RCL A	
C108	RCL B	
C109	STOP	Display TSS and RSS
C110	CSS,TrSS	This is a "tag" created using the EQN button
C111	PSE	Display output tag
C112	RCL C	
C113	RCL D	
C114	STOP	Display CSS and RSS
C115	RESS,DF1	This is a "tag" created using the EQN button
C116	PSE	
C117	RCL E	Display output tag
C118	RCL F	
C119	STOP	Display ReSS and df1
C120	DF2,RMS	This is a "tag" created using the EQN button
C121	PSE	
C122	RCL G	Display output tag
C123	RCL H	
C124	STOP	Display df2 and RMS
C125	CMS,TRMS	
C126	PSE	This is a "tag" created using the EQN button
C127	RCL O	Display output tag
C128	RCL P	
C129	STOP	Display CMS and TrMs
C130	REMS,F1	This is a "tag" created using the EQN button
C131	PSE	Display output tag
C132	RCL Q	
C133	RCL R	
C134	STOP	Display ReMS, and F1 statistic
C135	F2,F3	This is a "tag" created using the EQN button
C136	PSE	Display output tag
C137	RCL S	
C138	RCL T	
C139	STOP	Display F2 and F3
C140	END	This is a "tag" created using the EQN button
C141	PSE	Display output tag
C142	CF 10	
C143	RTN	
D001	LBL D	Calculate summation for RSS, CSS, or TrSS
D002	0	
D003	STO Y	Initialize a sum of squares
D004	1	
D005	RCL N	
D006	1E3	
D007	÷	

Line	Command	Comments
D008	+	
D009	STO F	Initialize loop control variable
D010	RCL Y	Start loop to update first double summation
D011	IP	
D012	RCL (J)	
D013	STOP	View the index of the summation and the summation
D014	x^2	
D015	STO+ Y	
D016	1	
D017	STO+ J	
D018	ISG F	
D019	GTO D010	End of loop
D020	RCL Y	
D021	RCL N	
D022	÷	
D023	RCL Z	
D024	-	Calculate a sum of squares
D025	RTN	

Example

Let's look at an example that exercises the various routines in the program. The next table contains set of observations. Determine the relevance of the row and column treatments at a 95% confidence level:

	Columns		
Rows	1	2	3
1	Treatment 1 0.194	Treatment 2 0.730	Treatment 3 1.187
2	Treatment 3 0.758	Treatment 1 0.311	Treatment 2 0.589
3	Treatment 2 0.369	Treatment 3 0.558	Treatment 1 0.311

After you enter the data point, calculate, display the elements of the ANOVA table, and examine the values of the row and column F statistics.

The following table shows the steps performed using the FIX 2 display mode. As mentioned earlier, the program briefly switches to the FIX 0 display when it displays the indices for the row and column entry.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	N?
	Enter the Latin square layout size.	3	R/S	9.00
2	Enter the tabulated data going across each row.		XEQ A	X?
	Enter the first value in the data table, in row 1 and column 1.	0.194	R/S	K?
	Enter the integer layout code for row 1 and column 1.	1	R/S	X?
	Enter the next value for x.	0.73	R/S	K?
	Enter the next integer layout code.	2	R/S	X?
	Repeat the above steps to enter the remaining values in the table.			
3	Calculate and display the row sums, column sums, treatment sums, and the entries in the ANOVA table. The program pauses to display ROW SUMS and then displays the first row sum.		XEQ C	0.00 2.11 Displays the sum of row 1
	View the second row sum.		R/S	4.00 1.66 Displays the sum of row 2
	View the third row sum.		R/S	7.00 1.24 Displays the sum of row 3

Step #	Step/Substep	Input	Command	Output
	The program pauses to display the message COL SUMS and then proceed to display the first column sum.		R/S	0.00 1.32 Displays the sum of column 1
	View the second column sum.		R/S	1.00 1.60 Displays the sum of column 2
	View the third column sum.		R/S	4.00 2.09 Displays the sum of column 3
	The program pauses to display the message TrT SUMS and then proceed to display the first treatment sum.		R/S	0.00 0.82 Displays the sum of treatment 1
	View the second treatment sum		R/S	0.00 1.69 Displays the sum of treatment 2
	View the third treatment sum		R/S	3.00 2.50 Display the column index and sum for column 1
	View the values of TSS and RSS.		R/S	0.76 0.13
	View CSS and TrSS.		R/S	0.10 0.47
	View RESS and df_1.		R/S	0.05 2.00
	View df_2 and RMS.		R/S	2.00 0.06
	View CMS and TrMS.		R/S	0.05 0.24

Step #	Step/Substep	Input	Command	Output
	View REMS and F_1.		R/S	0.03 2.33
	View F_2 and F_3.		R/S	1.84 8.70
	The program pauses to display the message END.			

The Latin square ANOVA table for the fixed row and fixed column effects is:

	SS	df	MS	F
Row	0.13	2	0.06	2.33
Column	0.10	2	0.05	1.84
Treatment	0.47	2	0.24	8.70
Residual	0.05	2	0.03	
Total	0.76			

The next table summarizes the calculated F statistics and compares them with values found in standard tables. The conclusion draw, at a 95% confidence level, is that we cannot reject the set of null hypotheses stating that the row, column, and treatment effects are not significant.

	F Calculated	df1	df2	F Tabulated
Row	2.33	2	2	19.00
Column	1.84	2	2	19.00
Interaction	8.70	2	2	19.00

Appendix A contains a new approximation that I recently developed for inverse F function. The appendix also contains tables for selected values.

Linear Regression Program NCS_LR1

Overview

The NCS_LR1 program implements linear regression and offers the following features:

1. Storing data points once and reusing them for different regression scenarios. The program stores the (x, y) data points as 2-D vectors, a new feature of the HP-35s, thus reducing the number of unnamed memory registers used by half.
2. The ability to delete, edit, view, and swap data points. Thus the regression program incorporates basic operations to manage the data points.
3. Transforming the x and y observations. You can edit the program to implement your own transformations and then recalculate the regression statistics without having to reenter the data points.
4. Calculating the regression intercept, slope, and coefficient of determination, using the built-in calculator feature.
5. Calculating the confidence intervals for intercept, slope, and coefficient of determination.
6. Calculating the regression ANOVA table.
7. Calculating projected value of y for existing or new values of x.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Data Collected

$\sum x$ = sum of x
$\sum x^2$ = sum of x^2
$\sum y^2$ = sum of y
$\sum y^2$ = sum of y^2
$\sum xy$ = sum of $x \cdot y$
n = number of observations

Regression Coefficients

$x_m = \sum x / n$
$y_m = \sum y / n$
$S_{xx} = \sum x^2 - (\sum x)^2 / n = \sum x^2 - n (x_m)^2$
$S_{yy} = \sum y^2 - (\sum y)^2 / n = \sum y^2 - n (y_m)^2$
$S_{xy} = \sum xy - (\sum x)(\sum y)/ n = \sum xy - n\, x_m\, y_m$
Slope $B = S_{xy} / S_{xx} = (\sum xy - n\, x_m\, y_m) / (\sum x^2 - n (x_m)^2)$
Intercept $A = y_m - B\, x_m$
For line: $y = A + B x$

ANOVA Table

Source of Variation	*Sum of Squares*	*Degrees of Freedom*	*Mean Square*	*F*
Regression	$SS_R = B\ S_{xy}$	1	MS_R	MS_R / MS_E
Residual	$SS_E = S_{yy} - B\ S_{xy}$	n – 2	MS_E	
Total	$SS_T = S_{yy}$	n – 1		

Interval for Slope

At 100(1 – α) confidence:
$SlopeStdErr = \sqrt{(MS_E / S_{xx})}$
$\Delta B = t_{\alpha/2,n-2} \cdot SlopeStdErr$
Upper Range: $B + \Delta B$
Lower Range: $B - \Delta B$

Interval for Intercept

At 100(1 – α) confidence:
$IntStdErr = \sqrt{(MS_E\ [1/n + (x_m)^2 / S_{xx}])}$
$\Delta A = t_{\alpha/2,n-2} \cdot IntStdErr$
Upper Range: $A + \Delta A$
Lower Range: $A - \Delta A$

Interval for Projected Y

At 100(1 – α) confidence:
$\hat{y} = A + B\ x_0$
$YhatStdErr = \sqrt{(MS_E\ [1/n + (x_0 - x_m)^2 / S_{xx}])}$
$\Delta\hat{y} = t_{\alpha/2,n-2} \cdot YhatStdErr$
Upper Range: $\hat{y} + \Delta\hat{y}$
Lower Range: $\hat{y} - \Delta\hat{y}$

For a *new* observation:
$YhatStdErr = t_{\alpha/2,n-2} \cdot \sqrt{(MS_E\ [1 + 1/n + (x_0 - x_m)^2 / S_{xx}])}$
$\Delta\hat{y} = t_{\alpha/2,n-2} \cdot YhatStdErr$
Upper Range: $\hat{y} + \Delta\hat{y}$
Lower Range: $\hat{y} - \Delta\hat{y}$

Coefficient of Determination

$R^2 = r^2 = B\ (S_{xy} / S_{yy})$
At 100(1 – α) confidence:
Let $P_1 = \tanh^{-1}(r)$ and $P_2 = Z_{\alpha/2} / \sqrt{(n - 3)}$
Upper Range for $r^2 = [\tanh(P_1 + P_2)]^2$
Lower Range for $r^2 = [\tanh(P_1 - P_2)]^2$

Instructions

The following table shows the steps involved in using the various parts of the linear regression program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value for x.	x	R/S	Y?		
	Enter a value for y.	y	R/S	Number of data points		
3	Delete a data point.		XEQ D	I?		
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message INVALID DATA.	Index of data point to delete	R/S	Number of data points		
4	Edit a data point.		XEQ E	I?		
	Enter the index of the data point to edit. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	X?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	Enter a new value for x.	x	R/S	Y?		
	Enter a new value for y.	y	R/S			
5	View a data point.		XEQ V	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	Data point	Index	
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	Data point	Index	
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1		
6	View all data points (Note the program displays the results using the PSE statement.) Note: If you spot an error in the data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	Data point	Index	
7	Swap data points.		XEQ S	I?		
	Enter the index of	Index of the	R/S	J?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	the first data point. If you enter an invalid index the program displays the message INVALID DATA.	first data point				
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S			
8	Calculate the regression coefficients.		XEQ C	slope	Intercept	r^2
9	Calculate the regression ANOVA table and then display the components of the table.		XEQ N	Regression DF	Regression sum of squares (SSR)	
	View the value for MSR.		R/S	Mean Square for regression (MSR)		
	View the F statistic.		R/S	Calculated F statistic		
	View the values for SSE and the residual degrees of freedom.		R/S	Residual df	Residual sum of squares (SSE)	
	View the value of MSE.		R/S	Mean Square for residuals (MSE)		
	View the total sum of squares and the total degrees of freedom,		R/S	Total df	Total Sum of Squares	
10	Calculate the confidence intervals.		XEQ F	T?		
	Enter a value for the inverse	Student-t for n-3 degrees	R/S	Intercept lower limit	Interval upper limit	

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	Student-t probability distribution function.	of freedom and α/2 significance level (α is based on 1-α confidence level.)				
			R/S	Slope lower limit	Slope upper limit	
			R/S	Z?		
	Enter a value for the inverse normal probability distribution function.	Z value for α/2 significance level (α is based on 1–α confidence level.)	R/S	r^2 lower limit	r^2 upper limit	
11	Project an existing x onto y.		XEQ P	T?		
	Enter a value for the inverse Student-t probability distribution function.	Student-t for n–3 degrees of freedom and α/2 significance level (α is based on 1–α confidence level.)	R/S	X?		
		x	R/S	$\hat{y}$ lower limit	$\hat{y}$	$\hat{y}$ upper limit
12	Project a new x onto y.		XEQ Q	T?		
	Enter a value for the inverse Student-t probability distribution function.	Student-t for n–3 degrees of freedom and α/2 significance level (α is based on 1–α	R/S	X?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
		confidence level.)				
		x	R/S	$\hat{y}$ lower limit	$\hat{y}$	$\hat{y}$ upper limit
13	You can temporarily (or even permanently) reduce the number of data points entering in the regression calculations.		RCL C	Number of data points		
	Record the number of original data points.					
	Enter a lower value for the number of data points (must be greater than 3).	New number of data points	STO C			
14	You can restore the original number of data points entering in the regression calculations.	Original number of data points	STO C			
15	Alter the transformation for x.		GTO X001			
	Switch to program mode.		PRGM	X001 LBL X		
	Edit the commands between LBL X and the RTN statement to reflect the new transformation you desire.					
	Exit program mode when you are done.		PRGM			
16	Alter the transformation for		GTO Y001			

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	y.					
	Switch to program mode.		PRGM	Y001 LBL Y		
	Edit the commands between LBL Y and the RTN statement to reflect the new transformation you desire.					
	Exit program mode when you are done.		PRGM			
17	Alter the inverse transformation for y.		GTO Z001			
	Switch to program mode.		PRGM	Z001 LBL Z		
	Edit the commands between LBL Z and the RTN statement to reflect the new inverse transformation you desire.					
	Exit program mode when you are done.		PRGM			
18	To display the program name. The program pauses to display the program name.		XEQ U	NCS-LR1		

Comments on Runtime Errors

The program validates the indices that you enter to manage data points to prevent the corruption of the data points. Aside from this feature, you are responsible to enter data that will avoid runtime errors. For example, if you are using the square root or logarithms to transform the x and/or y data, then entering data points with negative values will raise runtime errors. Likewise, using reciprocals and logarithms will generate runtime errors for data that are zeros. Entering redundant data points also has its peril of giving you meaningless results. Remember the old saying about program input and output data "Junk in, junk out."

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	C	Number of data points n	1	Index invalid when set
D	Delete an existing data point	S	Sxx	2	Project a new x onto y when set
E	Edit a data point	U	Syy		
V	View a data point	V	Sxy		
W	View all data points	M	MSE (Mean square error)		
I	Initialize	T	Inverse Student-t pdf		
S	Swap two data points	Z	Inverse normal pdf		
C	Calculate regression coefficients	X	Used		
N	Calculate regression ANOVA	Y	Used		
F	Calculate confidence intervals	I	Index, used		
P	Project an observed x onto y with confidence interval	J	Index, used		
Q	Project a new x onto y with confidence interval	K	Used		
X	Transform x data	1	Data point #1		
Y	Transform y data	2	Data point #2		
Z	Inverse transform of y data	3	Data point #3		
U	Display program name	…			
Labels for internal use		n	Data point #n		
H	Validate indices and display error message if an index is invalid				
G	Prompt for x and y values				
R	Display the message INVALID DATA				

Program Listing

Line	Command	Comments
X001	LBL X	Label to contain the transformation for x. No code is needed, between the label and the RTN command, for linear values.
X002	RTN	
Y001	LBL Y	Label to contain the transformation for y. No code is needed, between the label and the RTN command, for linear values.
Y002	RTN	
Z001	LBL Z	Label to contain the inverse transformation for z. No code is needed, between the label and the RTN command, for linear values.
Z002	RTN	
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	

Line	Command	Comments
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO– C	
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	Validate index
S006	RCL (I)	Recall data points and place them on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x↔y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	FS? 1	Index out or range?
V005	GTO R001	
V006	RCL (I)	Recall data point
V007	STOP	
V008	1	
V009	RCL I	
V010	+	
V011	STO I	
V012	GTO V003	
W001	LBL W	View all data points

Line	Command	Comments
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate regression coefficients
C002	1	
C003	RCL C	
C004	1E3	
C005	÷	
C006	+	
C007	STO I	Store loop index
C008	ClΣ	Clear statistical registers
C009	RCL (I)	Start the loop
C010	[1,0]	
C011	×	Get x
C012	XEQ X	Transform value of x
C013	STO X	Store transformed x
C014	RCL (I)	
C015	[0,1]	
C016	×	Get y
C017	XEQ Y	Transform value of y
C018	RCL X	
C019	Σ+	Accumulate (x, y) in statistical registers
C020	ISG I	
C021	GTO C009	Resume the next loop iteration
C022	Σx^2	Calculate Sxx
C023	Σx	
C024	x^2	
C025	n	
C026	÷	
C027	–	
C028	STO S	Store Sxx
C029	Σy^2	Calculate Syy
C030	Σy	
C031	x^2	

Line	Command	Comments
C032	n	
C033	÷	
C034	–	
C035	STO U	Store Syy
C036	Σxy	Calculate Sxy
C037	Σx	
C038	Σy	
C039	×	
C040	n	
C041	÷	
C042	–	
C043	STO V	Store Sxy
C044	m	Calculate MSE
C045	×	
C046	+/–	
C047	RCL U	
C048	+	
C049	n	
C050	2	
C051	–	
C052	÷	
C053	STO M	Store MSE
C054	r	Push r^2, intercept, and slope in the stack
C055	x^2	
C056	b	
C057	m	
C058	SF 10	Set EQN display flag mode
C059	RSQR, INTR, SLP	This is a "tag" created using EQN button
C060	PSE	Display the output tag for the r^2, intercept, and slope
C061	CF10	
C062	RTN	Display the values for for the r^2, intercept, and slope
N001	LBL N	Display components of ANOVA table
N002	m	Calculate regression sum of squares (SSR)
N003	RCL V	
N004	×	
N005	1	Push DF ins tack
N006	SF10	
N007	SSR, DF	This is a "tag" created using EQN button
N008	PSE	Display output tag for SSR and DF
N009	STOP	Display numerical value(s) associated with the above tag
N010	÷	Calculate mean square for regression
N011	MSR	This is a "tag" created using EQN button

Line	Command	Comments
N012	PSE	Display output tag for MSR
N013	STOP	Display numerical value(s) associated with the above tag
N014	RCL M	
N015	÷	Calculate F statistic
N016	F STAT	This is a "tag" created using EQN button
N017	PSE	Display output tag for the F statistic
N018	STOP	Display numerical value(s) associated with the above tag
N019	RCL U	
N020	m	
N021	RCL V	
N022	×	
N023	–	Calculate residual sum of squares (SSE)
N024	n	
N025	2	
N026	–	Calculate degrees of freedom for SSE
N027	SSE, DF	This is a "tag" created using EQN button
N028	PSE	Display the output tag for SSE and its degrees of freedom
N029	STOP	Display numerical value(s) associated with the above tag
N030	÷	Calculate mean square for residual (MSE)
N031	MSE	This is a "tag" created using EQN button
N032	PSE	Display the output tag for MSE
N033	STOP	Display numerical value(s) associated with the above tag
N034	RCL U	Get total sum of squares
N035	n	
N036	1	
N037	–	Calculate df for total sum of squares
N038	TSS, DF	This is a "tag" created using EQN button
N039	PSE	Display output tag for TSS and its degrees of freedom
N040	CF 10	
N041	RTN	Display numerical value(s) associated with the above tag
F001	LBL F	Confidence intervals for intercept, slope, and r^2
F002	INPUT T	Get inverse Student-t
F003	$\bar{x}$	
F004	x^2	
F005	RCL S	
F006	÷	
F007	n	
F008	1/x	
F009	+	
F010	RCL M	
F011	×	
F012	√x	Calculate standard error for intercept
F013	RCL T	
F014	×	Calculate confidence interval / 2

Line	Command	Comments
F015	STO X	
F016	b	
F017	+	
F018	b	
F019	RCL X	
F020	–	
F021	SF 10	
F022	INT UL,LL	This is a “tag” created using EQN button
F023	PSE	Display output tag for the confidence interval of the intercept
F024	STOP	Display confidence interval for intercept
F025	RCL M	
F026	RCL S	
F027	÷	
F028	√x	Calculate standard error for slope
F029	RCL T	
F030	×	Calculate confidence interval / 2
F031	STO X	
F032	m	
F033	+	
F034	m	
F035	RCL X	
F036	–	
F037	SLP UL, LL	This is a “tag” created using EQN button
F038	PSE	Display output tag for the confidence interval of the slope
F039	STOP	Display confidence interval for slope
F040	INPUT Z	Get inverse normal pdf
F041	r	
F042	ATANH	
F043	STO X	
F044	RCL Z	
F045	n	
F046	3	
F047	–	
F048	√x	
F049	÷	
F050	STO Y	
F051	+	
F052	TANH	
F053	x^2	Calculate upper limit for r^2
F054	RCL X	
F055	RCL Y	
F056	–	
F057	TANH	
F058	x^2	Calculate lower limit for r^2

Line	Command	Comments
F059	x↔y	
F060	RSQ UL, LL	This is a “tag” created using EQN button
F061	PSE	Display output tag for the confidence interval of the slope
F062	CF 10	
F063	RTN	Display confidence interval for r^2
Q001	LBL Q	Project a new x onto y
Q002	SF 2	
Q003	GTO P003	
P001	LBL P	Project an existing x onto y
P002	CF 2	
P003	INPUT T	Get inverse Student-t
P004	INPUT X	Get value for x
P005	XEQ X	Transform x
P006	STO X	
P007	$\hat{y}$	Calculate projected y
P008	STO Y	
P009	RCL X	
P010	$\bar{x}$	
P011	–	
P012	x^2	
P013	RCL S	
P014	÷	
P015	n	
P016	1/x	
P017	+	
P018	FS? 2	Use flag 2 to add one if x is a new value
P019	1	
P020	FS? 2	
P021	+	
P022	CF 2	
P023	RCL M	
P024	×	
P025	√x	Calculate standard error in y hat
P026	RCL T	
P027	×	
P028	STO K	
P029	+/–	
P030	RCL Y	
P031	+	Calculate lower limit for $\hat{y}$
P032	XEQ Z	Inverse transform $\hat{y}$ value
P033	STO I	
P034	RCL Y	
P035	XEQ Z	Inverse transform $\hat{y}$ value

Line	Command	Comments
P036	STO J	
P037	RCL Y	
P038	RCL K	
P039	+	Calculate upper limit for $\hat{y}$
P040	XEQ Z	Inverse transform $\hat{y}$ value
P041	RCL J	Get $\hat{y}$
P042	RCL I	Get lower limit $\hat{y}$
P043	SF 10	
P044	Y^U,Y^,Y^L	This is a "tag" created using EQN button
P045	PSE	Display output tag for $\hat{y}$ upper limit, $\hat{y}$, and $\hat{y}$ lower limit
P046	CF 10	
P047	RTN	Display $\hat{y}$ upper limit, $\hat{y}$, and $\hat{y}$ lower limit
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT Y	Prompt user for y value
G004	[X,Y]	Create data point. Enter this statement using the EQN key.
G005	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	Enter this statement using the EQN key.
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-LR1	This is a "tag" created using EQN button
U004	PSE	Display program name
U005	CF 10	
U006	RTN	

Example

Let's look at an example that exercises various routines in the regression program. The next table contains (x, y) data points that we want to fit in a simple linear regression model.

x	y
2	24.3
4	19.7
6	17.8
8	14.0
10	12.3
12	7.2
14	5.5

After you enter the above data, perform the following steps:

1. View all of the data points.
2. Calculate the regression slope, intercept, and r^2.
3. Calculate the regression ANOVA table.
4. Calculate confidence interval for the intercept, slope, and r^2 at a 95% confidence level.
5. Calculate the projected value of y and its confidence interval for x = 8 at 95% confidence.

The following table shows the steps using the FIX 3 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0.000
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	2	R/S	Y?
	Enter a value for y.	24.3	R/S	1.000
3	Add the second data point.		XEQ A	X?
	Enter a value for x.	4	R/S	Y?
	Enter a value for y.	19.7	R/S	2.000
4	Repeat step 3 for all other data point.			7.000
5	View all data points (Note the		XEQ W	7.007 [14.000,5.500]

Step #	Step/Substep	Input	Command	Output
	program displays the results using the PSE statement). If you spot an error use xEQ E to overwrite the erroneous data point with the correct one.			
6	Calculate the regression coefficients.		XEQ C	26.814 -1.552 Displays the intercept and slope
			R↓	0.987 26.814 Displays the values for r^2 and the intercept
	The fitted line is: y = 26.814 – 1.552 x with an r^2 of 0.987.			
7	Calculate the regression ANOVA table.		XEQ N	269.700 1.000 Shows the Regression sum of squares (SSR) and the Regression DF
			R/S	26.814 269.700 Shows the Mean Square for regression (MSR)
			R/S	26.814 376.714 Shows the calculated F statistic
			R/S	3.580 5.000 Shows the Residual sum of squares (SSE) and the Residual

Step #	Step/Substep	Input	Command	Output
				df
			R/S	376.714 0.716 Shows the Mean Square for residuals (MSE)
			R/S	273.280 6.000 Shows the Total Sum of Squares and the Total df
8	Calculate confidence intervals.		XEQ F	T?
	Enter the value for the inverse Student-t for 5 DF and a 0.025 probability.	2.571	R/S	28.653 24.976 Shows the Intercept confidence interval
			R/S	⁻1.346 ⁻1.757 Shows the confidence interval for the slope
			R/S	Z?
	Enter the value for the inverse Normal for a 0.025 probability.	1.960	R/S	0.998 0.911 Shows the confidence interval for r^2
9	Project an existing x onto y.		XEQ P	T?
	Enter the value for the inverse Student-t for 5 DF and a 0.025 probability.	2.571	R/S	X?
	Enter value for X.	8	R/S	14.400 13.578 Shows $\hat{y}$ and $\hat{y}$ lower limit
			R↓	15.222 14.400

Step #	Step/Substep	Input	Command	Output
				Shows $\hat{y}$ upper limit and $\hat{y}$

If you desire to apply data transformations to the x and/or y observations you need to edit the labels X, Y, and Z. For example, if you wish to use a power fit y = ax^b, then you need to have the labels X, Y, and Z a follows:

X001 LBL X
X002 LN
X003 RTN
Y001 LBL Y
Y002 LN
Y003 RTN
Z001 LBL Z
Z002 Exp
Z003 RTN

As another example, if you wish to use a power fit 1/y = a + b/x, then you need to have the labels X, Y, and Z a follows:

X001 LBL X
X002 1/x
X003 RTN
Y001 LBL Y
Y002 1/x
Y003 RTN
Z001 LBL Z
Z002 1/x
Z003 RTN

Notice that the code in labels Y and Z is the same because the reciprocal function is the reciprocal of itself. Thus, one can rewrite the above code in a slightly more compact form taking advantage of the specific case:

X001 LBL X
Y001 LBL Y
Z001 LBL Z
Z002 1/x
Z003 RTN

Appendix A contains new approximations that I recently developed for inverse normal and Student-t functions. The appendix also contains tables for selected values.

Weighted Linear Regression Program NCS_LR2

Overview

The NCS_LR2 program implements weighted linear regression and offers the following features:

1. Storing data points once and reusing them for different regression scenarios. The program stores the (x, y, w) data points as 3-D vectors, a new feature of the HP-35s, thus reducing the number of unnamed memory registers used by a third. Each data point has values for the x and y observations and the weight, w, associated with that observation. When you assign 1 to all the weights, the program performs an ordinary linear(ized) regression, just like program NCS_LR1.
2. The ability to delete, edit, view, and swap data points. Thus the regression program incorporates basic operations to manage the data points.
3. Transforming the x and y observations. You can edit the program to implement your custom transformations and then recalculate the regression statistics without having to reenter the data points.
4. Calculating the regression intercept, slope, and coefficient of determination, using the built-in calculator feature.
5. Calculating the confidence intervals for intercept, slope, and coefficient of determination.
6. Calculating the regression ANOVA table.
7. Calculating projected value of y for existing or new values of x.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Data Collected

$\sum wx$ = sum of weight times x
$\sum wx^2$ = sum of weight times x^2
$\sum wy^2$ = sum of weight times y
$\sum wy^2$ = sum of weight times y^2
$\sum wxy$ = sum of weight times $x \cdot y$
$n = \sum w$

Regression Coefficients

$x_m = \sum wx / n$
$y_m = \sum wy / n$
$S_{xx} = \sum wx^2 - (\sum wx)^2 / n = \sum wx^2 - n (x_m)^2$
$S_{yy} = \sum wy^2 - (\sum wy)^2 / n = \sum wy^2 - n (y_m)^2$
$S_{xy} = \sum wxy - (\sum wx)(\sum wy)/ n = \sum wxy - n\, x_m\, y_m$
Slope $B = S_{xy} / S_{xx} = (\sum wxy - n\, x_m\, y_m) / (\sum wx^2 - n (x_m)^2)$
Intercept $A = y_m - B\, x_m$
For line: $y = A + B x$

ANOVA Table

Source of Variation	*Sum of Squares*	*Degrees of Freedom*	*Mean Square*	*F*
Regression	$SS_R = B\ S_{xy}$	1	MS_R	MS_R / MS_E
Residual	$SS_E = S_{yy} - B\ S_{xy}$	n – 2	MS_E	
Total	$SS_T = S_{yy}$	n – 1		

Interval for Slope

At 100(1 – α) confidence:
$SlopeStdErr = \sqrt{(MS_E / S_{xx})}$
$\Delta B = t_{\alpha/2,n-2} \cdot SlopeStdErr$
Upper Range: $B + \Delta B$
Lower Range: $B - \Delta B$

Interval for Intercept

At 100(1 – α) confidence:
$IntStdErr = \sqrt{(MS_E [1/n + (x_m)^2 / S_{xx}])}$
$\Delta A = t_{\alpha/2,n-2} \cdot IntStdErr$
Upper Range: $A + \Delta A$
Lower Range: $A - \Delta A$

Interval for Projected Y

At 100(1 – α) confidence:
$\hat{y} = A + B\ x_0$
$YhatStdErr = \sqrt{(MS_E [1/n + (x_0 - x_m)^2 / S_{xx}])}$
$\Delta\hat{y} = t_{\alpha/2,n-2} \cdot YhatStdErr$
Upper Range: $\hat{y} + \Delta\hat{y}$
Lower Range: $\hat{y} - \Delta\hat{y}$

For a *new* observation:
$YhatStdErr = t_{\alpha/2,n-2} \cdot \sqrt{(MS_E [1 + 1/n + (x_0 - x_m)^2 / S_{xx}])}$
$\Delta\hat{y} = t_{\alpha/2,n-2} \cdot YhatStdErr$
Upper Range: $\hat{y} + \Delta\hat{y}$
Lower Range: $\hat{y} - \Delta\hat{y}$

Coefficient of Determination

$R^2 = r^2 = B\ (S_{xy} / S_{yy})$
At 100(1 – α) confidence:
Let $P_1 = \tanh^{-1}(r)$ and $P_2 = Z_{\alpha/2} / \sqrt{(n - 3)}$
Upper Range for $r^2 = [\tanh(P_1 + P_2)]^2$
Lower Range for $r^2 = [\tanh(P_1 - P_2)]^2$

Instructions

The following table shows the steps involved in using the various parts of the linear regression program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value for x.	x	R/S	Y?		
	Enter a value for y.	y	R/S	W?		
	Enter a value for the weight,	w	R/S	Number of data points		
3	Delete a data point.		XEQ D	I?		
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message INVALID DATA.	Index of data point to delete	R/S	Number of data points		
4	Edit a data point.		XEQ E	I?		
	Enter the index of the data point to edit. If you enter an invalid index the program	Index	R/S	X?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	displays the message INVALID DATA.					
	Enter a new value for x.	x	R/S	Y?		
	Enter a new value for y.	y	R/S	W?		
	Enter a value for the weight.	w	R/S			
5	View a data point.		XEQ V	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	Data point	Index	
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	Data point	Index	
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1		
6	View all data points (Note the program displays the results using the PSE statement.) Note: If you spot an error in the		XEQ W	Data point	Index	

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.					
7	Swap data points.		XEQ S	I?		
	Enter the index of the first data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the first data point	R/S	J?		
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S			
8	Calculate the regression coefficients.		XEQ C	Slope	Intercept	r^2
9	Calculate the regression ANOVA table and then display the components of the table.		XEQ N	Regression DF	Regression sum of squares (SSR)	
	View the value for MSR.		R/S	Mean Square for regression (MSR)		
	View the F statistic.		R/S	Calculated F statistic		
	View the values for SSE and the residual degrees		R/S	Residual df	Residual sum of squares	

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	of freedom.				(SSE)	
	View the value of MSE.		R/S	Mean Square for residuals (MSE)		
	View the total sum of squares and the total degrees of freedom,		R/S	Total df	Total Sum of Squares	
10	Calculate the confidence intervals.		XEQ F	T?		
	Enter a value for the inverse Student-t probability distribution function.	Student-t for n-3 degrees of freedom and $\alpha/2$ significance level (α is based on $100(1–\alpha)\%$ confidence level.)	R/S	Intercept lower limit	Interval upper limit	
			R/S	Slope lower limit	Slope upper limit	
			R/S	Z?		
	Enter a value for the inverse normal probability distribution function.	Z value for $\alpha/2$ significance level (α is based on $100(1–\alpha)\%$ confidence level.)	R/S	r^2 lower limit	r^2 upper limit	
11	Project an existing value of x onto y.		XEQ P	T?		
	Enter a value for the inverse Student-t probability distribution function.	Student-t for n–3 degrees of freedom and $\alpha/2$ significance level (α is	R/S	X?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
		based on 100(1–α)% confidence level.)				
		x	R/S	$\hat{y}$ lower limit	$\hat{y}$	$\hat{y}$ upper limit
12	Project a new value of x onto y.		XEQ Q	T?		
	Enter a value for the inverse Student-t probability distribution function.	Student-t for n–3 degrees of freedom and α/2 significance level (α is based on 100(1–α)% confidence level.)	R/S	X?		
		x	R/S	$\hat{y}$ lower limit	$\hat{y}$	$\hat{y}$ upper limit
13	You can temporarily (or even permanently) reduce the number of data points entering in the regression calculations.		RCL C	Number of data points		
	Record the number of original data points.					
	Enter a lower value for the number of data points (must be greater than 3).	New number of data points	STO C			
14	You can restore the original number of data points entering in the regression	Original number of data points	STO C			

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	calculations.					
15	Alter the transformation for x.		GTO X001			
	Switch to program mode.		PRGM	X001 LBL X		
	Edit the commands between LBL X and the RTN statement to reflect the new transformation you desire.					
	Exit program mode when you are done.		PRGM			
16	Alter the transformation for y.		GTO Y001			
	Switch to program mode.		PRGM	Y001 LBL Y		
	Edit the commands between LBL Y and the RTN statement to reflect the new transformation you desire.					
	Exit program mode when you are done.		PRGM			
17	Alter the inverse transformation for y.		GTO Z001			
	Switch to program mode.		PRGM	Z001 LBL Z		
	Edit the commands between LBL Z and the RTN statement to reflect the new					

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	inverse transformation you desire.					
	Exit program mode when you are done.		PRGM			
18	To display the program name. The program pauses to display the program name.		XEQ U	NCS-LR2		

Comments on Runtime Errors

The program validates the indices that you enter to manage data points to prevent the corruption of the data points. Aside from this feature, you are responsible to enter data that will avoid runtime errors. For example, if you are using the square root or logarithms to transform the x and/or y data, then entering data points with negative values will raise runtime errors. Likewise, using reciprocals and logarithms will generate runtime errors for data that are zeros. Entering redundant data points also has its peril of giving you meaningless results and possiblt runtime error. Remember the old saying about program input and output data "Junk in, junk out."

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A		1	Index invalid when set
D	Delete an existing data point	B		2	Project a new x onto y when set
E	Edit a data point	C	Number of data points n		
V	View a data point	D			
W	View all data points	E			
I	Initialize	F			
S	Swap two data points	G			
C	Calculate regression coefficients	H			
N	Calculate regression ANOVA	I	Index		
F	Calculate confidence intervals	J	Index		
P	Project an observed	K	Used		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
	x onto y with confidence interval				
Q	Project a new x onto y with confidence interval	L			
X	Transform x data	M	MSE (Mean square error)		
Y	Transform y data	N			
Z	Inverse transform of y data	O			
U	Display program name	P			
Labels for internal use		Q			
H	Validate indices and display error message if an index is invalid	R			
G	Prompt for x and y values	S	Sxx		
R	Display the message INVALID DATA	T	Inverse Student-t pdf		
B	Emulates Σ+for weighted data	U	Syy		
		V	Sxy		
		W	w		
		X	x		
		Y	y		
		Z	Inverse normal pdf		
		1	[x, y, w] for point 1		
		2	[x, y, w] for point 2		
		…			
		n	[x, y, w] for point # n		

Program Listing

Line	Command	Comments
X001	LBL X	Label to contain the transformation for x. No code is needed, between the label and the RTN command, for linear values.
X002	RTN	
Y001	LBL Y	Label to contain the transformation for y. No code is needed, between the label and the RTN command, for linear values.
Y002	RTN	
Z001	LBL Z	Label to contain the inverse transformation for z. No code is needed, between the label and the RTN command, for linear values.

Line	Command	Comments
Z002	RTN	
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO– C	
D022	RCL C	
D023	RTN	

Line	Command	Comments
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	Validate index
S006	RCL (I)	Recall data points and place them on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x↔y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	FS? 1	Index out or range?
V005	GTO R001	
V006	RCL (I)	Recall data point
V007	STOP	
V008	1	
V009	RCL I	
V010	+	
V011	STO I	
V012	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points
W009	RCL (I)	
W010	PSE	

Line	Command	Comments
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate regression coefficients
C002	1	
C003	RCL C	
C004	1E3	
C005	÷	
C006	+	
C007	STO I	Store loop index
C008	ClΣ	Clear statistical registers
C009	RCL (I)	Start the loop
C010	[1,0,0]	
C011	×	Get x
C012	XEQ X	Transform value of x
C013	STO X	Store transformed x
C014	RCL (I)	
C015	[0,1,0]	
C016	×	Get y
C017	XEQ Y	Transform value of y
C018	STO Y	
C019	RCL (I)	
C020	[0,0,1]	
C021	×	Get w
C022	STO W	
C023	XEQ B	Accumulate (x, y) in statistical registers
C024	ISG I	
C025	GTO C009	Resume the next loop iteration
C026	Σx^2	Calculate Sxx
C027	Σx	
C028	x^2	
C029	n	
C030	÷	
C031	–	
C032	STO S	Store Sxx
C033	Σy^2	Calculate Syy
C034	Σy	
C035	x^2	
C036	n	
C037	÷	
C038	–	
C039	STO U	Store Syy
C040	Σxy	Calculate Sxy

Line	Command	Comments
C041	Σx	
C042	Σy	
C043	×	
C044	n	
C045	÷	
C046	–	
C047	STO V	Store Sxy
C048	m	Calculate MSE
C049	×	
C050	+/–	
C051	RCL U	
C052	+	
C053	n	
C054	2	
C055	–	
C056	÷	
C057	STO M	Store MSE
C058	r	Push r^2, intercept, and slope in the stack
C059	x^2	
C060	b	
C061	m	
C062	SF 10	Set EQN display flag mode
C063	RSQR, INTR, SLP	This is a "tag" created using EQN button
C064	PSE	Display the output tag for the r^2, intercept, and slope
C065	CF10	
C066	RTN	Display the values for for the r^2, intercept, and slope
N001	LBL N	Display components of ANOVA table
N002	m	Calculate regression sum of squares (SSR)
N003	RCL V	
N004	×	
N005	1	Push DF ins tack
N006	SF10	
N007	SSR, DF	This is a "tag" created using EQN button
N008	PSE	Display output tag for SSR and DF
N009	STOP	Display numerical value(s) associated with the above tag
N010	÷	Calculate mean square for regression
N011	MSR	This is a "tag" created using EQN button
N012	PSE	Display output tag for MSR
N013	STOP	Display numerical value(s) associated with the above tag
N014	RCL M	
N015	÷	Calculate F statistic
N016	F STAT	This is a "tag" created using EQN button
N017	PSE	Display output tag for the F statistic

Line	Command	Comments
N018	STOP	Display numerical value(s) associated with the above tag
N019	RCL U	
N020	m	
N021	RCL V	
N022	×	
N023	–	Calculate residual sum of squares (SSE)
N024	n	
N025	2	
N026	–	Calculate degrees of freedom for SSE
N027	SSE, DF	This is a “tag” created using EQN button
N028	PSE	Display the output tag for SSE and its degrees of freedom
N029	STOP	Display numerical value(s) associated with the above tag
N030	÷	Calculate mean square for residual (MSE)
N031	MSE	This is a “tag” created using EQN button
N032	PSE	Display the output tag for MSE
N033	STOP	Display numerical value(s) associated with the above tag
N034	RCL U	Get total sum of squares
N035	n	
N036	1	
N037	–	Calculate df for total sum of squares
N038	TSS, DF	This is a “tag” created using EQN button
N039	PSE	Display output tag for TSS and its degrees of freedom
N040	CF 10	
N041	RTN	Display numerical value(s) associated with the above tag
F001	LBL F	Confidence intervals for intercept, slope, and r^2
F002	INPUT T	Get inverse Student-t
F003	$\bar{x}$	
F004	x^2	
F005	RCL S	
F006	÷	
F007	n	
F008	1/x	
F009	+	
F010	RCL M	
F011	×	
F012	√x	Calculate standard error for intercept
F013	RCL T	
F014	×	Calculate confidence interval / 2
F015	STO X	
F016	b	
F017	+	
F018	b	
F019	RCL X	
F020	–	

Line	Command	Comments
F021	SF 10	
F022	INT UL,LL	This is a “tag” created using EQN button
F023	PSE	Display output tag for the confidence interval of the intercept
F024	STOP	Display confidence interval for intercept
F025	RCL M	
F026	RCL S	
F027	÷	
F028	√x	Calculate standard error for slope
F029	RCL T	
F030	×	Calculate confidence interval / 2
F031	STO X	
F032	m	
F033	+	
F034	m	
F035	RCL X	
F036	–	
F037	SLP UL, LL	This is a “tag” created using EQN button
F038	PSE	Display output tag for the confidence interval of the slope
F039	STOP	Display confidence interval for slope
F040	INPUT Z	Get inverse normal pdf
F041	r	
F042	ATANH	
F043	STO X	
F044	RCL Z	
F045	n	
F046	3	
F047	–	
F048	√x	
F049	÷	
F050	STO Y	
F051	+	
F052	TANH	
F053	x^2	Calculate upper limit for r^2
F054	RCL X	
F055	RCL Y	
F056	–	
F057	TANH	
F058	x^2	Calculate lower limit for r^2
F059	x↔y	
F060	RSQ UL, LL	This is a “tag” created using EQN button
F061	PSE	Display output tag for the confidence interval of the slope
F062	CF 10	
F063	RTN	Display confidence interval for r^2
Q001	LBL Q	Project a new x onto y

Line	Command	Comments
Q002	SF 2	
Q003	GTO P003	
P001	LBL P	Project an existing x onto y
P002	CF 2	
P003	INPUT T	Get inverse Student-t
P004	INPUT X	Get value for x
P005	XEQ X	Transform x
P006	STO X	
P007	$\hat{y}$	Calculate projected y
P008	STO Y	
P009	RCL X	
P010	$\bar{x}$	
P011	–	
P012	x^2	
P013	RCL S	
P014	÷	
P015	n	
P016	1/x	
P017	+	
P018	FS? 2	Use flag 2 to add one if x is a new value
P019	1	
P020	FS? 2	
P021	+	
P022	CF 2	
P023	RCL M	
P024	×	
P025	√x	Calculate standard error in y hat
P026	RCL T	
P027	×	
P028	STO K	
P029	+/–	
P030	RCL Y	
P031	+	Calculate lower limit for $\hat{y}$
P032	XEQ Z	Inverse transform $\hat{y}$ value
P033	STO I	
P034	RCL Y	
P035	XEQ Z	Inverse transform $\hat{y}$ value
P036	STO J	
P037	RCL Y	
P038	RCL K	
P039	+	Calculate upper limit for $\hat{y}$
P040	XEQ Z	Inverse transform $\hat{y}$ value
P041	RCL J	Get $\hat{y}$

Line	Command	Comments
P042	RCL I	Get lower limit $\hat{y}$
P043	SF 10	
P044	Y^U,Y^,Y^L	This is a "tag" created using EQN button
P045	PSE	Display output tag for $\hat{y}$ upper limit, $\hat{y}$, and $\hat{y}$ lower limit
P046	CF 10	
P047	RTN	Display $\hat{y}$ upper limit, $\hat{y}$, and $\hat{y}$ lower limit
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT Y	Prompt user for y value
G004	INPUT W	Prompt user for weight
G005	[X,Y,W]	Create data point. Enter this statement using the EQN key.
G006	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	Enter this statement using the EQN key.
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-LR1	This is a "tag" created using EQN button
U004	PSE	Display program name
U005	CF 10	
U006	RTN	
B001	LBL B	Emulate the action of built-in Σ+
B002	-27	
B003	STO J	Access register n
B004	RCL W	
B005	STO+ (J)	
B006	1	
B007	STO- J	Access register Σx
B008	RCL W	
B009	RCL X	
B010	×	
B011	STO+ (J)	
B012	1	
B013	STO- J	Access register Σy
B014	RCL W	
B015	RCL Y	
B016	×	
B017	STO+ (J)	
B018	1	
B019	STO- J	Access register Σx^2
B020	RCL W	
B021	RCL X	

Line	Command	Comments
B022	x^2	
B023	×	
B024	STO+ (J)	
B025	1	
B026	STO- J	Access register Σy^2
B027	RCL W	
B028	RCL Y	
B029	x^2	
B030	×	
B031	STO+ (J)	
B032	1	
B033	STO- J	Access register Σxy
B034	RCL W	
B035	RCL X	
B036	×	
B037	RCL Y	
B038	×	
B039	STO+ (I)	
B040	RTN	

Example 1

Let's look at an example that exercises various routines in the regression program. The next table contains (x, y, w) data points that we want to fit in a simple linear regression model. Notice that the values of the weights are all ones. The second example, presented in the next section, shows the same (x, y) data with some of the weights being greater than one.

x	y	w
2	24.3	1
4	19.7	1
6	17.8	1
8	14.0	1
10	12.3	1
12	7.2	1
14	5.5	1

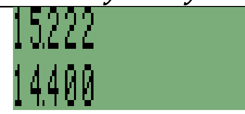
After you enter the above data, perform the following steps:

1. View all of the data points.
2. Calculate the regression slope, intercept, and r^2.
3. Calculate the regression ANOVA table.
4. Calculate confidence interval for the intercept, slope, and r^2 at a 95% confidence level.
5. Calculate the projected value of y and its confidence interval for x = 8 at 95% confidence.

The following table shows the steps using the FIX 3 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0.000
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	2	R/S	Y?
	Enter a value for y.	24.3	R/S	W?
	Enter a value for the weight.	1	R/S	1.000
3	Add the second data point.		XEQ A	X?
	Enter a value for x.	4	R/S	Y?
	Enter a value for y.	19.7	R/S	W?
	Enter a value for the weight.	1	R/S	2.000
4	Repeat step 3 for all other data point.			7.000
5	View all data points (Note the program displays the results using the PSE statement). If you spot an error use XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	7.007 [14.000,5.500,1.0
6	Calculate the regression coefficients.		XEQ C	26.814 -1.552 Displays the intercept and slope
			R↓	0.987 26.814 Displays the values for r^2 and the intercept
	The fitted line is: y = 26.814 – 1.552 x with an r^2 of 0.987.			
7	Calculate the regression ANOVA table.		XEQ N	269.700 1.000 Shows the Regression sum of squares (SSR) and the Regression DF
			R/S	26.814 269.700 Shows the Mean Square for regression (MSR)

Step #	Step/Substep	Input	Command	Output
			R/S	26.814 376.714 Shows the calculated F statistic
			R/S	3.580 5.000 Shows the Residual sum of squares (SSE) and the Residual df
			R/S	376.714 0.716 Shows the Mean Square for residuals (MSE)
			R/S	273.280 6.000 Shows the Total Sum of Squares and the Total df
8	Calculate confidence intervals.		XEQ F	T?
	Enter the value for the inverse Student-t for 5 DF and a 0.025 probability.	2.571	R/S	28.653 24.976 Shows the Intercept confidence interval
			R/S	-1.346 -1.757 Shows the confidence interval for the slope
			R/S	Z?
	Enter the value for the inverse Normal for a 0.025 probability.	1.960	R/S	0.998 0.911 Shows the confidence interval for r^2
9	Project an existing x onto y.		XEQ P	T?
	Enter the value for the inverse Student-t for 5 DF and a 0.025 probability.	2.571	R/S	X?
	Enter value for X.	8	R/S	14.400 13.578

Step #	Step/Substep	Input	Command	Output
				Shows $\hat{y}$ and $\hat{y}$ lower limit
			R↓	 Shows $\hat{y}$ upper limit and $\hat{y}$

Example 2

Let's look at the data in example 1 with some of the weights having values greater than 1.

x	y	w
2	24.3	1
4	19.7	2
6	17.8	2
8	14.0	3
10	12.3	4
12	7.2	2
14	5.5	1

After you enter the above data, perform the following steps:

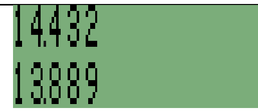
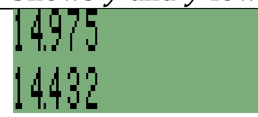
1. View all of the data points.
2. Calculate the regression slope, intercept, and r^2.
3. Calculate the regression ANOVA table.
4. Calculate confidence interval for the intercept, slope, and r^2 at a 95% confidence level.
5. Calculate the projected value of y and its confidence interval for x = 8 at 95% confidence.
6. Compare the intercept, slope, r^2, MSR, F, and n for examples 1 and 2.

The following table shows the steps using FIX 3 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0.000
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	2	R/S	Y?
	Enter a value for y.	24.3	R/S	W?
	Enter a value for the weight.	1	R/S	1.000
3	Add the second		XEQ A	X?

Step #	Step/Substep	Input	Command	Output
	data point.			
	Enter a value for x.	4	R/S	Y?
	Enter a value for y.	19.7	R/S	W?
	Enter a value for the weight.	2	R/S	2.000
4	Repeat step 3 for all other data point.			7.000
5	View all data points (Note the program displays the results using the PSE statement). If you spot an error use XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	7.007 [14.000,5.500,1.0
6	Calculate the regression coefficients.		XEQ C	26.599 -1.521 Displays the intercept and slope
			R↓	0.977 26.599 Displays the values for r^2 and the intercept
	The fitted line is: y = 26.599 – 1.521 x with an r^2 of 0.977.			
7	Calculate the regression ANOVA table.		XEQ N	367.589 1.000 Shows the Regression sum of squares (SSR) and the Regression DF

Step #	Step/Substep	Input	Command	Output
			R/S	26.599 367.589 Shows the Mean Square for regression (MSR)
			R/S	26.599 553.051 Shows the calculated F statistic
			R/S	8.641 13.000 Shows the Residual sum of squares (SSE) and the Residual df
			R/S	553.051 0.665 Shows the Mean Square for residuals (MSE)
			R/S	376.229 14.000 Shows the Total Sum of Squares and the Total df
8	Calculate confidence intervals.		XEQ F	T?
	Enter the value for the inverse Student-t for 5 DF and a 0.025 probability.	2.571	R/S	28.076 25.122 Shows the Intercept confidence interval
			R/S	-1.355 -1.687 Shows the confidence interval for the slope
			R/S	Z?
	Enter the value for the inverse Normal for a 0.025 probability.	1.960	R/S	0.993 0.930 Shows the confidence interval for r^2
9	Project an		XEQ P	T?

Step #	Step/Substep	Input	Command	Output
	existing x onto y.			
	Enter the value for the inverse Student-t for 5 DF and a 0.025 probability.	2.571	R/S	X?
	Enter value for X.	8	R/S	 Shows $\hat{y}$ and $\hat{y}$ lower limit
			R↓	 Shows $\hat{y}$ upper limit and $\hat{y}$

The following table compares a selection of the regression statistics for the two examples.

Regression Statistic	Value in Example 1	Value in Example 2
Intercept	26.814	26.599
Slope	-1.552	-1.521
r^2	0.987	0.977
MSE	0.716	0.665
F statistic	376.714	553.051
n	7	15

If you desire to apply data transformations to the x and/or y observations you need to edit the labels X, Y, and Z. For example, if you wish to use a power fit y = ax^b, then you need to have the labels X, Y, and Z a follows:

X001 LBL X
X002 LN
X003 RTN
Y001 LBL Y
Y002 LN
Y003 RTN
Z001 LBL Z
Z002 Exp
Z003 RTN

As another example, if you wish to use a power fit 1/y = a + b/x, then you need to have the labels X, Y, and Z a follows:

```
X001 LBL X
X002 1/x
X003 RTN
Y001 LBL Y
Y002 1/x
Y003 RTN
Z001 LBL Z
Z002 1/x
Z003 RTN
```

Notice that the code in labels Y and Z is the same because the reciprocal function is the reciprocal of itself. Thus, one can rewrite the above code in a slightly more compact form taking advantage of the specific case:

```
X001 LBL X
Y001 LBL Y
Z001 LBL Z
Z002 1/x
Z003 RTN
```

Appendix A contains new approximations that I recently developed for inverse normal and Student-t functions. The appendix also contains tables for selected values.

Moving Average Program NCS_MA1

Overview

The NCS_MA1 program calculates the moving average for a set of observations. As you add new observations, the program discards the oldest observations and recalculates the moving average using the new data. The program supports the following operations:

- Initializing the program by specifying the number of data points that you want to use to calculate the moving average. This step also includes entering the number of observations you specified.
- Adding a new value which replaces the oldest one.
- Deleting the last entered value and re-entering a new value.
- Editing a data point.
- Calculating the moving average and the standard deviation for a set of current set of values.

The program stores the averaged data set x, in the unnamed memory registers. In addition, the program uses the register J to maintain the index to the *oldest* element in the array x. This oldest element is the candidate for replacement with a new value. The program does not rearrange the data elements, since that may take a while, especially for large arrays. Instead, using the register J, the program maintains a *logical circular array* that tracks the oldest element.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	N?	
	Enter the number of elements to maintain.	N	R/S	X?	
	Enter X(1).	X(1)	R/S	X?	
	Enter X(2).	X(2)	R/S	X?	
	Repeat the above steps to enter all N elements of the array x. When you are done, the program displays the standard deviation and the average for the initial array.		R/S	Moving average	Standard deviation
2	Add a new data point.		XEQ A	X?	
	Enter a new value for the array X.	X	R/S	Moving average	Standard deviation
3	Replace the last value you entered with a new value.		XEQ D	X?	
	Enter a new value for the array X.	X	R/S	Moving average	Standard deviation
4	Edit a data point in the array of X. You can use the command XEQ V to view the current values in the array X and the indices of each value. Use the index associated with the value you wish to edit.		XEQ E	I?	
	Enter the index of the array element. If you enter a value that exceeds the value of N, the program generates the error message INVALID DATA.	index	R/S	X?	
	Enter a new value for the array X.	X	R/S	Moving average	Standard deviation
5	To view the array elements. The program displays the index and value for each array element. The program pauses between values.		XEQ V	X(I)	I
6	Display the program name. The program pauses to		XEQ U	NCS-MA1	

Step #	Step/Substep	Input	Command	Output X	Output Y
	display the program name.				

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A		1	Index invalid when set
B		B		10	Set = Display equations as output tags
C	View sdev and mean	C			
D	Delete the last point	D			
E	Edit a data point	E			
I	Initialize program and enter the first set of data points	F			
V	View data points	G			
U	Display program name	H			
	Internally used labels	I	Index		
J	Set the loop index I to REGX + N/1000	J	Index of next point to remove		
W	Helps in viewing data	K	Data point counter		
		L			
		M			
		N	Number of points N		
		O			
		P			
		Q			
		R			
		S			
		T			
		U			
		V			
		W			
		X	x		
		Y			
		Z			
		0			
		1	X(1)		
		2	X(2)		
		…			
		#N	X(N)		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize program and enter the first set of data points
I002	ClΣ	
I003	INPUT N	
I004	1	
I005	XEQ J	
I006	INPUT X	Start loop: prompt user to enter a value
I007	RCL X	
I008	STO (I)	Store the value
I009	ENTER	
I010	Σ+	Add to summation registers
I011	ISG I	
I012	GTO I006	End of loop
I013	1	
I014	STO J	Store 1 in index of next data point to overwrite
I015	GTO C001	
J001	LBL J	Set the loop index I to REGX + N/1000
J002	RCL N	
J003	1E3	
J004	÷	
J005	+	
J006	STO I	
J007	RTN	
A001	LBL A	Add a new observation
A002	RCL (J)	Recall oldest value
A003	ENTER	
A004	Σ-	
A005	INPUT X	Prompt user to enter a value
A006	RCL X	
A007	ENTER	
A008	STO (J)	Replace oldest value with newest value
A009	Σ+	
A010	1	
A011	STO+ J	Increment point to the next oldest value
A012	RCL N	
A013	RCL J	
A014	x>x?	Did value in J exceed the number of point
A015	1	Assign 1 to index J
A016	STO J	
A017	GTO C001	
D001	LBL D	Delete the last point
D002	RCL J	Determine the index of the last point updated, which is

Line	Command	Comments
D003	1	equal to the value of J - 1. Take into account wrapping
D004	-	values. If J-1 = 0 then I= N Else I = J - 1
D005	x=0?	
D006	RCL N	
D007	STO I	
D008	RCL (I)	Recall the last point
D009	ENTER	
D010	Σ-	Decrement old value from stat registers
D011	INPUT X	Prompt user to enter a value
D012	RCL X	
D013	STO (I)	
D014	ENTER	
D015	Σ+	Increment new value to stat registers
D016	GTO C001	
V001	LBL V	
V002	1	
V003	STO K	Initialize counter
V004	RCL J	
V005	XEQ J	Initialize loop variable as J + N/1000
V006	XEQ W	View value
V007	1	Did J start with the first array element?
V008	RCL J	
V009	x=y?	
V010	RTN	If yes, then exit
V011	1	
V012	-	
V013	1E3	
V014	÷	
V015	+	
V016	STO I	Initialize loop variable as 1 + (J-1)/1000
V017	XEQ W	View value
V018	RTN	
W001	LBL W	
W002	RCL K	
W003	RCL (I)	
W004	PSE	
W005	1	
W006	STO+ K	
W007	ISG I	
W008	GTO W002	
W009	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the value to edit
E003	RCL I	

Line	Command	Comments
E004	RCL N	
E005	x<y?	Is index greater than N?
E006	ACOS(2)	Generate INVALID DATA error message
E007	RCL (I)	
E008	ENTER	
E009	Σ-	
E010	INPUT X	Prompt user for the new value
E011	RCL X	
E012	ENTER	
E013	STO (I)	
E014	Σ+	
C001	LBL C	View basic statistics
C002	SF 10	
C003	SDEV<MEAN	Create a tag using the EQN button
C004	PSE	Display tag
C005	CF 10	
C006	sx	
C007	$\bar{x}$	Display sdev and mean
C008	RTN	

Example

Let's look at an example that exercises various routines in the NCS_MA1 program. The next table has the data used to be calculate moving averages over 5 points. .

I	X
1	77
2	79
3	82
4	80
5	78
6	80
7	82
8	85
9	79
10	76
11	77
12	81

Perform the following steps:

1. Initialize the program by entering the first five values of X.
2. Obtain the moving averages for I = 5 to 12, one point at a time.
3. View the current array after you enter all of the data in the above table.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	N?
	Enter the number of elements to average.	5	R/S	
	Enter X(1)	77	R/S	X?
	Enter X(2)	79	R/S	X?
	Repeat the above steps to enter all N elements of the array X. When you are done, the program displays the standard deviation and the average for the initial array.		R/S	1.92 79.2
2	Add the value of X(6).		XEQ A	X?
	Enter the new value.	80	R/S	1.48 79.8
	Repeat the above steps for the rest of the values in data table.	81	R/S	3.58 79.60
3	View the current data. The program displays the index and value for each array element. The program pauses between values.		XEQ V	5.00 81.00

Here is the resulting table that shows the original data, the moving average values, and the standard deviation values:

i	X	MA	Sdev
1	77		
2	79		
3	82		
4	80		
5	78	79.20	1.92
6	80	79.80	1.48

i	X	MA	Sdev
7	82	80.4	1.67
8	85	81.00	2.65
9	79	80.80	2.77
10	76	80.40	3.36
11	77	79.80	3.70
12	81	79.60	3.58

Moving Average Program NCS_MA2

Overview

The NCS_MA2 program calculates the moving average for a set of observations. As you add new observations, the program discards the oldest observations and recalculates the moving average using the new data. The program supports the following operations:

- Initializing the program by specifying the number of data points that you want to use to calculate the moving average. This step also includes entering the number of observations you specified.
- Adding a new value which replaces the oldest one.
- Deleting the last entered value and re-entering a new value.
- Editing a data point.
- Calculating the moving average, the standard deviation, the confidence interval for the moving average, and the confidence interval for the standard deviation, for the current set of values. The calculation of the two confidence intervals is how the NCS_MA2 program differs from the NCS_MA1 version.

The program stores the averaged data set x, in the unnamed memory registers. In addition, the program uses the register J to maintain the index to the *oldest* element in the array x. This oldest element is the candidate for replacement with a new value. The program does not rearrange the data elements, since that may take a while for large arrays. Instead, using the register J, the program maintains a logical circular array that tracks the oldest element.

✍ You can use the program to merely calculate the mean, standard deviation, the confidence intervals for the mean value, and the standard deviation for a set of observations. To do this, specify the number of observations for all your data when the program prompts you for the number of points to average. Next, enter the appropriate values for the Student-t and chi-square that take into account the number of observations. Finally, key in the data as the program prompts you. When you are done entering your data, the program provides with the mean, standard deviation, the confidence interval for the mean, and the confidence interval for the standard deviation. Once you get these results you need not add more observations.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	N?	
	Enter the number of elements to maintain.	N	R/S	T?	
	Enter the value for Student-t for N-1 degrees of freedom and (1-α) confidence level.	Student-t	R/S	C?	
	Enter the value for chi-square for N-1 degrees of freedom and (α/2) significance level.	chi-square	R/S	D?	
	Enter the value for chi-square for N-1 degrees of freedom and (1-α/2) significance level.	chi-square	R/S	X?	
	Enter X(1).	X(1)	R/S	X?	
	Enter X(2).	X(2)	R/S	X?	
	Repeat the above steps to enter all N elements of the array x. When you are done, the program displays the standard deviation and the average for the initial array.		R/S	Moving average	Standard deviation
2	Add a new data point.		XEQ A	X?	
	Enter a new value for the array X.	X	R/S	Moving average	Standard deviation
	View the upper and lower values for the confidence interval for the moving average.		R/S	Lower limit for the moving average	Upper limit for the moving average
	View the upper and lower values for the confidence interval for the standard deviation.			Lower limit for the standard deviation	Upper limit for the standard deviation
3	Replace the last value you		XEQ D	X?	

Step #	Step/Substep	Input	Command	Output X	Output Y
	entered with a new value.				
	Enter a new value for the array X.	X	R/S	Moving average	Sdev
	View the upper and lower values for the confidence interval for the moving average.		R/S	Lower limit for the moving average	Upper limit for the moving average
	View the upper and lower values for the confidence interval for the standard deviation.			Lower limit for the standard deviation	Upper limit for the standard deviation
4	Edit a data point in the array of X. You can use the command XEQ V to view the current values in the array X and the indices of each value. Use the index associated with the value you wish to edit.		XEQ E	I?	
	Enter the index of the array element. If you enter a value that exceeds the value of N, the program generates the error message INVALID DATA.	index	R/S	X?	
	Enter a new value for the array X.	X	R/S	Moving average	Standard deviation
	View the upper and lower values for the confidence interval for the moving average.		R/S	Lower limit for the moving average	Upper limit for the moving average
	View the upper and lower values for the confidence interval for the standard deviation.			Lower limit for the standard deviation	Upper limit for the standard deviation
5	To view the array elements. The program displays the index and value for each array element. The program pauses between values.		XEQ V	X(I)	I
6	Display the program name. The program		XEQ U	NCS-MA2	

Step #	Step/Substep	Input	Command	Output X	Output Y
	pauses to display the program name.				

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A		1	Index invalid when set
B		B		10	Set = Display equations as output tags
C	View sdev and mean and their confidence intervals	C	Chi-square used in calculating the confidence interval for the standard deviation. This is the value for chi-square(n-1,α/2)		
D	Delete the last point	D	Chi-square used in calculating the confidence interval for the standard deviation. This is the value for chi-square(n-1,1-α/2)		
E	Edit a data point	E			
I	Initialize program and enter the first set of data points	F			
V	View data points	G			
U	Display program name	H			
	Internally used labels	I	Index		
J	Set the loop index I to REGX + N/1000	J	Index of next point to remove		
W	Helps in viewing data	K	Data point counter		
		L			
		M			
		N	Number of points N		
		O			
		P			
		Q			
		R			
		S			
		T	Student-t used in		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
			calculating the confidence interval for the moving average		
		U			
		V			
		W			
		X	x		
		Y			
		Z	Used		
		0			
		1	X(1)		
		2	X(2)		
		…			
		#N	X(N)		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize program and enter the first set of data points
I002	ClΣ	
I003	INPUT N	Enter the number of points to average
I004	INPUT T	Enter the Student-t value
I005	INPUT C	Enter the chi-square(n-1,α/2) value
I006	INPUT D	Enter the chi-square(n-1,1-α/2) value
I007	1	
I008	XEQ J	
I009	INPUT X	Start loop: prompt user to enter a value
I010	RCL X	
I011	STO (I)	Store the value
I012	ENTER	
I013	Σ+	Add to summation registers
I014	ISG I	
I015	GTO I009	End of loop
I016	1	
I017	STO J	Store 1 in index of next data point to overwrite
I018	GTO C001	
J001	LBL J	Set the loop index I to REGX + N/1000
J002	RCL N	
J003	1E3	
J004	÷	
J005	+	
J006	STO I	

Line	Command	Comments
J007	RTN	
A001	LBL A	Add a new data point
A002	RCL (J)	Recall oldest value
A003	ENTER	
A004	Σ-	
A005	INPUT X	Prompt user to enter a value
A006	RCL X	
A007	ENTER	
A008	STO (J)	Replace oldest value with newest value
A009	Σ+	
A010	1	
A011	STO+ J	Increment point to the next oldest value
A012	RCL N	
A013	RCL J	
A014	x>x?	Did value in J exceed the number of point
A015	1	Assign 1 to index J
A016	STO J	
A017	GTO C001	
D001	LBL D	Delete the last point
D002	RCL J	Determine the index of the last point updated, which is
D003	1	equal to the value of J - 1. Take into account wrapping
D004	-	values. If J-1 = 0 then I= N Else I = J - 1
D005	x=0?	
D006	RCL N	
D007	STO I	
D008	RCL (I)	Recall the last point
D009	ENTER	
D010	Σ-	Decrement old value from stat registers
D011	INPUT X	Prompt user to enter a value
D012	RCL X	
D013	STO (I)	
D014	ENTER	
D015	Σ+	Increment new value to stat registers
D016	GTO C001	
V001	LBL V	View a data point
V002	1	
V003	STO K	Initialize counter
V004	RCL J	
V005	XEQ J	Initialize loop variable as J + N/1000
V006	XEQ W	View value
V007	1	Did J start with the first array element?
V008	RCL J	
V009	x=y?	
V010	RTN	If yes, then exit

Line	Command	Comments
V011	1	
V012	-	
V013	1E3	
V014	÷	
V015	+	
V016	STO I	Initialize loop variable as 1 + (J-1)/1000
V017	XEQ W	View value
V018	RTN	
W001	LBL W	
W002	RCL K	
W003	RCL (I)	
W004	PSE	
W005	1	
W006	STO+ K	
W007	ISG I	
W008	GTO W002	
W009	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the value to edit
E003	RCL I	
E004	RCL N	
E005	x<y?	Is index greater than N?
E006	ACOS(2)	Generate INVALID DATA error message
E007	RCL (I)	
E008	ENTER	
E009	Σ-	
E010	INPUT X	Prompt user for the new value
E011	RCL X	
E012	ENTER	
E013	STO (I)	
E014	Σ+	
C001	LBL C	View basic statistics and their confidence intervals
C002	SF 10	
C003	SDEV,MEAN	This is an output "tag" created using the EQN key
C004	PSE	Display the output tag
C005	sx	
C006	x̄	
C007	STOP	Display sdev and mean
C008	x↔y	
C009	n	
C010	√x	
C011	÷	
C012	RCL T	

Line	Command	Comments
C013	×	Calculate difference from the moving average
C014	STO Z	
C015	$\bar{x}$	
C016	+	Calculate the upper limit of the moving average
C017	LASTx	
C018	RCL Z	
C019	-	Calculate the lower limit of the moving average
C020	MA UL,LL	This is an output "tag" created using the EQN key
C021	PSE	Display the output tag
C022	STOP	Display confidence limit for the moving average
C023	sx	
C024	x^2	
C025	n	
C026	1	
C027	-	
C028	×	
C029	STO Z	
C030	RCL D	
C031	÷	Calculate upper limit for the confidence interval
C032	RCL Z	
C033	RCL C	
C034	÷	Calculate upper lower for the confidence interval
C035	SDEV UL,LL	This is an output "tag" created using the EQN key
C036	PSE	Display the output tag
C037	CF 10	
C038	RTN	Display confidence limit for the standard deviation

Example

Let's look at an example that exercises various routines in the NCS_MA2 program. The next table has the data used to be calculate moving averages over 5 points. .

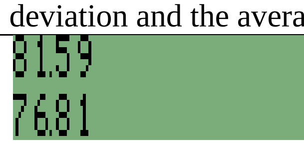
I	X
1	77
2	79
3	82
4	80
5	78
6	80
7	82
8	85
9	79
10	76

11	77
12	81

Perform the following steps:

1. Initialize the program by entering the first five values of X.
2. Obtain the moving averages for I = 5 to 12, one point at a time. Also use the program to calculate the confidence interval for the moving averages and standard deviations.
3. View the current array after you enter all of the data in the above table.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	N?
	Enter the number of elements to average.	5	R/S	T?
	Enter the Student-t at 4 degrees of freedom and a 0.25 significance level.	2.776	R/S	C?
	Enter chi-square(4,0.025)	11.1433	R/S	D?
	Enter chi-square(4,0.975)	0.484419	R/S	X?
	Enter X(1)	77	R/S	X?
	Enter X(2)	79	R/S	X?
	Repeat the above steps to enter all N elements of the array X. When you are done, the program displays the standard deviation and the average for the initial array.		R/S	 Displays the standard deviation and the average
	View the confidence interval of the moving average.		R/S	 Displays the upper level and lower level for the confidence interval of the moving average

Step #	Step/Substep	Input	Command	Output
	View the confidence interval of the standard deviation.		R/S	30.55 1.33 Displays the upper level and lower level for the confidence interval of the standard deviation
2	Add the value of X(6).		XEQ A	X?
	Enter the new value.	80	R/S	1.48 79.80 Displays the standard deviation and the moving average
	View the confidence interval of the moving average.			81.64 77.96 Displays the upper level and lower level for the confidence interval of the moving average
	View the confidence interval of the standard deviation.			18.17 0.79 Displays the upper level and lower level for the confidence interval of the standard deviation
	Repeat the above steps for the rest of the values in data table.	81	R/S	3.36 80.40 Displays the standard deviation and the moving average
	View the confidence interval of the moving average.			84.57 76.23 Displays the upper level and lower level for the confidence interval of the moving average
	View the confidence interval of the standard deviation.			93.31 4.06 Displays the upper level and lower level for the confidence

Step #	Step/Substep	Input	Command	Output
				interval of the standard deviation
3	View the current data. The program displays the index and value for each array element. The program pauses between values.		XEQ V	5.00 81.00

Here is the resulting table that shows the original data, the moving average values, and the standard deviation values:

i	X	MA	Sdev	MA LL	MA UL	SDEV LL	SDEV UL
1	77						
2	79						
3	82						
4	80						
5	78	79.20	1.92	76.81	81.59	1.33	30.55
6	80	79.80	1.48	77.96	81.64	0.79	18.17
7	82	80.4	1.67	78.32	82.48	1.01	23.12
8	85	81.00	2.65	77.72	84.28	2.51	57.80
9	79	80.80	2.77	77.36	84.24	2.76	63.58
10	76	80.40	3.36	76.23	84.57	4.06	93.31
11	77	79.80	3.70				
12	81	79.60	3.58				

Mann-Whitney Test Program NCS_MANN1

Overview

The NCS_MANN1 program calculates the Mann-Whitney test statistic for an array of paired or unpaired ranked values.

The program offers the following operations:

1. Adding an array of rank values.
2. Editing a rank value.
3. Viewing specific rank value.
4. Viewing the values of the array of ranks.
5. Calculating the Mann-Whitney test statistic and related statistics.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

The Mann-Whitney test compares the ranks of two paired or unpaired arrays of values to determine if the two arrays are of the same population. The two arrays (one with N elements, the other with M elements) are ranked together as if they were one array. However, you need to enter the ranks of only one array.

The Mann-Whitney test statistic is obtained using:

$$U = N\,M + \frac{N(N+1)}{2} - \sum_{i=1}^{N} R_i$$

Here N and M are the number of elements in the two arrays. The summation takes the ranks of the first array that has N elements. The choice of which of the two arrays is first and second is arbitrary. In the case the two arrays are of unequal sizes, chose the smaller array to be the first one. This choice will make you enter fewer rank values and will not affect the calculated value for U.

To test the null hypothesis:

H_0: The two arrays are of the same population based on their ranks.

We calculate an approximation to the inverse normal statistic Z (for M ≥ 8 and N ≥ 8) using the following equation:

$$Z = \left[U - \frac{N\,M}{2}\right] \div \left[\sqrt{N\,M\,(N + M + 1) \div 12}\right]$$

If the calculated Z does not exceed Z_α then we cannot reject the null hypothesis that the two arrays are from the same population based on their ranks.

For small values of N and M consult your favorite statistics book to find special tables for the hypothesis test.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	N?	
	Enter the size of the first array N.	N	R/S	M?	
	Enter the size of the second array M.	M	R/S	M	
2	Add the rank values for the first array.		XEQ A	R?	
	Enter a value for Rank(1).	Rank	R/S	R?	
	Enter a value for Rank(2).	Rank	R/S	R?	
	Enter the next rank value. Repeat this step until you have entered all of the rank value.	Rank	R/S	R?	
3	Edit a rank value.		XEQ E	I?	
	Enter the index for the value.	Index	R/S	R?	
	View the value and optionally enter a replacement value for the rank.	Rank	R/S		

Step #	Step/Substep	Input	Command	Output X	Output Y
4	View the data while pausing at each value. The program displays the values in the table by columns.		XEQ V	Rank	index
5	View the data while stopping at each value.		XEQ W	Rank	index
	To view the next rank value.		R/S	Rank	Index
	Repeat the above steps to view the rest of the array of ranks.		R/S	Rank	index
6	Calculate the Mann-Whitney test statistic and Z value.		XEQ C	Z	U
7	Display the program name. The program pauses to display the program name.		XEQ U	NCS-MANN1	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add the data points	A		1	Set = Pause, Clear = Stop
C	Calculate Mann-Whitney statistic	B			
E	View/Edit an observation	C			
W	View all data points in matrix x	D			
V	View all data points in matrix x without stopping the program	E			
I	Initialize the program	F			
U	Display program name	G			
	For internal use	H			
P	Prepare loop control variable	I	Index		
		J			
		K			
		L			
		M	M		
		N	N		
		O			

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		P	N × M		
		Q			
		R	Rank		
		S	Σ Rank		
		T			
		U	Mann-Whitney statistic U		
		V			
		W			
		X			
		Y			
		Z	Z		
		1	R(1)		
		2	R(2)		
		…			
		N	R(N)		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	INPUT N	Prompt user to enter n
I003	INPUT M	Prompt user to enter m
I004	RTN	
U001	LBL U	
U002	SF 10	
U003	NCS-MANN1	This is an "output tag" created using the EQN button
U004	PSE	
U005	CF 10	
U006	RTN	
A001	LBL A	Input data
A002	XEQ P	Initialize the loop variable
A003	INPUT R	Start the data input loop
A004	STO (I)	
A005	ISG I	
A006	GTO A003	End of the data input loop
A007	RTN	
E001	LBL E	Edit/View a data point
E002	INPUT I	Get the index of data point to edit or view
E003	RCL N	
E004	x<y?	Validate index
E005	ACOS(2)	
E006	RCL (I)	Recall data point

Line	Command	Comments
E007	STO R	
E008	INPUT R	Prompt and show current value
E009	STO (I)	
E010	RTN	
V001	LBL V	View data points without stopping the program
V002	SF 1	
V003	XEQ W003	
V004	CF 1	
V005	RTN	
W001	LBL W	View data points
W002	CF 1	
W003	XEQ P	Initialize the loop variable
W004	RCL I	Start of loop to view array elements
W005	RCL (I)	
W006	FS? 1	
W007	GTO W010	
W008	STOP	
W009	GTO W011	
W010	PSE	
W011	ISG I	
W012	GTO W004	End of loop that views array elements
W013	RTN	
P001	LBL P	Prepare loop index
P002	1	
P003	RCL N	
P004	1E3	
P005	÷	
P006	+	
P007	STO I	
P008	RTN	
C001	LBL C	Calculate Mann-Whitney statistic
C002	XEQ P	Prepare loop index
C003	0	
C004	STO S	Initialize sum of ranks
C005	RCL (I)	Start loop to sum ranks
C006	STO+ S	
C007	ISG I	
C008	GTO C005	End of loops that sums ranks
C009	RCL N	Start calculating U
C010	RCL M	
C011	×	
C012	STO P	Store N×M
C013	RCL N	
C014	RCL N	

Line	Command	Comments
C015	1	
C016	+	
C017	×	
C018	2	
C019	÷	
C020	+	
C021	RCL S	
C022	-	
C023	STO U	Calculate and store the Kendall coefficient of concordance U
C024	RCL P	Start calculating Z
C025	2	
C026	÷	
C027	-	
C028	RCL N	
C029	RCL M	
C030	+	
C031	1	
C032	+	
C033	12	
C034	÷	
C035	RCL P	
C036	×	
C037	√x	
C038	÷	
C039	STO Z	Calculate Z
C040	SF 10	
C041	U,Z	This is an "output tag" created using the EQN button
C042	PSE	Display output tag
C043	CF 10	
C044	RCL U	
C045	x↔y	
C046	RTN	Display the values of the Kendall coefficient of concordance and Z

Example

Let’s look at an example that exercises various routines in the NCS_MANN1 program. The next table contains the raw data and its ranks. You can obtain the rank values using the program NCS_RANKER1, after you edit line C041 in that program to reverse the order of sorting and ranking.

Student ID	Value	Rank
1	14.9	7
2	11.3	1

Student ID	Value	Rank
3	13.2	4
4	16.6	12
5	17	14
6	14.1	5
7	15.4	10
8	13	3
9	16.9	13
1	15.2	8
2	19.8	18
3	14.7	6
4	18.3	15
5	16.2	11
6	21.2	19
7	18.9	16
8	12.2	2
9	15.3	9
10	19.4	17

The program requires that you only enter the ranks of the first array (9 data points). After you enter the above data, calculate the Mann-Whitney test statistic, and Z value. Use the value of Z to check the null hypothesis at a 95% confidence level. If you enter erroneous values, execute the XEQ E command to edit these values. You can also view the data using the command XEQ W before (or even after) you calculate the statistical results.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	N?
	Enter the number of elements in the first array.	9	R/S	M?
	Enter the number of elements in the second array.	10	R/S	10.00
2	Enter the data.		XEQ A	R?
	Enter the value of Rank(1).	7	R/S	R?
	Enter the value of Rank(2).	1	R/S	R?
	Repeat the above steps to enter the rest of the rank values.			
3	Calculate Mann-Whitney test statistic and Z value.		XEQ C	66.00 1.71

Step #	Step/Substep	Input	Command	Output
				Displays the Mann-Whitney test statistic and Z value

Since the calculated Z value of 1.71 exceeds the value of $Z_{0.05}$ (equal to 1.65) we cannot accept the null hypothesis, stating that the two arrays are from the same population based on their ranks.

Using the approximation for Z, shown in the appendix, you get Z = 1.61 for a significance level of 0.05.

Appendix A contains a new approximation that I recently developed for inverse normal function.

Mean-Square Successive Difference Program NCS_MEANSSDF1

Overview

The NCS_MEANSSDF1 program implements the mean-square successive difference statistic to test the randomness of an array of values. The program does not store the raw data for each data group. The program supports the following operations:

- Initializing the program.
- Adding data points.
- Calculating mean-square successive difference statistics.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Given a set of n values drawn as random numbers from a certain population. The null hypothesis states that the numbers are random.

The program calculates the mean-square successive difference statistics, η, which is used to calculate a normally distributed value that tests the null hypothesis.

H_0 : The n values drawn from a population are random.

The program calculates the mean-square successive difference statistic using the following equation:

$$\eta = \sum_{i=2}^{n}(x_i - x_{i-1}) \,/\, [\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2 \,/\, n]$$

$$Z = (1 - \eta/2) \,/\, \sqrt{\frac{(n-2)}{(n^2-1)}}$$

If the value of Z does not exceed the value of Z_α we cannot reject the null hypothesis which states that numbers are random drawn from a certain population. Large values of Z reflect long trends. Large negative values of Z reflect oscillating values in the tested array.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value for x.	x	R/S			
	Enter another value.		R/S	X?		
	Resume with the steps above.					
3	Calculate the degrees of freedom, η, and Z statistics.		XEQ C	η	Z	Degrees of freedom
4	Display the program name. The program pauses to display the program name.		XEQ U	NCS-MEANSSDF1		

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a data point	A	Σ(x(i)-x(i-1))^2	1	Set = First value
C	Calculate the mean-square successive difference statistics.	B			
I	Initialize the program	C			
U	Display the program name	D			
		E			
		F			
		G			
		H			
Internally used labels		I			

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		J			
		K			
		L			
		M			
		N	η		
		O			
		P			
		Q			
		R			
		S			
		T			
		U			
		V			
		W			
		X	X		
		Y	Previous value of X		
		Z	Z		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize program and enter the first set of data points
I002	ClΣ	
I003	0	
I004	STO A	
I005	SF 1	
I006	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-MEANSSDF1	This is an output "tag" created with the EQN key
U004	PSE	
U005	CF 10	
U006	RTN	
A001	LBL A	Add data
A002	INPU X	
A003	RCL X	
A004	Σ+	
A005	FS? 1	
A006	GTO A012	
A007	RCL X	
A008	RCL Y	
A009	-	

Line	Command	Comments
A010	x^2	
A011	STO+ A	
A012	RCL X	
A013	STO Y	
A014	CF 1	
A015	STOP	
A016	GTO A001	
C001	LBL C	Calculate statistics
C002	RCL A	
C003	Σx^2	
C004	Σx	
C005	x^2	
C006	n	
C007	÷	
C008	-	
C009	÷	
C010	STO N	Calculate and store η statistic
C011	-2	
C012	÷	
C013	1	
C014	+	
C015	n	
C016	2	
C017	-	
C018	n	
C019	x^2	
C020	1	
C021	-	
C022	÷	
C023	√x	
C024	÷	
C025	STO Z	
C026	SF 10	
C027	DF,N,Z	This is an output "tag" created with the EQN key
C028	PSE	Display output tag. Note: Program uses N to mean η
C029	CF 10	
C030	n	
C031	1	
C032	-	Calculate degrees of freedom
C033	RCL N	
C034	RCL Z	
C035	RTN	Display degrees of freedom, η, and Z

Example

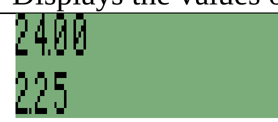
Let's look at an example that exercises various routines in the NCS_MEANSSDF1 program. The next table contains data points that we want to use to test the null hypothesis.

I	X(I)
1	52
2	47
3	76
4	67
5	55
6	46
7	78
8	45
9	51
10	56
11	63
12	64
13	60
14	58
15	53
16	51
17	53
18	60
19	49
20	59
21	61
22	67
23	55
24	46
25	78

After you enter the above data, perform the following steps:

1. Calculate mean-square successive difference statistics.
2. Test the null hypothesis at a 95% confidence level.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Add the first data point in the first data group.		XEQ A	X?
	Enter a value for x.	52	R/S	
	To enter another value.		R/S	X?
	Repeat the above steps for the rest of the values in the table.			
3	Calculate the statistical results. The program displays the value of η and Z.		XEQ C	 Displays the values of η and Z
4	To view the degrees of freedom,		R↓	 Displays the values for the degrees of freedom and Z

Since the absolute value of the calculated Z statistic has a value of 0.64 which does not exceed the value of $Z_{0.95}$ of 1.644 we cannot reject the null hypothesis stating that samples come from the same population.

Appendix A contains a new approximation that I recently developed for inverse normal function.

Multiple Linear Regression Program NCS_MLR1

Overview

The NCS_MLR1 program implements multiple linear regression between three variables, as shown next:

z = a + b x + c y

The program offers the following features:

1. Storing data points once and reusing them for different regression scenarios. The program stores the (x, y, z) data points as 3-D vectors, a new feature of the HP-35s, thus reducing the number of unnamed memory registers used by one third.
2. The ability to delete, edit, view, and swap data points. Thus the regression program incorporates basic operations to manage the data points.
3. Transforming the x, y, and z observations. You can edit the program to implement your custom transformations and then recalculate the regression statistics without having to reenter the data points.
4. Calculating the regression intercept, slopes, and coefficient of determination.
5. Calculating the confidence intervals for intercept and slopes.
6. Calculating the regression ANOVA table.
7. Calculating projected value of z for existing or new values of x and y.

Since the program allows you to transform the x, y, and z values, the general form for the regression model fitted is:

f(z) = a + b g(x) + c h(y)

Where f(z), g(x), and h(y) are functions that you select and code in to perform the required transformation that supports a linearized regression.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Data Collected

$\sum x$ = sum of x
$\sum x^2$ = sum of x^2
$\sum y$ = sum of y
$\sum y^2$ = sum of y^2
$\sum z$ = sum of z
$\sum z^2$ = sum of z^2
$\sum xy$ = sum of $x \cdot y$
$\sum xz$ = sum of $x \cdot z$
$\sum yz$ = sum of $y \cdot z$
n = number of observations

Regression Coefficients

$x_m = \sum x / n$
$y_m = \sum y / n$
$z_m = \sum z / n$
$S_{xx} = \sum x^2 - (\sum x)^2 / n = \sum x^2 - n (x_m)^2$
$S_{yy} = \sum y^2 - (\sum y)^2 / n = \sum y^2 - n (y_m)^2$
$S_{zz} = \sum z^2 - (\sum z)^2 / n = \sum z^2 - n (z_m)^2$
$S_{xy} = \sum xy - (\sum x) (\sum y)/ n = \sum xy - n\, x_m\, y_m$
$S_{xz} = \sum xz - (\sum x) (\sum z)/ n = \sum xz - n\, x_m\, z_m$
$S_{yz} = \sum yz - (\sum y) (\sum z)/ n = \sum yz - n\, y_m\, z_m$

$D = S_{xx} S_{yy} - (S_{xy})^2$
Slope $b = [S_{xz} S_{yy} - S_{yz} S_{xy}] / [S_{xx} S_{yy} - (S_{xy})^2]$
Slope $c = [S_{yz} S_{xx} - S_{xz} S_{xy}] / [S_{xx} S_{yy} - (S_{xy})^2]$
Intercept $a = z_m - b\, x_m - c\, y_m$
$R^2 = r^2 = [a \sum z + b \sum xz + c \sum yz - (\sum z)^2 / n] / [\sum z^2 - (\sum z)^2 / n]$
$\quad = [a \sum z + b \sum xz + c \sum yz - (\sum z)^2 / n] / S_{zz}$

ANOVA Table

Source of Variation	***Sum of Squares***	***Degrees of Freedom***	***Mean Square***	***F***
Regression	$SS_R = b\, S_{xz} + c\, S_{yz}$	2	MS_R	MS_R / MS_E
Residual	$SS_E = S_{zz} - b\, S_{xz} - c\, S_{yz}$	n – 3	MS_E	
Total	$SS_T = S_{zz}$	n – 1		

Interval for Slope b

At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$
SlopeStdErr = $\sqrt{[MS_E (S_{yy} / D)]}$
$\Delta b = t_{\alpha/2,n-3} \cdot$ SlopeStdErr

Upper Range: $b + \Delta b$
Lower Range: $b - \Delta b$

Interval for Slope c

At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$
SlopeStdErr = $\sqrt{[MS_E (S_{xx} / D)]}$
$\Delta c = t_{\alpha/2,n-3} \cdot$ SlopeStdErr

Upper Range: $c + \Delta c$
Lower Range: $c - \Delta c$

Interval for Intercept a

At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$
IntStdErr = $\sqrt{(MS_E [1/n + (S_{yy} / D)(x_m)^2 + (S_{xx} / D)(y_m)^2 - 2(S_{xy} / D) x_m y_m])}$
$= \sqrt{(MS_E [1/n + \{S_{yy} (x_m)^2 + S_{xx} (y_m)^2 - 2(S_{xy} x_m y_m)\}/D])}$

$\Delta a = t_{\alpha/2,n-3} \cdot$ IntStdErr

Upper Range: $a + \Delta a$
Lower Range: $a - \Delta a$

Interval for Projected z

At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$

$\hat{z} = a + b x_0 + c y_0$

zhatStdErr = $\sqrt{(MS_E [1/n + (S_{yy} / D)(x_0 - x_m)^2 + (S_{xx} / D)(y_0 - y_m)^2 - 2 (S_{xy} / D) (x_0 - x_m) (y_0 - y_m)])}$
$= \sqrt{(MS_E [1/n + \{S_{yy} (x_0 - x_m)^2 + S_{xx} (y_0 - y_m)^2 - 2 S_{xy} (x_0 - x_m) (y_0 - y_m)\} / D])}$

$\Delta\hat{z} = t_{\alpha/2,n-3} \cdot$ zhatStdErr
Upper Range: $\hat{z} + \Delta\hat{z}$
Lower Range: $\hat{z} - \Delta\hat{z}$

For a *new* observation:

zhatStdErr = $t_{\alpha/2,n-3} \cdot \sqrt{((MS_E)^2 [1+1/n + (S_{yy} / D)(x_0 - x_m)^2 + (S_{xx} / D)(y_0 - y_m)^2 - 2 (S_{xy} / D) (x_0 - x_m) (y_0 - y_m)])}$

$= \sqrt{((MS_E)^2 [1 + 1/n + \{S_{yy} (x_0 - x_m)^2 + S_{xx} (y_0 - y_m)^2 - 2 S_{xy} (x_0 - x_m) (y_0 - y_m)\} / D])}$

$\Delta\hat{z} = t_{\alpha/2,n-3} \cdot$ zhatStdErr
Upper Range: $\hat{z} + \Delta\hat{z}$
Lower Range: $\hat{z} - \Delta\hat{z}$

Instructions

The following table shows the steps involved in using the various parts of the regression program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value of x.	x	R/S	Y?		
	Enter a value of y.	y	R/S	Z?		
	Enter a value of z.	z	R/S	Number of data points		
3	Delete a data point.		XEQ D	I?		
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message INVALID DATA.	Index of data point to delete	R/S	Number of data points		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
4	Edit a data point.		XEQ E	I?		
	Enter the index of the data point to edit. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	X?		
	Enter a new value for x.	x	R/S	Y?		
	Enter a new value for y.	y	R/S	Z?		
	Enter a new value for z.	z	R/S			
5	View a data point.		XEQ V	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	Data point	Index	
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	Data point	Index	
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA,		R/S	Number of data points + 1		
6	View all data points (Note the program displays the results using the PSE statement.)		XEQ W	Data point	Index	
7	Swap data points.		XEQ S	I?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	Enter the index of the first data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the first data point	R/S	J?		
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S			
8	Calculate the regression coefficients (note that the value of r^2 is in the T stack register).		XEQ C	Slope c	Slope b	Intercept
	Use the R↓ key to scroll down the stack and examine the values of the four results.		R↓	Slope b	Intercept	r^2
			R↓	Intercept	r^2	Slope c
			R↓	r^2	Slope c	Slope b
			R↓	Slope c	Slope b	Intercept
9	Calculate the regression ANOVA table.		XEQ N	Regression DF	Regression sum of squares (SSR)	
			R/S	Mean Square for regression (MSR)		
			R/S	Calculated F statistic		
			R/S	Residual df	Residual sum of squares (SSE)	
			R/S	Mean		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
				Square for residuals (MSE)		
			R/S	Total df	Total Sum of Squares	
10	Calculate the confidence intervals.		XEQ F	T?		
	Enter a value of the inverse Student-t probability distribution function.	Student-t for n–3 degrees of freedom and an α/2 significance (α is based on 1–α confidence level.)	R/S	Intercept lower limit	Interval upper limit	
			R/S	Slope b lower limit	Slope b upper limit	
			R/S	Slope c lower limit	Slope c upper limit	
11	Project an existing x and y onto z.		XEQ P	T?		
	Enter a value of the inverse Student-t probability distribution function.	Student-t for n–3 degrees of freedom and an α/2 significance (α is based on 1–α confidence level.)	R/S	X?		
		x	R/S	Y?		
		y	R/S	$\hat{z}$ lower limit	$\hat{z}$	$\hat{z}$ upper limit
12	Project a new x and y onto z.		XEQ Q	T?		
	Enter a value of the inverse Student-t probability distribution function.	Student-t for n–3 degrees of freedom and an α/2 significance (α is based	R/S	X?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
		on 1–α confidence level.)				
		x	R/S	Y?		
		y	R/S	$\hat{z}$ lower limit	$\hat{z}$	$\hat{z}$ upper limit
13	You can temporarily (or even permanently) reduce the number of data points entering in the regression calculations.		RCL C	Number of data points		
	Record the number of original data points.					
	Enter a lower value of the number of data points (must be greater than 4.)	New number of data points	STO C			
14	You can restore the original number of data points entering in the regression calculations.	Original number of data points	STO C			
15	Alter the transformation for x.		GTO X001			
	Switch to program mode.		PRGM	X001 LBL X		
	Edit the commands between LBL X and the RTN statement to reflect the new transformation you desire.					
	Exit program mode when you are done.		PRGM			
16	Alter the		GTO Y001			

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	transformation for y.					
	Switch to program mode.		PRGM	Y001 LBLY		
	Edit the commands between LBL Y and the RTN statement to reflect the new transformation you desire.					
	Exit program mode when you are done.		PRGM			
17	Alter the inverse transformation for y.		GTO Z001			
	Switch to program mode.		PRGM	Z001 LBL Z		
	Edit the commands between LBL Z and the RTN statement to reflect the new transformation and inverse transformation you desire. Flag 3 is dedicated to distinguish between the transformation and inverse transformation. When flag 3 is set, LBL Z performs an inverse transformation.					
	Exit program mode when you are done.		PRGM			
18	To display the program name. The program		XEQ U	NCS-MLR1		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	pauses to display the program name.					

Comments on Runtime Errors

The program validates the indices that you enter to manage data points to prevent the corruption of the data points. Aside from this feature, you are responsible to enter data that will avoid runtime errors. For example, if you are using the square root or logarithms to transform the x and/or y data, then entering data points with negative values will raise runtime errors. Likewise, using reciprocals and logarithms will generate runtime errors for data that are zeros. Entering redundant data points also has its peril of giving you meaningless results. Remember the old saying about program input and output data "Junk in, junk out."

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A	Used, r^2	1	Index invalid when set
D	Delete an existing data point	B	Used	2	Project a new x and y onto z when set
E	Edit a data point	C	Count for number of data points	3	Inverse transform z
V	View a data point	D	Σz		
W	View all data points	E	Σz^2		
I	Initialize	F	Σxz		
S	Swap two data points	G	Σyz		
C	Calculate regression coefficients	H	MSE		
N	Calculate regression ANOVA	I	Index, used		
F	Calculate confidence intervals	J	Index, used		
P	Project an observed x and y onto z with confidence interval	K	Intercept a		
Q	Project a new x and y onto z with confidence interval	L	Slope b		
X	Transform x data	M	Slope c		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
Y	Transform y data	N	Sxx		
Z	Transform and Inverse transform of z data	O	Syy		
U	Display program name	P	Sxy		
Labels for internal use		Q	Sxz		
H	Validate indices and display error message if the an index is invalid	R	Sxy		
G	Prompt for x, y and z	S	Szz		
R	Display the message INVALID DATA	T	Inverse Student-t pdf		
		U	$D = S_{xx} S_{yy} – (S_{xy})^2$		
		V	Upper limit		
		W	Lower limit		
		X	x		
		Y	y		
		Z	z		
		1	Data point #1		
		2	Data point #2		
		3	Data point #3		
		…			
		#C	Data point #C		

Program Listing

Line	Command	Comments
X001	LBL X	Label to contain the transformation for x. No code is needed, between the label and the RTN command, for linear values.
X002	RTN	
Y001	LBL Y	Label to contain the transformation for y. No code is needed, between the label and the RTN command, for linear values.
Y002	RTN	
Z001	LBL Z	Label to contain the transformation and inverse transformation for z. No code is needed, between the label and the RTN command, for linear values.
Z002	RTN	
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	STOP	
A011	GTO A001	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	

Line	Command	Comments
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO– C	
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	Validate index
S006	RCL (I)	Recall data points and place them on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x↔y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	RCL (I)	Recall data point
V005	STOP	
V006	1	
V007	RCL I	
V008	+	
V009	STO I	
V010	GTO V003	
W001	LBL W	View all data points

Line	Command	Comments
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate regression coefficients
C002	ClΣ	Clear statistical registers
C003	0	Initialize the additional statistical registers
C004	STO D	
C005	STO E	
C006	STO F	
C007	STO G	
C008	1	
C009	RCL C	
C010	1E3	
C011	÷	
C012	+	
C013	STO I	Store loop index
C014	RCL (I)	Start the loop
C015	[1,0,0]	
C016	×	Get x
C017	XEQ X	Transform value of x
C018	STO X	Store transformed x
C019	RCL (I)	
C020	[0,1,0]	
C021	×	Get y
C022	XEQ Y	Transform value of y
C023	STO Y	
C024	RCL (I)	
C025	[0,0,1]	
C026	×	Get z
C027	XEQ Z	Transform value of z
C028	STO Z	
C029	RCL Y	
C030	RCL X	
C031	Σ+	Accumulate (x, y) in statistical registers

Line	Command	Comments
C032	RCL Z	
C033	STO+ D	Add to sum of z
C034	x^2	
C035	STO+ E	Add to sum of z^2
C036	RCL X	
C037	RCL Z	
C038	×	
C039	STO+ F	Add to sum of x times z
C040	RCL Y	
C041	RCL Z	
C042	×	
C043	STO+ G	Add to sum of y times z
C044	ISG I	
C045	GTO C014	Resume the next loop iteration
C046	Σx^2	Calculate Sxx
C047	Σx	
C048	x^2	
C049	n	
C050	÷	
C051	–	
C052	STO N	Store Sxx
C053	Σy^2	Calculate Syy
C054	Σ y	
C055	x^2	
C056	n	
C057	÷	
C058	–	
C059	STO O	Store Syy
C060	Σ xy	Calculate Sxy
C061	Σx	
C062	Σy	
C063	×	
C064	n	
C065	÷	
C066	–	
C067	STO P	Store Sxy
C068	RCL F	
C069	Σx	
C070	RCL D	
C071	×	
C072	n	
C073	÷	
C074	–	

Line	Command	Comments
C075	STO Q	Store Sxz
C076	RCL G	
C077	Σy	
C078	RCL D	
C079	×	
C080	n	
C081	÷	
C082	–	
C083	STO R	Store Syz
C084	RCL N	
C085	RCL O	
C086	×	
C087	RCL P	
C088	x^2	
C089	–	
C090	STO U	Store D
C091	RCL Q	
C092	RCL O	
C093	×	
C094	RCL R	
C095	RCL P	
C096	×	
C097	–	
C098	RCL U	
C099	÷	
C100	STO L	Store slope b
C101	RCL R	
C102	RCL N	
C103	×	
C104	RCL Q	
C105	RCL P	
C106	×	
C107	–	
C108	RCL U	
C109	÷	
C110	STO M	Store slope c
C111	RCL D	
C112	n	
C113	÷	
C114	RCL L	
C115	$\bar{x}$	
C116	×	
C117	–	

Line	Command	Comments
C118	RCL M	
C119	$\bar{y}$	
C120	×	
C121	–	
C122	STO K	Store intercept
C123	RCL D	Calculate r^2
C124	×	
C125	RCL L	
C126	RCL F	
C127	×	
C128	+	
C129	RCL M	
C130	RCL G	
C131	×	
C132	+	
C133	RCL D	
C134	x^2	
C135	n	
C136	÷	
C137	STO B	
C138	–	
C139	RCL E	
C140	RCL B	
C141	–	
C142	STO S	Store Szz
C143	÷	
C144	STO A	Store r^2
C145	RCL S	
C146	RCL L	
C147	RCL Q	
C148	×	
C149	–	
C150	RCL M	
C151	RCL R	
C152	×	
C153	–	
C154	n	
C155	3	
C156	–	
C157	÷	
C158	STO H	Store MSE
C159	RCL A	Push r^2 into the stack
C160	RCL K	Push the intercept into the stack

Line	Command	Comments
C161	RCL L	Push the slope b into the stack
C162	RCL M	Push the slope c into the stack
C163	SF 10	
C164	R2,A, B, C	This is a “tag” created using EQN button
C165	PSE	Display output tag for intercept, slope b, and slope c
C166	CF 10	
C167	RTN	Display values of intercept, slope b, and slope c
N001	LBL N	Display components of ANOVA table
N002	RCL L	
N003	RCL Q	
N004	×	
N005	RCL M	
N006	RCL R	
N007	×	
N008	+	
N009	2	
N010	SF 10	
N011	SSR, DF	This is a “tag” created using EQN button
N012	PSE	Display output tag for regression df and SSR
N013	STOP	Display regression df and SSR
N014	÷	Calculate mean square for regression
N015	MSR	This is a “tag” created using EQN button
N016	PSE	Display output tag for MSR
N017	STOP	Display MSR
N018	RCL H	
N019	÷	Calculate F statistic
N020	F STAT	This is a “tag” created using EQN button
N021	PSE	Display output tag for F statistic
N022	STOP	Display F statistic
N023	RCL H	
N024	n	
N025	3	
N026	–	
N027	×	Calculate residual sum of squares (SSE)
N028	LASTx	Calculate degrees of freedom for SSE
N029	SSE, DF	This is a “tag” created using EQN button
N030	PSE	Display output tag for SSE and DF
N031	STOP	Display values ofSSE and DF
N032	÷	Calculate mean square for residual (MSE)
N033	MSE	This is a “tag” created using EQN button
N034	PSE	Display output tag for MSE
N035	STOP	Display value of MSE
N036	RCL S	Get total sum of squares
N037	n	

Line	Command	Comments
N038	1	
N039	–	Calculate df for total sum of squares
N040	TSS, DF	This is a “tag” created using EQN button
N041	PSE	Display output tag for TSS and its degrees of freedom
N042	CF 10	
N043	RTN	Display values of TSS and its degrees of freedom
F001	LBL F	Confidence intervals for intercept and slopes
F002	INPUT T	Get inverse Student-t
F003	RCL O	
F004	$\bar{x}$	
F005	STO B	
F006	x^2	
F007	×	
F008	RCL N	
F009	$\bar{y}$	
F010	STO× B	
F011	x^2	
F012	×	
F013	+	
F014	RCL P	
F015	2	
F016	×	
F017	RCL B	
F018	×	
F019	-	
F020	RCL U	
F021	÷	
F022	n	
F023	1/x	
F024	+	
F025	RCL H	
F026	×	
F027	√x	
F028	RCL T	
F029	×	
F030	STO B	Store delta interval for intercept
F031	RCL K	
F032	+	Calculate upper limit for intercept
F033	RCL K	
F034	RCL B	
F035	–	Calculate lower limit for intercept
F036	SF 10	
F037	INTR UL,LL	This is a “tag” created using EQN button

Line	Command	Comments
F038	PSE	Display output tag
F039	STOP	Display the upper and lower limits of the intercept
F040	RCL O	
F041	RCL U	
F042	÷	
F043	RCL H	
F044	×	
F045	√x	
F046	RCL T	
F047	×	
F048	STO B	Store delta interval for slope b
F049	RCL L	
F050	+	Calculate upper limit for slope b
F051	RCL L	
F052	RCL B	
F053	–	Calculate lower limit for slope b
F054	B UL, B LL	This is a “tag” created using EQN button
F055	PSE	Display output tag
F056	STOP	Display the upper and lower limits of slope b
F057	RCL N	
F058	RCL U	
F059	÷	
F060	RCL H	
F061	×	
F062	√x	
F063	RCL T	
F064	×	
F065	STO B	Store delta interval for slope c
F066	RCL M	
F067	+	Calculate upper limit for slope c
F068	RCL M	
F069	RCL B	
F070	–	Calculate lower limit for slope c
F071	C UL, C LL	This is a “tag” created using EQN button
F072	PSE	Display output tag for upper limit and lower limit for slope c
F074	CF 10	
F074	RTN	Display values for upper limit and lower limit for slope c
Q001	LBL Q	Project a new x onto y
Q002	SF 2	
Q003	GTO P003	
P001	LBL P	Project an existing x onto y
P002	CF 2	
P003	INPUT T	Get inverse Student-t

Line	Command	Comments
P004	INPUT X	Get value of x
P005	XEQ X	Transform x
P006	STO X	
P007	INPUT Y	Get value of y
P008	XEQ Y	Transform y
P009	STO Y	
P010	RCL X	
P011	$\bar{x}$	
P012	–	
P013	STO B	
P014	x^2	
P015	RCL O	
P016	×	
P017	RCL Y	
P018	$\bar{y}$	
P019	–	
P020	STO× B	
P021	x^2	
P022	RCL N	
P023	×	
P024	+	
P025	RCL B	
P026	RCL P	
P027	×	
P028	2	
P029	×	
P030	–	
P031	RCL U	
P032	÷	
P033	n	
P034	1/x	
P035	FS? 2	
P036	1	
P037	FS? 2	
P038	+	
P039	CF 2	
P040	RCL H	
P041	×	
P042	√x	
P043	RCL T	
P044	×	
P045	STO B	
P046	RCL K	

Line	Command	Comments
P047	RCL L	
P048	RCL X	
P049	×	
P050	+	
P051	RCL M	
P052	RCL Y	
P053	×	
P054	+	
P055	STO Z	
P056	RCL B	
P057	+	
P058	SF 3	
P059	XEQ Z	
P060	STO V	
P061	RCL Z	
P062	RCL B	
P063	–	
P064	XEQ Z	
P065	STO W	
P066	RCL Z	
P067	XEQ Z	
P068	STO J	
P069	CF 3	
P070	RCL V	Push upper limit of $\hat{z}$ into the stack
P071	RCL J	Push value of $\hat{z}$ into the stack
P072	RCL W	Push lower limit of $\hat{z}$ into the stack
P073	SF 10	
P074	Z^U,Z^,Z^L	This is a “tag” created using EQN button
P075	PSE	Display output tag for upper limit of $\hat{z}$, $\hat{z}$, and lower limit of $\hat{z}$
P076	CF 10	
P077	RTN	Display values for upper limit of $\hat{z}$, $\hat{z}$, and lower limit of $\hat{z}$
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT Y	Prompt user for y value
G004	INPUT Z	Prompt user for z value
G005	[X,Y,Z]	Create data point. Use the EQN key to insert this line.
G006	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-MLR1	This is a “tag” created using EQN button

Line	Command	Comments
U004	PSE	Display program name
U005	CF 10	
U006	RTN	

Example

Let's look at an example that exercises various routines in the regression program. The next table contains (x, y, z) data points that we want to fit in a multiple linear regression model.

x	y	z
0	0	1.2
1	0	3.3
0	1	-2.6
2	0	5.1
2	1	1.9
0	2	-4.8
1	2	-2.9
2	2	-0.7

After you enter the above data, perform the following steps:

1. View all of the data points.
2. Calculate the regression slope, intercept, and r^2.
3. Calculate the regression ANOVA table.
4. Calculate confidence interval for the intercept and slopes at 95% confidence.
5. Calculate the projected value of z and its confidence interval for x = 2 and y = 2, at a 95% confidence level.

The following table shows the steps performed. The initial display mode of the program is FIX 2.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Add the first data point.		XEQ A	X?
	Enter a value of x.	0	R/S	Y?
	Enter a value of y.	0	R/S	Z?
	Enter a value of z.	1.2	R/S	1
3	Add the second data point		XEQ A	X?
	Enter a value of x	1	R/S	Y?
	Enter a value of y	0	R/S	Z?
	Enter a value of z.	3.3	R/S	2
4	Repeat step 3 for all other data point			8
5	View all data points (Note the program displays the results using the PSE statement). If you spot an error in the data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	8.01 [2.00,2.00,-0.70]
6	Calculate the regression coefficients.		XEQ C	2.08 -3.00 Display slope b and slope c
			R↓	
			R↓	0.99 0.98 Display r^2 and intercept
	The fitted line is: z = 0.9917 – 2.08333 x – 3.000 y, with r^2 of 0.99284			
	Switch to the FIX 5 display mode.		FIX 5	
7	Calculate the regression ANOVA table.		XEQ N	80.04167 2.00000 Shows the Regression sum of squares (SSR) and the Regression DF

Step #	Step/Substep	Input	Command	Output
			R/S	0.97917 40.02083 Shows the Mean Square for regression (MSR)
			R/S	0.97917 346.75090 Shows the calculated F statistic
			R/S	0.57708 5.00000 Shows the Residual sum of squares (SSE) and the Residual df
			R/S	346.75090 0.11542 Shows the Mean Square for residuals (MSE)
			R/S	80.61875 7.00000 Shows the Total Sum of Squares and the Total df
8	Calculate confidence intervals.		XEQ F	T?
	Enter the value of the inverse Student-t with 5 degrees of freedom and a probability of 0.025.	2.571	R/S	1.57049 0.38784 Shows the Intercept confidence interval
			R/S	2.43992 1.72675 Shows the confidence interval for the slope b
			R/S	-2.64342 -3.35658 Shows the confidence interval for the slope c
9	Project an existing x and y onto z.		XEQ P	T?
	Enter the value of the inverse	2.571	R/S	X?

Step #	Step/Substep	Input	Command	Output
	Student-t with 5 degrees of freedom and a probability of 0.025.			
	Enter value of x.	2	R/S	Y?
	Enter value of y.	2	R/S	-0.85417 -1.16298 Shows $\hat{z}$ and $\hat{z}$ lower limit
			R↓	-0.54536 -0.85417 Shows $\hat{z}$ upper limit and $\hat{z}$

If you desire to apply data transformations to the x, y and/or z observations you need to edit the labels X, Y, and Z. For example, if you wish to use a power fit z = a (x^b) (y^c), then you need to have the labels X, Y, and Z as follows:

```
X001 LBL X
X002 LN
X003 RTN
Y001 LBL Y
Y002 LN
Y003 RTN
Z001 LBL Z
Z002 FS?3
Z003 GTO Z006
Z004 LN              # transform z
Z005 RTN
Z006 EXP             # inverse transform z
Z007 RTN
```

The above fits the sought model as the following linearized form:

ln(z) = a' + b ln(x) + c ln(y)

Where a' = ln(a). Notice the use of flag 3 to manage the inverse transformation needed in calculating projected values of z.

As another example, if you wish to use a power fit √z= a + b/x + c ln(y), then you need to have the labels X, Y, and Z as follows:

```
X001 LBL X
X002 1/x
X003 RTN
```

```
Y001 LBL Y
Y002 LN
Y003 RTN
Z001 LBL Z
Z002 FS?3
Z003 GTO Z006
Z004 √x            # transform z
Z005 RTN
Z006 x^2           # inverse transform z
Z007 RTN
```

As a third example, if you wish to use a power fit 1/z = a + b/x + c/y, then you need to have the labels X, Y, and Z as follows:

```
X001 LBL X
X002 1/x
X003 RTN
Y001 LBL Y
Y002 1/x
Y003 RTN
Z001 LBL Z
Z002 1/x
Z003 RTN
```

Notice that the first and last examples, the code in LBL Z does not use flag 3 since keeping the values of variable z linear, and taking their reciprocals, each has the same mathematical inverses. The above code can be reduced as follows:

```
X001 LBL X
Y001 LBL Y
Z001 LBL Z
Z002 1/x
Z003 RTN
```

Appendix A contains a new approximation that I recently developed for inverse Student-t function. The appendix also contains tables for selected values.

Multiple Linear Regression Program NCS_MLR2

Overview

The NCS_MLR2 program implements multiple linear regression between *two* variables. They are the dependent variable Z and the independent X. The regression model uses two transformations for variable X, as shown below:

f(Z) = a + b g(X) + c h(X)

Where f(Z), g(X), and h(X) are functions that transform the variables X and Z. In addition, the function $f^{-1}(Z)$ is the inverse transformation of function f(Z). The function $f^{-1}(Z)$ must be defined and easy to evaluate. The functions g(X) and h(X) must be distinct enough to avoid creating redundant terms. For example, the functions g(X) = 2X and h(X) =X + 1, essentially create a model with identical terms that cause colinearity in the regression.

If we define the transformed variables x, y, and z, such that z = f(Z), x = g(X), and y = h(X), then the above equation takes on a more familiar form:

z = a + b x + c y

The program offers the following features:

1. Storing data points once and reusing them for different regression scenarios. The program stores the (X, Z) data points as 2-D vectors, a new feature of the HP-35s, thus reducing the number of unnamed memory registers used by one half.
2. The ability to delete, edit, view, and swap data points. Thus the regression program incorporates basic operations to manage the data points.
3. Transforming the X and Z observations. You can edit the program to implement your custom transformations and then recalculate the regression statistics without having to reenter the data points.
4. Calculating the regression intercept, slopes, and coefficient of determination.
5. Calculating the confidence intervals for intercept and slopes.
6. Calculating the regression ANOVA table.
7. Calculating projected value of Z for existing or new values of X.

You can regard the program NCS_POLY1 as a special case of program NCS_MLR2. Since quadratic fitting is quite popular, we can justify the specialized curve fitting performed by program NCS_POLY1.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Data Collected

z = f(Z)
x = h(X)
y = g(X)
$\sum x$ = sum of x
$\sum x^2$ = sum of x^2
$\sum y$ = sum of y
$\sum y^2$ = sum of y^2
$\sum z$ = sum of z
$\sum z^2$ = sum of z^2
$\sum xy$ = sum of $x \cdot y$
$\sum xz$ = sum of $x \cdot z$
$\sum yz$ = sum of $y \cdot z$
n = number of observations

Regression Coefficients

$x_m = \sum x / n$
$y_m = \sum y / n$
$z_m = \sum z / n$
$S_{xx} = \sum x^2 - (\sum x)^2 / n = \sum x^2 - n\,(x_m)^2$
$S_{yy} = \sum y^2 - (\sum y)^2 / n = \sum y^2 - n\,(y_m)^2$
$S_{zz} = \sum z^2 - (\sum z)^2 / n = \sum z^2 - n\,(z_m)^2$
$S_{xy} = \sum xy - (\sum x)(\sum y)/ n = \sum xy - n\, x_m\, y_m$
$S_{xz} = \sum xz - (\sum x)(\sum z)/ n = \sum xz - n\, x_m\, z_m$
$S_{yz} = \sum yz - (\sum y)(\sum z)/ n = \sum yz - n\, y_m\, z_m$

$D = S_{xx}\, S_{yy} - (S_{xy})^2$
Slope $b = [S_{xz}\, S_{yy} - S_{yz}\, S_{xy}] / [S_{xx}\, S_{yy} - (S_{xy})^2]$
Slope $c = [S_{yz}\, S_{xx} - S_{xz}\, S_{xy}] / [S_{xx}\, S_{yy} - (S_{xy})^2]$
Intercept $a = z_m - b\, x_m - c\, y_m$
$R^2 = [a \sum z + b \sum xz + c \sum yz - (\sum z)^2 / n] / [\sum z^2 - (\sum z)^2 / n\,]$
$\quad = [a \sum z + b \sum xz + c \sum yz - (\sum z)^2 / n] / S_{zz}$

ANOVA Table

Source of Variation	***Sum of Squares***	***Degrees of Freedom***	***Mean Square***	***F***
Regression	$SS_R = b\, S_{xz} + c\, S_{yz}$	2	MS_R	MS_R / MS_E
Residual	$SS_E = S_{zz} - b\, S_{xz} - c\, S_{yz}$	n – 1 – 2 = n – 3	MS_E	
Total	$SS_T = S_{zz}$	n – 1		

Interval for Slope b

At 100(1 – α) confidence:
$D = S_{xx}\, S_{yy} - (S_{xy})^2$

SlopeStdErr = $\sqrt{[MS_E (S_{yy} / D)]}$
$\Delta b = t_{\alpha/2,n-3} \cdot$ SlopeStdErr

Upper Range: $b + \Delta b$
Lower Range: $b - \Delta b$

Interval for Slope c

At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$
SlopeStdErr = $\sqrt{[MS_E (S_{xx} / D)]}$
$\Delta c = t_{\alpha/2,n-3} \cdot$ SlopeStdErr

Upper Range: $c + \Delta c$
Lower Range: $c - \Delta c$

Interval for Intercept a

At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$
IntStdErr = $\sqrt{(MS_E [1/n + (S_{yy} / D)(x_m)^2 + (S_{xx} / D)(y_m)^2 - 2(S_{xy} / D) x_m y_m])}$
$= \sqrt{(MS_E [1/n + \{S_{yy} (x_m)^2 + S_{xx} (y_m)^2 - 2(S_{xy} x_m y_m\}]/D)}$

$\Delta a = t_{\alpha/2,n-3} \cdot$ IntStdErr

Upper Range: $a + \Delta a$
Lower Range: $a - \Delta a$

Interval for Projected Z

$x_0 = g(X_0)$
$y_0 = h(X_0)$
At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$

$\hat{z} = a + b\, x_0 + c\, y_0$

zhatStdErr = $\sqrt{(MS_E [1/n + (S_{yy} / D)(x_0 - x_m)^2 + (S_{xx} / D)(y_0 - y_m)^2 - 2 (S_{xy} / D) (x_0 - x_m) (y_0 - y_m)])}$
$= \sqrt{(MS_E [1/n + \{S_{yy} (x_0 - x_m)^2 + S_{xx} (y_0 - y_m)^2 - 2 S_{xy} (x_0 - x_m) (y_0 - y_m)\} / D])}$

$\Delta\hat{z} = t_{\alpha/2,n-3} \cdot$ zhatStdErr
Upper Range: $\hat{z} + \Delta\hat{z}$
Lower Range: $\hat{z} - \Delta\hat{z}$

$\hat{Z} = f^{-1}(\hat{z})$
Upper limit $\hat{Z} = f^{-1}$(upper limit $\hat{z}$)
Lower limit $\hat{Z} = f^{-1}$(Lower limit $\hat{z}$)

For a *new* observation:

zhatStdErr = $t_{\alpha/2,n-3} \cdot \sqrt{((MS_E)^2 [1+1/n + (S_{yy} / D)(x_0 - x_m)^2 + (S_{xx} / D)(y_0 - y_m)^2 - 2 (S_{xy} / D) (x_0 - x_m) (y_0 - y_m)])}$

$= \sqrt{((MS_E)^2 [1 + 1/n + \{S_{yy} (x_0 - x_m)^2 + S_{xx} (y_0 - y_m)^2 - 2 S_{xy} (x_0 - x_m) (y_0 - y_m)\} / D])}$

$\Delta\hat{z} = t_{\alpha/2,n-3} \cdot$ zhatStdErr
Upper Range: $\hat{z} + \Delta\hat{z}$
Lower Range: $\hat{z} - \Delta\hat{z}$

$\hat{Z} = f^{-1}(\hat{z})$
Upper limit $\hat{Z} = f^{-1}$(upper limit $\hat{z}$)
Lower limit $\hat{Z} = f^{-1}$(Lower limit $\hat{z}$)

Instructions

The following table shows the steps involved in using the various parts of the regression program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value of X.	X	R/S	Y?		
	Enter a value of Z.	Z	R/S	Number of data points		
3	Delete a data point.		XEQ D	I?		
	Enter the index of the data point to delete. If you enter	Index of data point to delete	R/S	Number of data points		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	an invalid index the program displays the message INVALID DATA.					
4	Edit a data point.		XEQ E	I?		
	Enter the index of the data point to edit. If you enter an invalid index the program displays the message INVALID DATA.		R/S	X?		
	Enter a new value for X.	X	R/S	Z?		
	Enter a new value for Z.	Z	R/S			
5	View a data point.		XEQ V	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	Data point	Index	
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	Data point	Index	
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1		
6	View all data points (Note the program displays the results using		XEQ W	Data point	Index	

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	the PSE statement.)					
7	Swap data points.		XEQ S	I?		
	Enter the index of the first data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the first data point	R/S	J?		
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S			
8	Calculate the regression coefficients (note that the value of r^2 is in the T stack register).		XEQ C	Slope c	Slope b	Intercept
	Use the R↓ key to scroll down the stack and examine the values of the four results.		R↓	Slope b	Intercept	r^2
			R↓	Intercept	r^2	Slope c
			R↓	r^2	Slope c	Slope b
			R↓	Slope c	Slope b	Intercept
9	Calculate the regression ANOVA table.		XEQ N	Regression DF	Regression sum of squares (SSR)	
			R/S	Mean Square for regression (MSR)		
			R/S	Calculated F statistic		
			R/S	Residual df	Residual sum of	

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
					squares (SSE)	
			R/S	Mean Square for residuals (MSE)		
			R/S	Total df	Total Sum of Squares	
10	Calculate the confidence intervals.		XEQ F	T?		
	Enter a value of the inverse Student-t probability distribution function.	Student-t for n–3 degrees of freedom and an α/2 significance (α is based on 1–α confidence level.)	R/S	Intercept lower limit	Interval upper limit	
			R/S	Slope b lower limit	Slope b upper limit	
			R/S	Slope c lower limit	Slope c upper limit	
11	Project an existing x and y onto z.		XEQ P	T?		
	Enter a value of the inverse Student-t probability distribution function.	Student-t for n–3 degrees of freedom and an α/2 significance (α is based on 1–α confidence level.)	R/S	X?		
		X	R/S	$\hat{Z}$ lower limit	$\hat{Z}$	$\hat{Z}$ upper limit
12	Project a new x and y onto z.		XEQ Q	T?		
	Enter a value of the inverse Student-t probability distribution	Student-t for n–3 degrees of freedom and an α/2 significance	R/S	X?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	function.	(α is based on 1–α confidence level.)				
		X	R/S	$\hat{Z}$ lower limit	$\hat{Z}$	$\hat{Z}$ upper limit
13	You can temporarily (or even permanently) reduce the number of data points entering in the regression calculations.		RCL C	Number of data points		
	Record the number of original data points.					
	Enter a lower value of the number of data points (must be greater than 4.)	New number of data points	STO C			
14	You can restore the original number of data points entering in the regression calculations.	Original number of data points	STO C			
15	Alter the transformation for X.		GTO X001			
	Switch to program mode.		PRGM	X001 LBL X		
	Edit the commands between LBL X and the RTN statement to reflect the new g(X) transformation you desire.					
	Exit program		PRGM			

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	mode when you are done.					
16	Alter the h(X) transformation.		GTO Y001			
	Switch to program mode.		PRGM	Y001 LBLY		
	Edit the commands between LBL Y and the RTN statement to reflect the new h(X) transformation you desire.					
	Exit program mode when you are done.		PRGM			
17	Alter the inverse transformation for f(Z).		GTO Z001			
	Switch to program mode.		PRGM	Z001 LBL Z		
	Edit the commands between LBL Z and the RTN statement to reflect the new f(Z) transformation and inverse transformation you desire. Flag 3 is dedicated to distinguish between the transformation and inverse transformation. When flag 3 is set, LBL Z performs an inverse transformation.					

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	Exit program mode when you are done.		PRGM			
18	To display the program name. The program pauses to display the program name.		XEQ U	NCS-MLR2		

Comments on Runtime Errors

The program validates the indices that you enter to manage data points to prevent the corruption of the data points. Aside from this feature, you are responsible to enter data that will avoid runtime errors. For example, if you are using the square root or logarithms to transform the x and/or y data, then entering data points with negative values will raise runtime errors. Likewise, using reciprocals and logarithms will generate runtime errors for data that are zeros. Entering redundant data points also has its peril of giving you meaningless results. Remember the old saying about program input and output data "Junk in, junk out."

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A	Used, r^2	1	Index invalid when set
D	Delete an existing data point	B	Used	2	Project a new x and y onto z when set
E	Edit a data point	C	Count for number of data points	3	Inverse transform z
V	View a data point	D	Σz		
W	View all data points	E	Σz^2		
I	Initialize	F	Σxz		
S	Swap two data points	G	Σyz		
C	Calculate regression coefficients	H	MSE		
N	Calculate regression ANOVA	I	Index, used		
F	Calculate	J	Index, used		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
	confidence intervals				
P	Project an observed x and y onto z with confidence interval	K	Intercept a		
Q	Project a new x and y onto z with confidence interval	L	Slope b		
X	Transform x data	M	Slope c		
Y	Transform y data	N	Sxx		
Z	Transform and Inverse transform of z data	O	Syy		
U	Display program name	P	Sxy		
Labels for internal use		Q	Sxz		
H	Validate indices and display error message if the an index is invalid	R	Sxy		
G	Prompt for x, y and z	S	Szz		
R	Display the message INVALID DATA	T	Inverse Student-t pdf		
		U	$D = S_{xx} S_{yy} – (S_{xy})^2$		
		V	Upper limit		
		W	Lower limit		
		X	x		
		Y	y		
		Z	z		
		1	Data point #1		
		2	Data point #2		
		3	Data point #3		
		…			
		#C	Data point #C		

Program Listing

Line	Command	Comments
X001	LBL X	Label to contain the transformation for g(X). No code is needed, between the label and the RTN command, for linear values.
X002	RTN	
Y001	LBL Y	Label to contain the transformation for h(X). No code is needed, between the label and the RTN command, for linear values.
Y002	RTN	
Z001	LBL Z	Label to contain the transformation and inverse transformation for Z. No code is needed, between the label and the RTN command, for linear values.
Z002	RTN	
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	STOP	
A011	GTOA001	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	

Line	Command	Comments
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO– C	
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	Validate index
S006	RCL (I)	Recall data points and place them on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x↔y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	RCL (I)	Recall data point
V005	STOP	
V006	1	
V007	RCL I	
V008	+	
V009	STO I	
V010	GTO V003	
W001	LBL W	View all data points

Line	Command	Comments
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate regression coefficients
C002	ClΣ	Clear statistical registers
C003	0	Initialize the additional statistical registers
C004	STO D	
C005	STO E	
C006	STO F	
C007	STO G	
C008	1	
C009	RCL C	
C010	1E3	
C011	÷	
C012	+	
C013	STO I	Store loop index
C014	RCL (I)	Start the loop
C015	[1,0]	
C016	×	Get X
C017	STO Y	Store X in register Y
C018	XEQ X	Transform value of X
C019	STO X	Store transformed X → x
C020	RCL Y	Get X again
C021	XEQ Y	Transform value of X → y
C022	STO Y	
C023	RCL (I)	
C024	[0,1]	
C025	×	Get Z
C026	XEQ Z	Transform value of Z → z
C027	STO Z	
C028	RCL Y	
C029	RCL X	
C030	Σ+	Accumulate (x, y) in statistical registers
C031	RCL Z	
C032	STO+ D	Add to sum of z

Line	Command	Comments
C033	x^2	
C034	STO+ E	Add to sum of z^2
C035	RCL X	
C036	RCL Z	
C037	×	
C038	STO+ F	Add to sum of x times z
C039	RCL Y	
C040	RCL Z	
C041	×	
C042	STO+ G	Add to sum of y times z
C043	ISG I	
C044	GTO C014	Resume the next loop iteration
C045	Σx^2	Calculate Sxx
C046	Σx	
C047	x^2	
C048	n	
C049	÷	
C050	–	
C051	STO N	Store Sxx
C052	Σy^2	Calculate Syy
C053	Σ y	
C054	x^2	
C055	n	
C056	÷	
C057	–	
C058	STO O	Store Syy
C059	Σ xy	Calculate Sxy
C060	Σx	
C061	Σy	
C062	*	
C063	n	
C064	÷	
C065	–	
C066	STO P	Store Sxy
C067	RCL F	
C068	Σx	
C069	RCL D	
C070	×	
C071	n	
C072	÷	
C073	–	
C074	STO Q	Store Sxz
C075	RCL G	

Line	Command	Comments
C076	Σy	
C077	RCL D	
C078	*	
C079	n	
C080	÷	
C081	–	
C082	STO R	Store Syz
C083	RCL N	
C084	RCL O	
C085	×	
C086	RCL P	
C087	x^2	
C088	–	
C089	STO U	Store D
C090	RCL Q	
C091	RCL O	
C092	×	
C093	RCL R	
C094	RCL P	
C095	×	
C096	–	
C097	RCL U	
C098	÷	
C099	STO L	Store slope b
C100	RCL R	
C101	RCL N	
C102	×	
C103	RCL Q	
C104	RCL P	
C105	×	
C106	–	
C107	RCL U	
C108	÷	
C109	STO M	Store slope c
C110	RCL D	
C111	n	
C112	÷	
C113	RCL L	
C114	$\bar{x}$	
C115	×	
C116	–	
C117	RCL M	
C118	$\bar{y}$	

Line	Command	Comments
C119	×	
C120	–	
C121	STO K	Store intercept
C122	RCL D	Calculate r^2
C123	×	
C124	RCL L	
C125	RCL F	
C126	×	
C127	+	
C128	RCL M	
C129	RCL G	
C130	×	
C131	+	
C132	RCL D	
C133	x^2	
C134	n	
C135	÷	
C136	STO B	
C137	–	
C138	RCL E	
C139	RCL B	
C140	–	
C141	STO S	Store Szz
C142	÷	
C143	STO A	Store r^2
C144	RCL S	
C145	RCL L	
C146	RCL Q	
C147	×	
C148	–	
C149	RCL M	
C150	RCL R	
C151	×	
C152	–	
C153	n	
C154	3	
C155	–	
C156	÷	
C157	STO H	Store MSE
C158	RCL A	Push r^2 into the stack
C159	RCL K	Push the intercept into the stack
C160	RCL L	Push the slope b into the stack
C161	RCL M	Push the slope c into the stack

Line	Command	Comments
C162	SF 10	
C163	R2,A, B, C	This is a "tag" created using EQN button
C164	PSE	Display output tag for intercept, slope b, and slope c
C165	CF 10	
C166	RTN	Display values of intercept, slope b, and slope c
N001	LBL N	Display components of ANOVA table
N002	RCL L	
N003	RCL Q	
N004	×	
N005	RCL M	
N006	RCL R	
N007	×	
N008	+	
N009	2	
N010	SF 10	
N011	SSR, DF	This is a "tag" created using EQN button
N012	PSE	Display output tag for regression df and SSR
N013	STOP	Display regression df and SSR
N014	÷	Calculate mean square for regression
N015	MSR	This is a "tag" created using EQN button
N016	PSE	Display output tag for MSR
N017	STOP	Display MSR
N018	RCL H	
N019	÷	Calculate F statistic
N020	F STAT	This is a "tag" created using EQN button
N021	PSE	Display output tag for F statistic
N022	STOP	Display F statistic
N023	RCL H	
N024	n	
N025	3	
N026	–	
N027	×	Calculate residual sum of squares (SSE)
N028	LASTx	Calculate degrees of freedom for SSE
N029	SSE, DF	This is a "tag" created using EQN button
N030	PSE	Display output tag for SSE and DF
N031	STOP	Display values of SSE and DF
N032	÷	Calculate mean square for residual (MSE)
N033	MSE	This is a "tag" created using EQN button
N034	PSE	Display output tag for MSE
N035	STOP	Display value of MSE
N036	RCL S	Get total sum of squares
N037	n	
N038	1	
N039	–	Calculate df for total sum of squares

Line	Command	Comments
N040	TSS, DF	This is a “tag” created using EQN button
N041	PSE	Display output tag for TSS and its degrees of freedom
N042	CF 10	
N043	RTN	Display values of TSS and its degrees of freedom
F001	LBL F	Confidence intervals for intercept and slopes
F002	INPUT T	Get inverse Student-t
F003	RCL O	
F004	$\bar{x}$	
F005	STO B	
F006	x^2	
F007	×	
F008	RCL N	
F009	$\bar{y}$	
F010	STO× B	
F011	x^2	
F012	×	
F013	+	
F014	RCL P	
F015	2	
F016	×	
F017	RCL B	
F018	×	
F019	-	
F020	RCL U	
F021	÷	
F022	n	
F023	1/x	
F024	+	
F025	RCL H	
F026	×	
F027	√x	
F028	RCL T	
F029	×	
F030	STO B	Store delta interval for intercept
F031	RCL K	
F032	+	Calculate upper limit for intercept
F033	RCL K	
F034	RCL B	
F035	–	Calculate lower limit for intercept
F036	SF 10	
F037	INTRC UL,LL	This is a “tag” created using EQN button
F038	PSE	Display output tag
F039	STOP	Display upper and lower limits for the intercept

Line	Command	Comments
F040	RCL O	
F041	RCL U	
F042	÷	
F043	RCL H	
F044	×	
F045	√x	
F046	RCL T	
F047	×	
F048	STO B	Store delta interval for slope b
F049	RCL L	
F050	+	Calculate upper limit for slope b
F051	RCL L	
F052	RCL B	
F053	–	Calculate lower limit for slope b
F054	B UL, B LL	This is a “tag” created using EQN button
F055	PSE	Display output tag
F056	STOP	Display upper and lower limits for slope b
F057	RCL N	
F058	RCL U	
F059	÷	
F060	RCL H	
F061	×	
F062	√x	
F063	RCL T	
F064	×	
F065	STO B	Store delta interval for slope c
F066	RCL M	
F067	+	Calculate upper limit for slope c
F068	RCL M	
F069	RCL B	
F070	–	Calculate lower limit for slope c
F071	C UL, C LL	This is a “tag” created using EQN button
F072	PSE	Display output tag for upper limit and lower limit for slope c
F073	CF 10	
F074	RTN	Display values for upper limit and lower limit for slope c
Q001	LBL Q	Project a new x onto y
Q002	SF 2	
Q003	GTO P003	
P001	LBL P	Project an existing x onto y
P002	CF 2	
P003	INPUT T	Get inverse Student-t
P004	INPUT X	Get value of X
P005	STO Y	Store X in register Y temporarily

Line	Command	Comments
P006	XEQ X	Transform X to x
P007	STO X	
P008	RCL Y	Get value of X again from register Y
P009	XEQ Y	Transform X to y
P010	STO Y	
P011	RCL X	
P012	$\bar{x}$	
P013	–	
P014	STO B	
P015	x^2	
P016	RCL O	
P017	×	
P018	RCL Y	
P019	$\bar{y}$	
P020	–	
P021	STO× B	
P022	x^2	
P023	RCL N	
P024	×	
P025	+	
P026	RCL B	
P027	RCL P	
P028	×	
P029	2	
P030	×	
P031	–	
P032	RCL U	
P033	÷	
P034	n	
P035	1/x	
P036	FS? 2	
P037	1	
P038	FS? 2	
P039	+	
P040	CF 2	
P041	RCL H	
P042	×	
P043	√x	
P044	RCL T	
P045	×	
P046	STO B	
P047	RCL K	
P048	RCL L	

Line	Command	Comments
P049	RCL X	
P050	×	
P051	+	
P052	RCL M	
P053	RCL Y	
P054	×	
P055	+	
P056	STO Z	
P057	RCL B	
P058	+	
P059	SF 3	
P060	XEQ Z	
P061	STO V	
P062	RCL Z	
P063	RCL B	
P064	–	
P065	XEQ Z	
P066	STO W	
P067	RCL Z	
P068	XEQ Z	
P069	STO J	
P070	CF 3	
P071	RCL V	Push upper limit of $\hat{z}$ into the stack
P072	RCL J	Push value of $\hat{z}$ into the stack
P073	RCL W	Push lower limit of $\hat{z}$ into the stack
P074	SF 10	
P075	Z^U,Z^,Z^L	This is a "tag" created using EQN button
P076	PSE	Display output tag for upper limit of $\hat{z}$, $\hat{z}$, and lower limit of $\hat{z}$
P077	CF 10	
P078	RTN	Display values for upper limit of $\hat{z}$, $\hat{z}$, and lower limit of $\hat{z}$
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT Z	Prompt user for z value
G004	[X,Z]	Create data point. Use the EQN key to insert this line.
G005	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-MLR2	This is a "tag" created using EQN button
U004	PSE	Display program name
U005	CF 10	

Line	Command	Comments
U006	RTN	

Example

Let's look at an example that exercises various routines in the regression program. The next table contains (X, Z) data points that we want to fit using the following multiple linear regression model:

Z = a + b X + c / X

x	z
1	-5
2	3
3	8
4	12
5	15
6	18
7	22
8	25

After you enter the above data, perform the following steps:

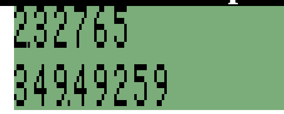
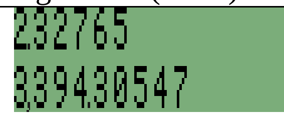
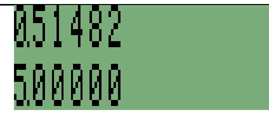
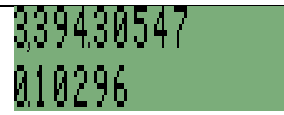
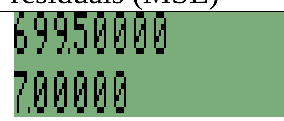
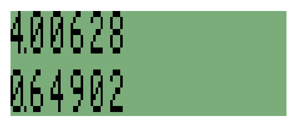
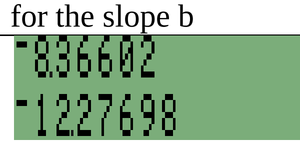
1. Edit LBL Y to insert the command 1/x in Y002.
2. View all of the data points.
3. Calculate the regression slope, intercept, and r^2.
4. Calculate the regression ANOVA table.
5. Calculate confidence interval for the intercept and slopes at 95% confidence.
6. Calculate the projected value of Z and its confidence interval for X = 2, at a 95% confidence level.

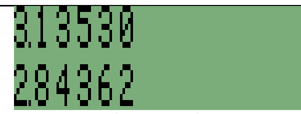
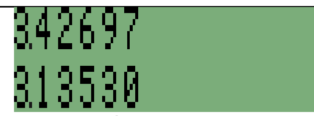
The following table shows the steps performed. The program assumes that the calculator is initially in the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Edit the program. First, switch to program mode.		PRGM	
	Go to label Y. Note I am using [.] to refer to the decimal key.		GTO [.] Y001	
	Insert the		1/x	

Step #	Step/Substep	Input	Command	Output
	reciprocal function. Make sure that this command is followed by RTN. If there are other commands before RTN, delete them.			
	Go to label X. Note I am using [.] to refer to the decimal key.		GTO [.] X001	
	Make sure that label X is immediately followed by the RTN command. If there are other commands before the RTN statement, delete them.			
	Go to label Z. Note I am using [.] to refer to the decimal key.		GTO [.] Z001	
	Make sure that label Z is immediately followed by the RTN command. If there are other commands before the RTN statement, delete them.			
	Switch out of program mode.		PRGM	
2	Initialize the program.		XEQ I	0
3	Add the first data point.		XEQ A	X?
	Enter a value of X.	1	R/S	Z?
	Enter a value of Z.	-5	R/S	1
4	Add the second		XEQ A	X?

Step #	Step/Substep	Input	Command	Output
	data point			
	Enter a value of X	2	R/S	Z?
	Enter a value of Z	3	R/S	2
5	Repeat step 3 for all other data point			8
6	View all data points (Note the program displays the results using the PSE statement). If you spot an error in the data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	8.01 [8.00, 25.00]
7	Calculate the regression coefficients.		XEQ C	2.98 -10.32 Display slope b and slope c
			R↓	
			R↓	1.00 2.33 Display r^2 and intercept
	The fitted line is: Z = 2.3276 + 2.9842 X – 10.3215/X, with r^2 of 0.99926			
	Switch to the FIX 5 display mode.		FIX 5	
8	Calculate the regression ANOVA table.		XEQ N	698.98518 2.00000 Shows the Regression sum of squares (SSR) and the Regression DF

Step #	Step/Substep	Input	Command	Output
			R/S	 Shows the Mean Square for regression (MSR)
			R/S	 Shows the calculated F statistic
			R/S	 Shows the Residual sum of squares (SSE) and the Residual df
			R/S	 Shows the Mean Square for residuals (MSE)
			R/S	 Shows the Total Sum of Squares and the Total df
9	Calculate confidence intervals.		XEQ F	T?
	Enter the value of the inverse Student-t with a DF of 5 and a probability of 0.025.	2.571	R/S	 Shows the Intercept confidence interval
			R/S	 Shows the confidence interval for the slope b
			R/S	 Shows the confidence interval for the slope c
10	Project an existing		XEQ P	T?

Step #	Step/Substep	Input	Command	Output
	X onto Z			
	Enter the value of the inverse Student-t with a DF of 5 and a significance level of 0.025.	2.571	R/S	X?
	Enter value of X.	2	R/S	 Shows $\hat{Z}$ and $\hat{Z}$ lower limit
			R↓	 Shows $\hat{Z}$ upper limit and $\hat{Z}$

If you desire to apply data transformations to the X and/or Z observations you need to edit the labels X, Y, and Z. For example, if you wish to use a power fit ln(Z) = a + b X + c/X, then you need to have the labels X, Y, and Z as follows:

```
X001 LBL X
X002 LN
X003 RTN
Y001 LBL Y
Y002 1/x
Y003 RTN
Z001 LBL Z
Z002 FS?3
Z003 GTO Z006
Z004 LN             # transform z
Z005 RTN
Z006 EXP            # inverse transform z
Z007 RTN
```

As another example, if you wish to use a power fit √Z= a + b/X + c ln(X), then you need to have the labels X, Y, and Z as follows:

```
X001 LBL X
X002 1/x
X003 RTN
Y001 LBL Y
Y002 LN
Y003 RTN
Z001 LBL Z
```

```
Z002 FS?3
Z003 GTO Z006
Z004 √x                # transform z
Z005 RTN
Z006 x^2               # inverse transform z
Z007 RTN
```

As a third example, if you wish to use a power fit $1/Z = a + b X + c X^2$, then you need to have the labels X, Y, and Z as follows:

```
X001 LBL X
X002 RTN
Y001 LBL
Y002 x^2
Y003 RTN
Z001 LBL Z
Z002 1/x
Z003 RTN
```

Notice that the first and last examples, the code in LBL Z does not use flag 3 since keeping the values of variable z linear, and taking their reciprocals, each has the same mathematical inverses.

As a fourth example, if you wish to use a power fit $Z = a + b / X + c / X^2$, then you need to have the labels X, Y, and Z as follows:

```
X001 LBL X
X002 1/x
X003 RTN
Y001 LBL
Y002 1/x
Y003 x^2
Y004 RTN
Z001 LBL Z
Z002 RTN
```

Appendix A contains a new approximation that I recently developed for inverse Student-t function. The appendix also contains tables for selected values.

Moments Program NCS_MOMENTS1

Overview

The NCS_MOMENTS1 program calculates statistical moments and offers the following features:

8. Storing data points.
9. The ability to delete, edit, view, and swap data points. Thus the program incorporates basic operations to manage the data points.
10. Calculating the first moments and the coefficients of skewness and kurtosis.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

The first moment is:

$$\bar{x} = \frac{1}{n} \Sigma\, x_i$$

The second moment is:

$$m_2 = \frac{1}{n} \Sigma\, x_i^2 - \bar{x}^2$$

The third moment is:

$$m_3 = \frac{1}{n} \Sigma\, x_i^3 - \frac{3}{n}\, \bar{x}\, \Sigma\, x_i^2 + 2\bar{x}^3$$

The fourth moment is:

$$m_4 = \frac{1}{n} \Sigma\, x_i^4 - \frac{4}{n}\, \bar{x}\, \Sigma\, x_i^3 + \frac{6}{n}\, \bar{x}^2\, \Sigma\, x_i^2 - 3\bar{x}^4$$

The moment coefficient of skewness is:

$$\gamma_1 = m_3 / m_2^{1.5}$$

The moment coefficient of kurtosis is:

$$\gamma_2 = m_4 / m_2^2$$

Instructions

The following table shows the steps involved in using the various parts of the moments program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value for x.	x	R/S	Number of data points		
3	Delete a data point.		XEQ D	I?		
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message INVALID DATA.	Index of data point to delete	R/S	Number of data points		
4	Edit a data point.		XEQ E	I?		
	Enter the index of the data point to edit. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	X?		
	Enter a new value for x.	x	R/S			

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
5	View a data point.		XEQ V	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	Data point	Index	
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	Data point	Index	
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1		
6	View all data points (Note the program displays the results using the PSE statement.) Note: If you spot an error in the data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	Data point	Index	
7	Swap data points.		XEQ S	I?		
	Enter the index of the first data point. If you enter an invalid index the program displays	Index of the first data point	R/S	J?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	the message INVALID DATA.					
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S			
8	Calculate the moments. The program first displays the first two moments.		XEQ C	Second moment	First moment	
	View the third and fourth moments.		R/S	Fourth moment	Third moment	
	View the coefficients of skewness and kurtosis.		R/S	Coefficient of kurtosis	Coefficient of skewness	
9	To display the program name. The program pauses to display the program name.		XEQ U	NCS-MOMENTS1		

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A	Σx^3	1	Index invalid when set
D	Delete an existing data point	B	Σx^4		
E	Edit a data point	C	Number of data points		
V	View a data point	D	First moment $\bar{x}$		
W	View all data points	E	Second moment m_2		
I	Initialize	F	Third moment m_3		
S	Swap two data points	G	Fourth moment m_4		
C	Calculate the moments	I	Index		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
U	Display program name	J	Index		
		X	x		
		Y	Moment coefficient of skewness γ_1		
		Z	Moment coefficient of kurtosis γ_2		
		1	Data point #1		
Labels for internal use		2	Data point #2		
H	Validate indices and display error message if an index is invalid	3	Data point #3		
G	Prompt for x	…			
R	Display the message INVALID DATA	#C	Data point #C		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	STOP	
A011	GTO A001	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	

Line	Command	Comments
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO– C	
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	Validate index
S006	RCL (I)	Recall data points and place them on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x↔y	

Line	Command	Comments
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	FS? 1	Index out or range?
V005	GTO R001	
V006	RCL (I)	Recall data point
V007	STOP	
V008	1	
V009	RCL I	
V010	+	
V011	STO I	
V012	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate regression coefficients
C002	1	
C003	RCL C	
C004	1E3	
C005	÷	
C006	+	
C007	STO I	Store loop index
C008	ClΣ	Clear statistical registers
C009	0	
C010	STO A	Initialize Σx^3
C011	STO B	Initialize Σx^4
C012	RCL (I)	Start the loop
C013	STO X	
C014	Σ+	Accumulate (x, y) in statistical registers
C015	RCL X	
C016	3	
C017	y^x	

Line	Command	Comments
C018	STO+ A	Update Σx^3
C019	RCL X	
C020	x^2	
C021	x^2	
C022	STO+B	Update Σx^4
C023	ISG I	
C024	GTO C012	Resume the next loop iteration
C025	$\bar{x}$	
C026	STO D	Store the first moment
C027	x^2	
C028	+/-	
C029	Σx^2	
C030	n	
C031	÷	
C032	+	Calculate the second moment
C033	STO E	
C034	RCL A	
C035	n	
C036	÷	
C037	RCLD	
C038	3	
C039	×	
C040	n	
C041	÷	
C042	Σx^2	
C043	×	
C044	-	
C045	RCL D	
C046	3	
C047	y^x	
C048	2	
C049	×	
C050	+	Calculate the third moment
C051	STO F	
C052	RCL B	
C053	n	
C054	÷	
C055	4	
C056	n	
C057	÷	
C058	RCL D	
C059	×	
C060	RCL A	

Line	Command	Comments
C061	×	
C062	-	
C063	6	
C064	n	
C065	÷	
C066	RCL D	
C067	x^2	
C068	×	
C069	Σx^2	
C070	×	
C071	+	
C072	RCL D	
C073	x^2	
C074	x^2	
C075	3	
C076	×	
C077	-	Calculate the fourth moment
C078	STO G	
C079	RCL F	
C080	RCL E	
C081	1.5	
C082	y^x	
C083	÷	Calculate the moment coefficient of skewness
C084	STO Y	
C085	RCL G	
C086	RCLE	
C087	x^2	
C088	÷	Calculate the moment coefficient of kurtosis
C089	STO Z	
C090	SF 10	
C091	M1,M2	This is a “tag” created using EQN button
C092	PSE	Display output tag
C093	RCL D	
C094	RCL E	
C095	STOP	Display first and second moments
C096	M3,M4	This is a “tag” created using EQN button
C097	PSE	Display output tag
C098	RCL F	
C099	RCL G	
C100	STOP	Display third and fourth moments
C101	GAM1,GAM2	This is a “tag” created using EQN button
C102	PSE	Display output tag
C103	CF 10	

Line	Command	Comments
C104	RCL Y	
C105	RCL Z	
C106	RTN	Display the coefficients of skewness and kurtosis
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	RCL X	
G004	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	Enter this statement using the EQN key.
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-MOMENTS1	This is a “tag” created using EQN button
U004	PSE	Display output tag
U005	CF 10	
U006	RTN	

Example

Let’s look at an example that exercises the program. We want to calculate the moments for the variable x whose values appear in the next table.

Index	x
1	2.1
2	3.5
3	4.2
4	6.5
5	4.1
6	3.6
7	5.3
8	3.7
9	4.9

After you enter the above data, calculate the moments.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0.00
2	Add the first		XEQ A	X?

Step #	Step/Substep	Input	Command	Output
	data point.			
	Enter a value for x.	2.1	R/S	1.00
3	Add the second data point.		XEQ A	X?
	Enter a value for x.	3.5	R/S	2.00
4	Repeat step 3 for all other data point.			9.00
5	Calculate the moments. The program first displays the first two moments.		XEQ C	 Displays the values for the first and second moments
	View the third and fourth moments.		R/S	 Displays the values for the third and fourth moments
	View the coefficients of skewness and kurtosis		R/S	 Displays the values for the coefficients of skewness and kurtosis

Moments Program NCS_MOMENTS2

Overview

The NCS_MOMENTS2 program calculates statistical moments and offers the following features:

11. Storing data points. Each data point consists of a value x and a frequency f. The program stores the data points using 2-D vectors, thus cutting the memory requirements in half.
12. The ability to delete, edit, view, and swap data points. Thus the program incorporates basic operations to manage the data points.
13. Calculating the first moments and the coefficients of skewness and kurtosis.

This program is a superset version of program NCS_MOMENTS1.While the latter program handles an array x of simple observations, the NCS_MOMENTS2 program allows you to specify a frequency value for each value of x.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

The first moment is:

$$F = \frac{1}{n} \Sigma f_i$$

$$\bar{x} = \frac{1}{F} \Sigma f_i x_i$$

The second moment is:

$$m_2 = \frac{1}{F} \Sigma f_i x_i^2 - \bar{x}^2$$

The third moment is:

$$m_3 = \frac{1}{F} \Sigma f_i x_i^3 - \frac{3}{F} \bar{x} \Sigma f_i x_i^2 + 2\bar{x}^3$$

The fourth moment is:

$$m_4 = \frac{1}{F} \Sigma f_i x_i^4 - \frac{4}{F} \bar{x} \Sigma f_i x_i^3 + \frac{6}{F} \bar{x}^2 \Sigma f_i x_i^2 - 3\bar{x}^4$$

The moment coefficient of skewness is:

$$\gamma_1 = m_3 / m_2^{1.5}$$

The moment coefficient of kurtosis is:

$$\gamma_2 = m_4 / m_2^{2}$$

Instructions

The following table shows the steps involved in using the various parts of the moments program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value for x.	x	R/S	F?		
	Enter a value for the frequency.	f	R/S	Number of data points		
3	Delete a data point.		XEQ D	I?		
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message INVALID DATA.	Index of data point to delete	R/S	Number of data points		
4	Edit a data point.		XEQ E	I?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	Enter the index of the data point to edit. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	X?		
	Enter a new value for x.	x	R/S	F?		
	Enter a value for the frequency.	f	R/S			
5	View a data point.		XEQ V	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	Data point	Index	
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	Data point	Index	
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1		
6	View all data points (Note the program displays the results using the PSE statement.) Note: If you spot an error in the data you entered		XEQ W	Data point	Index	

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.					
7	Swap data points.		XEQ S	I?		
	Enter the index of the first data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the first data point	R/S	J?		
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S			
8	Calculate the moments. The program first displays the first two moments.		XEQ C	Second moment	First moment	
	View the third and fourth moments.		R/S	Fourth moment	Third moment	
	View the coefficients of skewness and kurtosis.		R/S	Coefficient of kurtosis	Coefficient of skewness	
9	To display the program name. The program pauses to display the program name.		XEQ U	NCS-MOMENTS2		

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A	Σfx^3	1	Index invalid when set
D	Delete an existing data point	B	Σfx^4		
E	Edit a data point	C	Number of data points		
V	View a data point	D	First moment $\bar{x}$		
W	View all data points	E	Second moment m_2		
I	Initialize	F	f, Third moment m_3		
S	Swap two data points	G	Fourth moment m_4		
C	Calculate the moments	I	Index		
U	Display program name	J	Index		
		K	Σf		
		L	Σfx		
		M	Σfx^2		
		X	x		
		Y	Moment coefficient of skewness γ_1		
		Z	Moment coefficient of kurtosis γ_2		
		1	Data point #1		
Labels for internal use		2	Data point #2		
H	Validate indices and display error message if an index is invalid	3	Data point #3		
G	Prompt for x	…			
R	Display the message INVALID DATA	#C	Data point #C		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point

Line	Command	Comments
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	STOP	
A011	GTO A001	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO– C	
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	

Line	Command	Comments
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	Validate index
S006	RCL (I)	Recall data points and place them on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x↔y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	FS? 1	Index out or range?
V005	GTO R001	
V006	RCL (I)	Recall data point
V007	STOP	
V008	1	
V009	RCL I	
V010	+	
V011	STO I	
V012	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate regression coefficients
C002	1	
C003	RCL C	

Line	Command	Comments
C004	1E3	
C005	÷	
C006	+	
C007	STO I	Store loop index
C008	0	Clear statistical registers
C009	STO A	Initialize Σx^3
C010	STO B	Initialize Σx^4
C011	STO K	Initialize Σf
C012	STO L	Initialize Σx
C013	STO M	Initialize Σx^2
C014	RCL (I)	Start the loop
C015	[0,1]	
C016	×	Get f
C017	STO F	
C018	STO+ K	Update Σf
C019	RCL (I)	
C020	[1,0]	
C021	×	Get x
C022	STO X	
C023	×	Calculate f × x
C024	STO+ L	Update Σfx
C025	RCLX	
C026	×	
C027	STO+ M	Update Σfx^2
C028	RCL X	
C029	3	
C030	y^x	
C031	RCL F	
C032	×	
C033	STO+ A	Update Σfx^3
C034	RCL X	
C035	x^2	
C036	x^2	
C037	RCL F	
C038	×	
C039	STO+B	Update Σfx^4
C040	ISG I	
C041	GTO C014	Resume the next loop iteration
C042	RCL L	
C043	RCL K	
C044	÷	
C045	STO D	Store the first moment
C046	x^2	

Line	Command	Comments
C047	+/-	
C048	RCL M	
C049	RCL K	
C050	÷	
C051	+	Calculate the second moment
C052	STO E	
C053	RCL A	
C054	RCL K	
C055	÷	
C056	RCLD	
C057	3	
C058	×	
C059	RCL K	
C060	÷	
C061	RCL M	
C062	×	
C063	-	
C064	RCL D	
C065	3	
C066	y^x	
C067	2	
C068	×	
C069	+	Calculate the third moment
C070	STO F	
C071	RCL B	
C072	RCL K	
C073	÷	
C074	4	
C075	RCL K	
C076	÷	
C077	RCL D	
C078	×	
C079	RCL A	
C080	×	
C081	-	
C082	6	
C083	RCL K	
C084	÷	
C085	RCL D	
C086	x^2	
C087	×	
C088	RCL M	
C089	×	

Line	Command	Comments
C090	+	
C091	RCL D	
C092	x^2	
C093	x^2	
C094	3	
C095	×	
C096	-	Calculate the fourth moment
C097	STO G	
C098	RCL F	
C099	RCL E	
C100	1.5	
C101	y^x	
C102	÷	Calculate the moment coefficient of skewness
C103	STO Y	
C104	RCL G	
C105	RCLE	
C106	x^2	
C107	÷	Calculate the moment coefficient of kurtosis
C108	STO Z	
C109	SF 10	
C110	M1,M2	This is a “tag” created using EQN button
C111	PSE	
C112	RCL D	
C113	RCL E	
C114	STOP	Display first and second moments
C115	M3,M4	This is a “tag” created using EQN button
C116	PSE	
C117	RCL F	
C118	RCL G	
C119	STOP	Display third and fourth moments
C120	GAM1,GAM2	This is a “tag” created using EQN button
C121	PSE	Display output tag
C122	CF 10	
C123	RCL Y	
C124	RCL Z	
C125	RTN	Display the coefficients of skewness and kurtosis
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT F	Prompt use for frequency value
G004	[X,F]	
G005	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	Enter this statement using the EQN key.

Line	Command	Comments
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-MOMENTS2	This is a "tag" created using EQN button
U004	PSE	Display output tag
U005	CF 10	
U006	RTN	

Example

Let's look at an example that exercises the program. We want to calculate the moments for the variable x with frequency f, whose values appear in the next table.

Index	x	Frequency f
1	3	4
2	2	5
3	4	3
4	6	2
5	1	1

After you enter the above data, calculate the moments.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0.00
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	3	R/S	F?
	Enter a value for the frequency.	4	R/S	1.00
3	Add the second data point.		XEQ A	X?
	Enter a value for x.	2	R/S	F?
	Enter a value for the frequency.	5	R/S	2.00
4	Repeat step 3			5.00

Step #	Step/Substep	Input	Command	Output
	for all other data point.			
5	Calculate the moments. The program first displays the first two moments.		XEQ C	3.13 1.98 Displays the values for the first and second moments
	View the third and fourth moments.		R/S	2.14 11.05 Displays the values for the third and fourth moments
	View the coefficients of skewness and kurtosis		R/S	0.77 2.81 Displays the values for the coefficients of skewness and kurtosis

Paired Data T-Test Program NCS_PAIRED1

Overview

The NCS_PAIRED1 program implements the Student-t test for the means of paired data. The program is actually two sets of similar programs. One version allows you to store arrays of paired data and the other version does not. The version that allows you to store the array of paired observations offers the following features:

14. Storing the (x, y) data points as 2-D vectors, a new feature of the HP-35s, thus reducing the number of registers used by half.
15. The ability to delete, edit, view, and swap data points. Thus the program incorporates basic operations to manage the data points.
16. Calculating the mean, standard deviation, and Student-t statistic.

The version that processes the paired data without storing them offers you the following features:

1. Entering paired data.
2. Deleting previously entered data. You must enter the old and new pair of observations.
3. Calculating the mean, standard deviation, and Student-t statistic.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Give two arrays x and y of equal number of data elements. The null hypothesis is that the two arrays have the same statistical mean:

$H_0 : \mu_1 = \mu_2$

The difference for each paired observations is:

$$D_i = x_i - y_i$$

The mean and standard deviation for the array of differences is:

$$\overline{D} = \frac{1}{n} \sum_{i=1}^{n} D_i$$

$$s_D = \sqrt{\left(\sum \mathbf{D}^2_i - \frac{\mathbf{1}}{\mathbf{n}} \sum \mathbf{D}_i\right)/(\mathbf{n} - \mathbf{1})}$$

The Student-t statistic is calculated using the following equation:

$$t = \sqrt{n} \cdot \overline{D} / s_D$$

If the calculated t exceeds the value of $t_{\alpha/2,n-1}$ then we cannot accept the null hypothesis at a confidence level of (1-α)%--the paired data have the same statistical mean value,

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y
1	Initialize the program.		XEQ I	0	
2	Add a new data point.		XEQ A	X?	
	Enter a value for x.	x	R/S	Y?	
	Enter a value for y.	y	R/S	Number of data points	
3	Delete a data point.		XEQ D	I?	
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message INVALID DATA.	Index of data point to delete	R/S	Number of data points	
4	Edit a data point.		XEQ E	I?	
	Enter the index of the data point to edit. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	X?	
	Enter a new value for x.	x	R/S	Y?	
	Enter a new value for y.	y	R/S		
5	View a data point.		XEQ V	I?	
	Enter the index of the data point to view. If you enter	Index	R/S	[x,y] data point	Index

Step #	Step/Substep	Input	Command	Output X	Output Y
	an invalid index the program displays the message INVALID DATA.				
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	[x,y] data point	Index
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1	
6	View all data points (Note the program displays the results using the PSE statement.) Note: If you spot an error in the data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	[x,y] data point	Index
7	Swap data points.		XEQ S	I?	
	Enter the index of the first data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the first data point	R/S	J?	
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S		
8	Calculate the results (using arrays) and view the difference mean, and the difference standard deviation.		XEQ C	D mean	SD
	To view the Student-t and the degrees of freedom.		R/S	Student-t	n-1

Step #	Step/Substep	Input	Command	Output X	Output Y
9	To add a data point without storing it in the array		XEQ N	X?	
	Enter a value for x.	x	R/S	Y?	
	Enter a value for y.	y	R/S	Number of data points	
10	To delete a data point that you entered in step 9.		XEQ M	X?	
	Enter a value for x.	x	R/S	Y?	
	Enter a value for y.	y	R/S	Number of data points	
11	Calculate the results (without using arrays) and view the difference standard deviation and difference mean.		XEQ P	D mean	SD
	View the degrees of freedom and the Student-t statistic.		R/S	Student-t	n-1
12	Display the program name. The program pauses to display the program name.		XEQ U	NCS-PAIRED1	

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point to the arrays	C	Count for number of data points	1	Index invalid when set
D	Delete an existing data point from the arrays	X	x	2	Used
E	Edit a data point in the arrays	Y	y		
V	View a data point in the array	I	Index, used		
W	View all data points in the array	1	Data point #1		
I	Initialize	2	Data point #2		
S	Swap two data points in the array	3	Data point #3		
C	Calculate results using arrays	…			
N	Add a data point without storing the data in an array	#C	Data point #C		
M	Delete a data point				
P	Calculate results				
U	Display program name				
Labels for internal use					
H	Validate indices and display error message if an index is invalid				
G	Prompt for x and y values				
R	Display the message INVALID DATA				

Program Listing

Line	Command	Comments
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	ClΣ	
I005	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO- C	

Line	Command	Comments
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	
S006	RCL (I)	Recall data points on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x↔y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	FS? 1	
V005	GTO R001	
V006	RCL (I)	Recall data point
V007	STOP	
V008	1	
V009	RCL I	
V010	+	
V011	STO I	
V012	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points

Line	Command	Comments
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate results using the arrays
C002	ClΣ	
C003	1	
C004	RCL C	
C005	1E3	
C006	÷	
C007	+	
C008	STO I	
C009	RCL (I)	
C010	[0,1]	
C011	×	
C012	RCL (I)	
C013	[1,0]	
C014	×	
C015	-	
C016	Σ+	
C017	ISG I	
C018	GTO C009	
P001	LBL P	Display results
P002	sx	
P003	$\bar{x}$	
P004	SF 10	
P005	SD,D-BAR	This is a “tag” created using the EQN button
P006	PSE	Display output tag
P007	STOP	Display difference sdev and difference mean
P008	n	
P009	1	
P010	-	
P011	n	
P012	√x	
P013	$\bar{x}$	
P014	×	
P015	sx	
P016	÷	
P017	DF,STUDENT-T	This is a “tag” created using the EQN button
P018	PSE	Display output tag
P019	CF 10	

Line	Command	Comments
P020	RTN	Display degrees of freedom and Student-t
M001	LBL M	
M002	SF 2	
M003	GTO N003	
N001	LBL N	
N002	CF 2	
N003	INPUT X	
N004	INPUT Y	
N005	RCL X	
N006	RCL Y	
N007	-	
N008	FS? 2	
N009	Σ-	
N010	FS? 2	
N011	RTN	
N012	Σ+	
N013	STOP	
N014	GTO N001	
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT Y	Prompt user for y value
G004	[X,Y]	Create a data point
G005	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-PAIRED1	This is a “tag” created using the EQN button
U004	PSE	Display program name
U005	CF 10	
U006	RTN	

Example

Let's look at an example that exercises various routines in the NCS_PAIREDT1 program. The next table contains (x, y) data points that we want to test for equal statistical means.

x	y
14.0	17.0
17.5	20.7
17.0	21.6
17.5	20.9
15.4	17.2

After you enter the above data, perform the following steps:

1. Calculate the mean and standard deviation of the differences.
2. Calculate degrees of freedom and Student-t statistic.
3. Test the null hypothesis at a 95% confidence level.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	14.0	R/S	Y?
	Enter a value for y.	17.0	R/S	1
3	Add the second data point.		XEQ A	X?
	Enter a value for x.	17.5	R/S	Y?
	Enter a value for y.	20.7	R/S	2
4	Repeat step 3 for all other data point.			7
5	View all data points (Note the program displays the results using the PSE statement). Note: If you spot an error, execute XEQ E to overwrite the erroneous data point with the correct one.		XEQ W	

Step #	Step/Substep	Input	Command	Output
6	Calculate statistical results. The program first displays the value for the difference standard deviation and difference mean.		XEQ C	1.000 3.200 Displays the values for the difference standard deviation and difference mean
	View the values for the degrees of freedom and the Student-t.		R/S	4.000 7.155 Displays the values for the degrees of freedom and the Student-t.

Since the calculated t value of 7.155 exceeds the value of $t_{0.025,4}$ of 2.776 we cannot accept the null hypothesis stating that the paired samples have the same statistical means.

Appendix A contains a new approximation that I recently developed for inverse Student-t function. The appendix also contains tables for selected values.

Quadratic Regression Program NCS_POLY1

Overview

The NCS_POLY1 program implements quadratic regression between two variables, as shown next:

$z = a + b x + c x^2$

The program offers the following features:

17. Storing data points once and reusing them for different regression scenarios. The program stores the (x, z) data points as 3-D vectors, a new feature of the HP-35s, thus reducing the number of registers used by one third.
18. The ability to delete, edit, view, and swap data points. Thus the regression program incorporates basic operations to manage the data points.
19. Transforming the z observations. You can edit the program to implement your custom transformations and then recalculate the regression statistics without having to reenter the data points.
20. Calculating the regression intercept, slopes, and coefficient of determination.
21. Calculating the confidence intervals for intercept and slopes.
22. Calculating the regression ANOVA table.
23. Calculating projected value of z for existing or new values of x.

Since the program allows you to transform the z values, the general form for the regression model fitted is:

$f(z) = a + b x + c x^2$

Where f(z) is a function that you select and code in to perform the required transformation that supports a linearized regression.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations. The following equations and the rest of this document use the pseudo-variable y to represent x^2.

Data Collected

$\sum x$ = sum of x
$\sum x^2$ = sum of x^2
$\sum y$ = sum of y = sum of x^2
$\sum y^2$ = sum of y^2 = sum of x^4
$\sum z$ = sum of z
$\sum z^2$ = sum of z^2
$\sum xy$ = sum of $x \cdot y$ = sum of x^3
$\sum xz$ = sum of $x \cdot z$
$\sum yz$ = sum of $y \cdot z$ = sum of $x^2 \cdot z$
n = number of observations

Regression Coefficients

$x_m = \sum x / n$
$y_m = \sum y / n$
$z_m = \sum z / n$
$S_{xx} = \sum x^2 - (\sum x)^2 / n = \sum x^2 - n\,(x_m)^2$
$S_{yy} = \sum y^2 - (\sum y)^2 / n = \sum y^2 - n\,(y_m)^2$
$S_{zz} = \sum z^2 - (\sum z)^2 / n = \sum z^2 - n\,(z_m)^2$
$S_{xy} = \sum xy - (\sum x)(\sum y)/ n = \sum xy - n\, x_m\, y_m$
$S_{xz} = \sum xz - (\sum x)(\sum z)/ n = \sum xz - n\, x_m\, z_m$
$S_{yz} = \sum yz - (\sum y)(\sum z)/ n = \sum yz - n\, y_m\, z_m$

$D = S_{xx} S_{yy} - (S_{xy})^2$
Slope $b = [S_{xz} S_{yy} - S_{yz} S_{xy}] / [S_{xx} S_{yy} - (S_{xy})^2]$
Slope $c = [S_{yz} S_{xx} - S_{xz} S_{xy}] / [S_{xx} S_{yy} - (S_{xy})^2]$
Intercept $a = z_m - b\, x_m - c\, y_m$
$R^2 = [a \sum z + b \sum xz + c \sum yz - (\sum z)^2 / n] / [\sum z^2 - (\sum z)^2 / n]$
$= [a \sum z + b \sum xz + c \sum yz - (\sum z)^2 / n] / S_{zz}$

ANOVA Table

Source of Variation	***Sum of Squares***	***Degrees of Freedom***	***Mean Square***	***F***
Regression	$SS_R = b\,S_{xz} + c\,S_{yz}$	2	MS_R	MS_R / MS_E
Residual	$SS_E = S_{zz} - b\,S_{xz} - c\,S_{yz}$	n – 3	MS_E	
Total	$SS_T = S_{zz}$	n – 1		

Interval for Slope b

At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$
$SlopeStdErr = \sqrt{[MS_E (S_{yy} / D)]}$
$\Delta b = t_{\alpha/2,n-3} \cdot SlopeStdErr$

Upper Range: $b + \Delta b$
Lower Range: $b - \Delta b$

Interval for Slope c

At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$
$SlopeStdErr = \sqrt{[MS_E (S_{xx} / D)]}$
$\Delta c = t_{\alpha/2,n-3} \cdot SlopeStdErr$

Upper Range: $c + \Delta c$
Lower Range: $c - \Delta c$

Interval for Intercept a

At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$
$IntStdErr = \sqrt{(MS_E [1/n + (S_{yy} / D)(x_m)^2 + (S_{xx} / D)(y_m)^2 - 2(S_{xy} / D) x_m y_m])}$
$= \sqrt{(MS_E [1/n + \{S_{yy} (x_m)^2 + S_{xx} (y_m)^2 - 2(S_{xy} x_m y_m\}] / D)}$

$\Delta a = t_{\alpha/2,n-3} \cdot IntStdErr$

Upper Range: $a + \Delta a$
Lower Range: $a - \Delta a$

Interval for Projected z

At 100(1 – α) confidence:
$D = S_{xx} S_{yy} - (S_{xy})^2$

$\hat{z} = a + b x_0 + c y_0$

$zhatStdErr = \sqrt{(MS_E [1/n + (S_{yy} / D)(x_0 - x_m)^2 + (S_{xx} / D)(y_0 - y_m)^2 - 2 (S_{xy} / D) (x_0 - x_m) (y_0 - y_m)])}$
$= \sqrt{(MS_E [1/n + \{S_{yy} (x_0 - x_m)^2 + S_{xx} (y_0 - y_m)^2 - 2 S_{xy} (x_0 - x_m) (y_0 - y_m)\}/D])}$

$\Delta\hat{z} = t_{\alpha/2,n-3} \cdot zhatStdErr$
Upper Range: $\hat{z} + \Delta\hat{z}$
Lower Range: $\hat{z} - \Delta\hat{z}$

For a *new* observation:

$zhatStdErr = t_{\alpha/2,n-3} \cdot \sqrt{((MS_E)^2 [1+1/n + (S_{yy} / D)(x_0 – x_m)^2 + (S_{xx} / D)(y_0 – y_m)^2 – 2 (S_{xy} / D) (x_0 – x_m) (y_0 – y_m)])}$
$= \sqrt{((MS_E)^2 [1 + 1/n + \{S_{yy} (x_0 – x_m)^2 + S_{xx} (y_0 – y_m)^2 – 2 S_{xy} (x_0 – x_m) (y_0 – y_m)\}/D])}$

$\Delta\hat{z} = t_{\alpha/2,n-3} \cdot zhatStdErr$
Upper Range: $\hat{z}^{\wedge} + \Delta\hat{z}$
Lower Range: $\hat{z}^{\wedge} – \Delta\hat{z}$

Instructions

The following table shows the steps involved in using the various parts of the quadratic regression program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value for x.	x	R/S	Z?		
	Enter a value for z.	z	R/S	Number of data points		
3	Delete a data point.		XEQ D	I?		
	Enter the index of the data point to delete. If you enter an invalid index the program displays the message INVALID DATA.	Index of data point to delete	R/S	Number of data points		
4	Edit a data point.		XEQ E	X?		
	Enter a new value	x	R/S	Z?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	for x.					
	Enter a value for z.	z	R/S	I?		
	Enter the index of the data point to edit. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S			
5	View a data point.		XEQ V	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S	Data point	Index	
	To view the next data point. You can repeat this substep to view a succession of data points.		R/S	Data point	Index	
	When you have finished viewing the last data point and press R/S the calculator displays the message INVALID DATA.		R/S	Number of data points + 1		
6	View all data points (Note the program displays the results using the PSE statement.)		XEQ W	Data point	Index	
7	Swap data points.		XEQ S	I?		
	Enter the index of the first data point. If you enter an invalid index the program displays	Index of the first data point	R/S	J?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	the message INVALID DATA.					
	Enter the index of the second data point. If you enter an invalid index the program displays the message INVALID DATA.	Index of the second data point	R/S			
8	Calculate the regression coefficients (note that the value of r^2 is in the T stack register). Use the R↓ key to scroll down the stack.		XEQ C	Slope c	Slope b	Intercept
9	Calculate the regression ANOVA table. The program first displays the values for SSR and regression degrees of freedom.		XEQ N	Regression DF	Regression sum of squares (SSR)	
	View the value of MSR.		R/S	Mean Square for regression (MSR)		
	View the value of the calculated F statistic.		R/S	Calculated F statistic		
	View the values of SSE and the degrees of freedom for the residual.		R/S	Residual df	Residual sum of squares (SSE)	
	View the value of MSE.		R/S	Mean Square for residuals (MSE)		
	View the values of total sum of		R/S	Total df	Total sum of squares	

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	squares and the total degrees of freedom.					
10	Calculate the confidence intervals.		XEQ F	T?		
	Enter a value for the inverse Student-t probability distribution function. The program first displays the confidence interval for the intercept.	Student-t for n-3 degrees of freedom and α/2 significance level (α is based on 1-α confidence level.)	R/S	Intercept lower limit	Interval upper limit	
	View the confidence interval for the slop b.		R/S	Slope b lower limit	Slope b upper limit	
	View the confidence interval for the slop c.		R/S	Slope c lower limit	Slope c upper limit	
11	Project an existing x onto z.		XEQ P	T?		
	Enter a value for the inverse Student-t probability distribution function.	Student-t for n-3 degrees of freedom and α/2 significance level (α is based on 1-α confidence level.)	R/S	X?		
		x	R/S	$\hat{z}$ lower limit	$\hat{z}$	$\hat{z}$ upper limit
12	Project a new x onto z.		XEQ Q	T?		
	Enter a value for the inverse Student-t probability	Student-t for n-3 degrees of freedom and α/2	R/S	X?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	distribution function.	significance level (α is based on 1-α confidence level.)				
		x	R/S	$\hat{z}$ lower limit	$\hat{z}$	$\hat{z}$ upper limit
13	You can temporarily (or even permanently) reduce the number of data points entering in the regression calculations.		RCL C	Number of data points		
	Record the number of original data points.					
	Enter a lower value for the number of data points (must be greater than 4.)	New number of data points	STO C			
14	You can restore the original number of data points entering in the regression calculations.	Original number of data points	STO C			
15	Alter the transformation for z.		GTO Z001			
	Switch to program mode.		PRGM	Z001 LBL Z		
	Edit the commands between LBL Z and the RTN statement to reflect the new transformation you desire.					
	Exit program mode when you		PRGM			

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	are done.					
16	Display the program name. The program pauses to display the program name.		XEQ U	NCS-POLY1		

Comments on Runtime Errors

The program validates the indices that you enter to manage data points to prevent the corruption of the data points. Aside from this feature, you are responsible to enter data that will avoid runtime errors. For example, if you are using the square root or logarithms to transform the x and/or y data, then entering data points with negative values will raise runtime errors. Likewise, using reciprocals and logarithms will generate runtime errors for data that are zeros. Entering redundant data points also has its peril of giving you meaningless results.

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	A	Used, r^2	1	Index invalid when set
D	Delete an existing data point	B	Used	2	Project a new x onto z when set
E	Edit a data point	C	Count for number of data points	3	Inverse transform z
V	View a data point	D	Σz		
W	View all data points	E	Σz^2		
I	Initialize	F	Σxz		
S	Swap two data points	G	Σyz		
C	Calculate regression coefficients	H	MSE		
N	Calculate regression ANOVA	I	Index, used		
F	Calculate confidence intervals	J	Index, used		
P	Project an observed x and y onto z with	K	Intercept a		

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
	confidence interval				
Q	Project a new x and y onto z with confidence interval	L	Slope b		
Z	Transform and Inverse transform of z data	M	Slope c		
U	Display program name	N	Sxx		
Labels for internal use			O	Syy	
H	Validate indices and display error message if the an index is invalid	P	Sxy		
G	Prompt for x and z	Q	Sxz		
R	Display the message INVALID DATA	R	Sxy		
		S	Szz		
		T	Inverse Student-t pdf		
		U	$D = S_{xx} S_{yy} – (S_{xy})^2$		
		V	Upper limit		
		W	Lower limit		
		X	x		
		Y	$y=x^2$		
		Z	z		
		1	Data point #1		
		2	Data point #2		
		3	Data point #3		
		…			
		#C	Data point #C		

Program Listing

Line	Command	Comments
Z001	LBL Z	Label to contain the transformation and inverse transformation for z
Z002	RTN	
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point

Line	Command	Comments
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO- C	
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	

Line	Command	Comments
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	
S006	RCL (I)	Recall data points on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x↔y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	RCL (I)	Recall data point
V005	STOP	
V006	1	
V007	RCL I	
V008	+	
V009	STO I	
V010	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate regression coefficients
C002	ClΣ	Clear statistical registers
C003	0	Initialize the additional statistical registers
C004	STO D	
C005	STO E	

Line	Command	Comments
C006	STO F	
C007	STO G	
C008	1	
C009	RCL C	
C010	1E3	
C011	÷	
C012	+	
C013	STO I	Store loop index
C014	RCL (I)	Start the loop
C015	[1,0]	
C016	×	Get x
C017	XEQ X	Transform value of x
C018	STO X	Store transformed x
C019	x^2	
C020	STO Y	Store y
C021	RCL (I)	
C022	[0,1]	
C023	×	Get z
C024	XEQ Z	Transform value of z
C025	STO Z	
C026	RCL Y	
C027	RCL X	
C028	Σ+	Accumulate (x, y) in statistical registers
C029	RCL Z	
C030	STO+ D	Add to sum of z
C031	x^2	
C032	STO+ E	Add to sum of z^2
C033	RCL X	
C034	RCL Z	
C035	×	
C036	STO+ F	Add to sum of x times z
C037	RCL Y	
C038	RCL Z	
C039	×	
C040	STO+ G	Add to sum of y times z
C041	ISG I	
C042	GTO C014	Resume the next loop iteration
C043	Σx^2	Calculate Sxx
C044	Σx	
C045	x^2	
C046	n	
C047	÷	
C048	-	

Line	Command	Comments
C049	STO N	Store Sxx
C050	Σy^2	Calculate Syy
C051	Σy	
C052	x^2	
C053	n	
C054	÷	
C055	-	
C056	STO O	Store Syy
C057	Σxy	Calculate Sxy
C058	Σx	
C059	Σy	
C060	×	
C061	n	
C062	÷	
C063	-	
C064	STO P	Store Sxy
C065	RCL F	
C066	Σx	
C067	RCL D	
C068	×	
C069	n	
C070	÷	
C071	-	
C072	STO Q	Store Sxz
C073	RCL G	
C074	Σy	
C075	RCL D	
C076	×	
C077	n	
C078	÷	
C079	-	
C080	STO R	Store Syz
C081	RCL N	
C082	RCL O	
C083	×	
C084	RCL P	
C085	x^2	
C086	-	
C087	STO U	Store D
C088	RCL Q	
C089	RCL O	
C090	×	
C091	RCL R	

Line	Command	Comments
C092	RCL P	
C093	×	
C094	-	
C095	RCL U	
C096	÷	
C097	STO L	Store slope b
C098	RCL R	
C099	RCL N	
C100	×	
C101	RCL Q	
C102	RCL P	
C103	×	
C104	-	
C105	RCL U	
C106	÷	
C107	STO M	Store slope c
C108	RCL D	
C109	n	
C110	÷	
C111	RCL L	
C112	$\bar{x}$	
C113	×	
C114	-	
C115	RCL M	
C116	$\bar{y}$	
C117	×	
C118	-	
C119	STO K	Store intercept
C120	RCL D	Calculate r^2
C121	×	
C122	RCL L	
C123	RCL F	
C124	×	
C125	+	
C126	RCL M	
C127	RCL G	
C128	×	
C129	+	
C130	RCL D	
C131	x^2	
C132	n	
C133	÷	
C134	STO B	

Line	Command	Comments
C135	-	
C136	RCL E	
C137	RCL B	
C138	-	
C139	STO S	Store Szz
C140	÷	
C141	STO A	Store r^2
C142	RCL S	
C143	RCL L	
C144	RCL Q	
C145	×	
C146	-	
C147	RCL M	
C148	RCL R	
C149	×	
C150	-	
C151	n	
C152	3	
C153	-	
C154	÷	
C155	STO H	Store MSE
C156	SF 10	
C157	R2,A,B,C	This is a “tag” created using the EQN button
C158	PSE	Display output tag for $r^2$and the regression coefficients
C159	CF 10	
C160	RCL A	Push r^2 into the stack
C161	RCL K	Push the intercept into the stack
C162	RCL L	Push the slope b into the stack
C163	RCL M	Push the slope c into the stack
C164	RTN	Display values of $r^2$and the regression coefficients
N001	LBL N	Display components of ANOVA table
N002	RCL L	
N003	RCL Q	
N004	×	
N005	RCL M	
N006	RCL R	
N007	×	
N008	+	
N009	2	
N010	SF 10	
N011	SSR, DF	This is a “tag” created using the EQN button
N012	PSE	Display output tags
N013	STOP	Display regression df and SSR
N014	÷	Calculate mean square for regression

Line	Command	Comments
N015	MSR	This is a “tag” created using the EQN button
N016	PSE	Display output tag
N017	STOP	Display value of MSR
N018	RCL H	
N019	÷	Calculate F statistic
N020	F STAT	This is a “tag” created using the EQN button
N021	PSE	Display output tag
N022	STOP	Display value of F statistic
N023	RCL H	
N024	n	
N025	3	
N026	-	
N027	×	Calculate residual sum of squares (SSE)
N028	LASTx	Calculate degrees of freedom for SSE
N029	SSE, DF	This is a “tag” created using the EQN button
N030	PSE	Display output tag
N031	STOP	Display SSE and degrees of freedom
N032	÷	Calculate mean square for residual (MSE)
N033	MSE	This is a “tag” created using the EQN button
N034	PSE	Display output tag
N035	STOP	Display value of MSE
N036	RCL S	Get total sum of squares
N037	n	
N038	1	
N039	-	Calculate df for total sum of squares
N040	TSS, DF	This is a “tag” created using the EQN button
N041	PSE	Display output tag
N042	CF 10	
N043	RTN	Display TSS and its degrees of freedom
F001	LBL F	Confidence intervals for intercept and slopes
F002	INPUT T	Get inverse Student-t
F003	RCL O	
F004	$\bar{x}$	
F005	STO B	
F006	x^2	
F007	×	
F008	RCL N	
F009	$\bar{y}$	
F010	STO× B	
F011	x^2	
F012	×	
F013	+	
F014	RCL P	

Line	Command	Comments
F015	2	
F016	×	
F017	RCL B	
F018	×	
F019	-	
F020	RCL U	
F021	÷	
F022	n	
F023	1/x	
F024	+	
F025	RCL H	
F026	×	
F027	√x	
F028	RCL T	
F029	×	
F030	STO B	Store delta interval for intercept
F031	RCL K	
F032	+	Calculate upper limit for intercept
F033	RCL K	
F034	RCL B	
F035	-	Calculate lower limit for intercept
F036	SF 10	
F037	INT UL, LL	This is a “tag” created using the EQN button
F038	PSE	Display output tag
F039	STOP	Display upper and lower limits of the intercept
F040	RCL O	
F041	RCL U	
F042	÷	
F043	RCL H	
F044	×	
F045	√x	
F046	RCL T	
F047	×	
F048	STO B	Store delta interval for slope b
F049	RCL L	
F050	+	Calculate upper limit for slope b
F051	RCL L	
F052	RCL B	
F053	-	Calculate lower limit for slope b
F054	SLP B UL,LL	This is a “tag” created using the EQN button
F055	PSE	Display output tag
F056	STOP	Display upper and lower limits for slope b
F057	RCL N	

Line	Command	Comments
F058	RCL U	
F059	÷	
F060	RCL H	
F061	×	
F062	√x	
F063	RCL T	
F064	×	
F065	STO B	Store delta interval for slope c
F066	RCL M	
F067	+	Calculate upper limit for slope c
F068	RCL M	
F069	RCL B	
F070	-	Calculate lower limit for slope c
F071	SLP C UL,LL	This is a “tag” created using the EQN button
F072	PSE	Display output tag
F073	CF 10	
F074	RTN	Display upper and lower limits for slope c
Q001	LBL Q	Project a new x onto y
Q002	SF 2	
Q003	GTO P003	
P001	LBL P	Project an existing x onto y
P002	CF 2	
P003	INPUT T	Get inverse Student-t
P004	INPUT X	Get value for x
P005	STO X	
P006	x^2	Get value for y
P007	STO Y	
P008	RCL X	
P009	$\bar{x}$	
P010	-	
P011	STO B	
P012	x^2	
P013	RCL O	
P014	×	
P015	RCL Y	
P016	$\bar{y}$	
P017	-	
P018	STO× B	
P019	x^2	
P020	RCL N	
P021	×	
P022	+	
P023	RCL B	

Line	Command	Comments
P024	RCL P	
P025	×	
P026	2	
P027	×	
P028	-	
P029	RCL U	
P030	÷	
P031	n	
P032	1/x	
P033	FS? 2	
P034	1	
P035	FS? 2	
P036	+	
P037	CF 2	
P038	RCL H	
P039	×	
P040	√x	
P041	RCL T	
P042	×	
P043	STO B	
P044	RCL K	
P045	RCL L	
P046	RCL X	
P047	×	
P048	+	
P049	RCL M	
P050	RCL Y	
P051	×	
P052	+	
P053	STO Z	
P054	RCL B	
P055	+	
P056	SF 3	
P057	XEQ Z	
P058	STO V	
P059	RCL Z	
P060	RCL B	
P061	-	
P062	XEQ Z	
P063	STO W	
P064	RCL Z	
P065	XEQ Z	
P066	STO J	

Line	Command	Comments
P067	CF 3	
P068	RCL V	Push upper limit of $\hat{z}$ into the stack
P069	RCL J	Push value of $\hat{z}$ into the stack
P070	RCL W	Push lower limit of $\hat{z}$ into the stack
P071	SF 10	
P072	Z^U,Z^,Z^L	This is a "tag" created using the EQN button
P073	PSE	Display output tag
P074	CF 10	
P075	RTN	Display confidence interval for projected value
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT Z	Prompt user for z value
G004	RCL X	
G005	[X,Z]	Create data point
G006	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-POLY1	This is a "tag" created using the EQN button
U004	PSE	Display program name
U005	CF 10	
U006	RTN	

Example

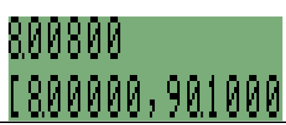
Let's look at an example that exercises various routines in the regression program. The next table contains (x, z) data points that we want to fit in a quadratic regression model.

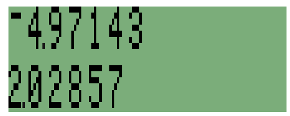
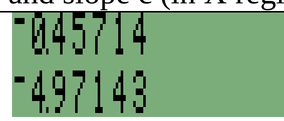
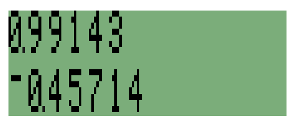
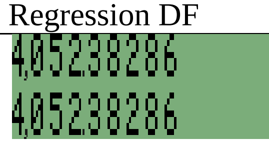
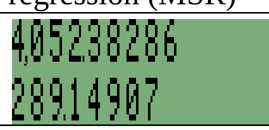
x	z
1	-7.1
2	4.8
3	1.0
4	10.9
5	24.5
6	43.0
7	64.0
8	90.1

After you enter the above data, perform the following steps:

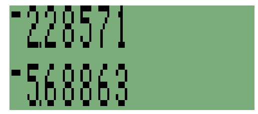
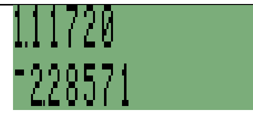
1. View all of the data points.
2. Calculate the regression slope, intercept, and r^2.
3. Calculate the regression ANOVA table.
4. Calculate confidence interval for the intercept and slopes at 95% confidence.
5. Calculate the projected value of z and its confidence interval for x = 2 at a 95% confidence level.

The following table shows the steps performed in the FIX 5 display mode.

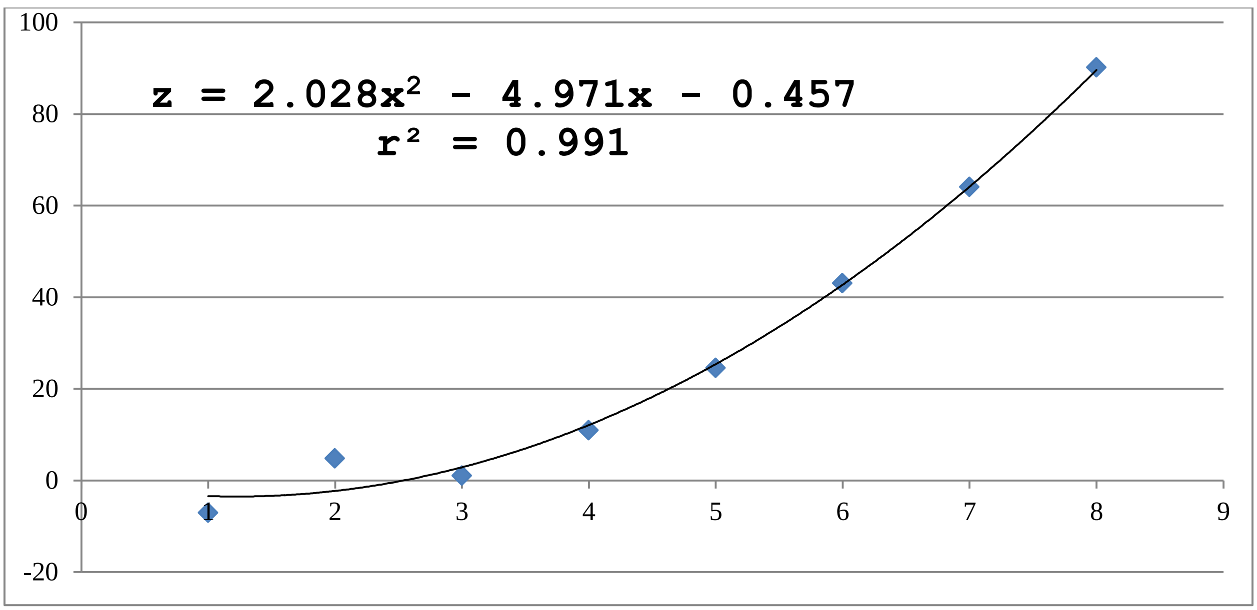
Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Add the first data point.		XEQ A	X?
	Enter a value for x.	1	R/S	Z?
	Enter a value for y.	-7.1	R/S	1.00000
3	Add the second data point		XEQ A	X?
	Enter a value for x	2	R/S	Z?
	Enter a value for y	4.8	R/S	2.00000
4	Repeat step 3 for all other data point			8.00000
5	View all data points (Note the program displays the results using the PSE statement).		XEQ W	

Step #	Step/Substep	Input	Command	Output
	Note: If you spot an error in the data you entered earlier, use the command XEQ E to overwrite the erroneous data point with the correct one.			
6	Calculate the regression coefficients. The program displays the values for slope b and slope c.		XEQ C	 Display slope b (in Y register) and slope c (in X register)
	View the value for the intercept.		R↓	 Displays the value for the intercept and slope b.
	View the value for r^2. The fitted line is: z = – 0.045714 – 4..97143 + 2.02857 x^2 with r^2 of 0.99143		R↓	 Display r^2 and the intercept
7	Calculate the regression ANOVA table. The program first displays the values for SSR and regression degrees of freedom.		XEQ N	 Shows the Regression sum of squares (SSR) and the Regression DF
	View the value of MSR.		R/S	 Shows the Mean Square for regression (MSR)
	View the value of the calculated F statistic.		R/S	

Step #	Step/Substep	Input	Command	Output
				Shows the calculated F statistic
	View the values of SSE and the degrees of freedom for the residual.		R/S	70.07429 5.00000 Shows the Residual sum of squares (SSE) and the Residual df
	View the value of MSE.		R/S	289.14907 14.01486 Shows the Mean Square for residuals (MSE)
	View the values of total sum of squares and the total degrees of freedom.		R/S	8,174.84000 7.00000 Shows the Total Sum of Squares and the Total df
8	Calculate confidence intervals.		XEQ F	T?
	Enter the value for the inverse Student-t for df of 5 and 0.025 significance. The program first displays the confidence interval for the intercept.	2.571	R/S	12.97099 -13.88528 Shows the Intercept confidence interval
	View the confidence interval for the slop b.		R/S	1.87480 -11.81765 Shows the confidence interval for the slope b
	View the confidence interval for the slop c.		R/S	2.77115 1.28599 Shows the confidence interval for the slope c
9	Project an existing x onto z		XEQ P	T?
	Enter the value for the inverse Student-t for df of	2.571	R/S	X?

Step #	Step/Substep	Input	Command	Output
	5 and 0.025 significance.			
	Enter value for x. The program displays the projected value of z and the lower limit for the projected value.	2	R/S	 Shows $\hat{z}$ and $\hat{z}$ lower limit
	View the upper limit for the projected value.		R↓	 Shows $\hat{z}$ upper limit and $\hat{z}$

Here is a plot for the original data and the least-square quadratic fit:



If you desire to apply data transformations to the z observations you need to edit the label Z. For example, if you wish to use a power fit ln(z) = a + b x + c x^2, then you need to have the label Z as follows:

```
Z001 LBL Z
Z002 FS? 3
Z003 GTO Z006
Z004 LN          # transform z
Z005 RTN
Z006 EXP         # inverse transform z
Z007 RTN
```


Notice the use of flag 3 to manage the inverse transformation needed in calculating projected values of z.

Appendix A contains a new approximation that I recently developed for inverse Student-t function. The appendix also contains tables for selected values.

Data Ranking Program NCS_RANKER1

Overview

The NCS_RANKER1 program calculates the numeric ranks for an array of values. You can use the numeric ranks in other programs, such as NCS-SPEARANK1 that calculates the Spearman rank correlation.

The program allows you to enter the values of an array of numbers and then produces the ranks of these numbers. The highest number in the array has the rank of 1. Conversely, the lowest number in the array has the highest rank value, which equals the number of *distinct* values in the array.

The program supports the following operations:

1. Adding values to the array that is being ranked.
2. Deleting values from the array.
3. Viewing and editing specific array elements.
4. Viewing the elements of the array.
5. Ranking the elements of the array.

The process of ranking the elements of the array (call it x) includes the following set of steps:

1. Duplicating the elements of array x to create the array y.
2. Sorting the elements of array x, while keeping the array y unsorted. The program uses the simple Bubble sort method to order the elements of array x in a numerical descending order.
3. Packing the elements of the sorted array x to remove elements with duplicate values. The presence of duplicates would introduce errors in the numeric ranks, and therefore must be removed.
4. Ranking the values of the original array, whose values are stored in array y. The program takes each element in array y and finds the index of the matching value in the sorted array x. The value of this index is the rank of the corresponding element in array y.
5. Restoring the sorted array x to its original values. This step “un-sorts” the array x by copying the values of array y back into x. Why is this step needed? The answer lies in the fact that sorting and packing the elements of array x alters the values of the original array x and prevents the program from correctly performing the operations outlined earlier *after* the values are ranked. Thus, the program restores the array x so you *can* add to, edit, delete from, and view the original data *after* you generate a set of numeric ranks.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

✍ When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X
1	Initialize the program.		XEQ I	0
2	Add a data points.		XEQ A	X?
	Enter a value.	x	R/S	x
	To enter the next value.		R/S	X?
	Repeat the above steps to enter the rest of the array elements.			
3	View/Edit a data point.		XEQ E	I?
	Enter the index for the data point to view/edit.	Index	R/S	X?
	View the value and optionally enter a replacement value for x.	x	R/S	x
4	View the data while pausing at each value		XEQ V	x
5	View the data while stopping at each value.		XEQ W	x(1)
	To view the next value in the array x.		R/S	x(2)
	Repeat this step for the rest of the elements in array x.		R/S	x(n)
6	Delete a data point.		XEQ D	I?
	Enter the index for the data point to delete.	Index	R/S	
7	Obtain the ranks for the array of values you have entered so far. The program pauses to display the messages COPY, SORT, PACK, and RANK before each step that leads to generating the numeric ranks. The program stops to display the		XEQ C	x(1) Rank of x(1)

Step #	Step/Substep	Input	Command	Output X
	rank of the first array element in the X register and the value of the first array element in the Y register.			
	To view the rank of the second array value.		R/S	x(2) Rank of x(2)
	Repeat the above step to obtain the remaining ranks. When you are done, the program pauses to display RESTORE before it restores the original values of the array you entered. When the program is done restoring the original array it pauses to display the message DONE.		R/S	
8	Display the program name. The program pauses to display the program name.		XEQ U	NCS-RANKER1

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a data point	A	n+1	1	Set = Pause, Clear = Stop
C	Rank the data	B	2n		
E	View/Edit an observation	C	Data count, n		
D	Delete a data point	D	Number of elements in packed array = number of unique elements in array ≤ n		
W	View all data points in array x	E			
V	View all data points in array x without stopping the program	F			
I	Initialize the program	G			
U	Display program name	H			
		I	Index		
		J	Index		
L	Prepare loop index	K	Copy of I		
P	Shift array to pack it	L	Copy of J		
		M			
		N			
		O			
		P			

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		Q			
		R			
		S			
		T			
		U			
		V			
		W			
		X	Used		
		Y			
		Z			
		1	x(1)		
		2	x(2)		
		…			
		I	x(I)		
		…			
		n	x(n)		
		n+1	x(1) copy		
		n+2	x(2) copy		
		…			
		n+I	x(I) copy		
		…			
		2n	x(n) copy		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	0	
I003	STO C	
I004	RTN	
L001	LBL L	Help in preparing a loop control value
L002	1E3	
L003	÷	
L004	+	
L005	RTN	
A001	LBL A	Add a data point
A002	INPUT X	
A003	1	
A004	STO+ C	
A005	RCL C	
A006	STO I	
A007	RCL X	
A008	STO (I)	

Line	Command	Comments
A009	STOP	
A010	GTO A001	
D001	LBL D	Delete a data point
D002	INPUT I	
D003	RCL I	
D004	RCL C	
D005	x<y?	
D006	ACOS(2)	Generate INVALID DATA message
D007	x=y?	Delete the last element of array x?
D008	GTO D021	Skip the copying process
D009	RCL I	Copy higher index elements
D010	1	
D011	+	
D012	RCL C	
D013	XEQ L	
D014	STO J	Set index of "copied from" elements J = 1+I + N/1000
D015	RCL (J)	
D016	STO (I)	
D017	1	
D018	STO+ I	
D019	ISG J	
D020	GTO D015	
D021	1	Decrement the number of array elements
D022	STO- C	
D023	RTN	
E001	LBL E	Edit/View a single data point
E002	INPUT I	
E003	RCL I	
E004	RCL C	
E005	x<y?	
E006	ACOS(2)	
E007	RCL (I)	Recall x(I)
E008	STO X	
E009	INPUT X	Display value of x(I) in prompt
E010	STO (I)	Update value
E011	RTN	
V001	LBL V	View all data points in array x without stopping the program
V002	SF 1	
V003	XEQ W003	
V004	CF 1	
V005	RTN	
W001	LBL W	View all data points
W002	CF 1	
W003	1	

Line	Command	Comments
W004	RCL C	
W005	XEQ L	
W006	STO I	
W007	RCL I	Start loop to view data
W008	RCL (I)	
W009	FS? 1	
W010	PSE	
W011	FS? 1	
W012	GTO W014	
W013	STOP	
W014	ISG I	
W015	GTO W007	End loop
W016	RTN	
U001	LBL U	
U002	SF 10	
U003	NCS-RANKER1	This is an "output tag" created using the EQN button
U004	PSE	Display the output tag
U005	CF 10	
U006	RTN	
C001	LBL C	Rank the data
C002	SF 10	
C003	COPY	This is an "output tag" created using the EQN button
C004	PSE	Display the output tag
C005	1	
C006	RCL C	
C007	+	
C008	STO A	Store 1+n in A
C009	STO J	Set index of target registers
C010	RCL C	
C011	2	
C012	×	
C013	STO B	Store 1+2n in B
C014	1	
C015	RCL C	
C016	XEQ L	
C017	STO I	Set index of source registers as 1 + n/1000
C018	RCL (I)	Start copy loop
C019	STO (J)	
C020	1	
C021	STO+ J	
C022	ISG I	
C023	GTO C018	End copy loop
C024	SORT	This is an "output tag" created using the EQN button

Line	Command	Comments
C025	PSE	Display the output tag
C026	1	Sort the array in unnamed registers 1 to N in a descending order
C027	RCL C	
C028	1	
C029	-	
C030	XEQ L	
C031	STO I	Set index for outer loop I = 1 to N-1
C032	RCL I	
C033	IP	
C034	1	
C035	+	
C036	RCL C	
C037	XEQ L	
C038	STO J	Set index for inner loop J = I+1 to N
C039	RCL (I)	
C040	RCL (J)	
C041	x<y?	Reverse this test to sort in ascending order. See comment after listing.
C042	GTO C046	
C043	STO (I)	Swap x(I) and x(J)
C044	x↔y	
C045	STO (J)	
C046	ISG J	
C047	GTO C039	
C048	ISG I	
C049	GTO C032	
C050	PACK	This is an "output tag" created using the EQN button
C051	PSE	Display the output tag
C052	RCL C	
C053	STO D	Store copy of C into D and use D to track the number of elements in packed array
C054	STO I	Store index of first elements I = N
C055	1	
C056	-	
C057	STO J	Store index of second elements J = N-1
C058	RCL (I)	Start loop to remove duplicates
C059	RCL (J)	
C060	x=y?	
C061	XEQ P	Remove duplicate
C062	1	
C063	STO- J	
C064	DSE I	
C065	GTO C058	End of loop to remove duplicate
C066	RANK	This is an "output tag" created using the EQN button
C067	PSE	Display the output tag

Line	Command	Comments
C068	RCLA	
C069	RCL B	
C070	XEQ L	
C071	STO J	Store index to access elements in the unsorted array
C072	1	Start outer loop for search
C073	RCL D	D holds the number of elements in the packed array
C074	XEQ L	
C075	STO I	Store index of sorted array
C076	RCL (J)	Start inner loop to search for index of matching element in sorted array
C077	RCL (I)	
C078	x≠y?	No match found?
C079	GTO C084	
C080	RCL I	Match found
C081	IP	
C082	STOP	Display value and rank
C083	GTO C086	Skip to find the rank of the next value
C084	ISG I	
C085	GTO C076	End of inner search loop
C086	ISG J	
C087	GTO C072	End of outer search loop
C088	RESTORE	This is an "output tag" created using the EQN button
C089	PSE	Display the output tag
C090	RCL A	
C091	RCL B	
C092	XEQ L	
C093	STO J	Store loop control value in J
C094	1	
C095	RCL C	
C096	XEQ L	
C097	STO I	Stop loop control value in I
C098	RCL (J)	Start loop to restore unsorted array
C099	STO (I)	Copy element y(J) to x(I)
C100	1	
C101	STO+ I	
C102	ISG J	
C103	GTO C098	End loop to restore unsorted array
C104	DONE	This is an "output tag" created using the EQN button
C105	PSE	Display the output tag
C106	CF 10	
C107	RTN	
P001	LBL P	Shift array to pack it
P002	RCL I	
P003	RCL D	
P004	x=y?	

Line	Command	Comments
P005	GTO P024	Skip the step of copying over duplicate values
P006	RCL I	
P007	STO K	Copy I into K
P008	RCL J	
P009	STO L	Copy J into L
P010	RCL I	
P011	RCLD	
P012	XEQ L	
P013	STO I	I = I + D / 1000
P014	RCL (I)	Start the loop to overwrite array elements
P015	STO (J)	
P016	1	
P017	STO+J	
P018	ISG I	
P019	GTO P014	End of loop to overwrite array elements
P020	RCL K	Restore values of indices I and J
P021	STO I	
P022	RCL L	
P023	STO J	
P024	1	Decrement counter for sorted array
P025	STO- D	
P026	RTN	

Line C041 contains a conditional test used in the sort phase of the program. The test in line C041 causes the sort process to order the array x in descending order. Consequently, the array elements with higher values generate lower rank values, and vice versa. To reverse the sort order and the order of the rank values, replace line C041 with the following:

C041 x>y?

The proposed change generates high rank values for array elements with high value, and vice versa.

Example

Let's look at an example that exercises various routines in the NCS_RANKER1 program. The next table contains an array of values that need to be ranked.

Index	Array Value
1	82
2	67
3	91
4	98
5	74

Index	Array Value
6	52
7	86
8	95
9	79
10	78
11	84
12	80
13	69
14	81
15	73

After you enter the above data, view the array values, and then obtain the ranks for these values.

The following table shows the steps using the FIX 0 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Enter the data.		XEQ A	X?
	Enter the first value of x.	82	R/S	1. 82.
	Enter the next value of x.		R/S	X?
		67	R/S	2. 67.
	Repeat the above steps to enter the remaining values in the table.			
3	View the data elements. The program pauses to display each array elements.		XEQ V	16. 73.
4	Obtain the ranks for the array. The program pauses to display COPY, SORT, PACK, and RANK before each step that leads to generating the numeric ranks. The program stops to display the rank of the first array element in the X register and the value of the first array element in the Y register.		XEQ C	82. 6.
	Obtain the rank for the		R/S	67.

Step #	Step/Substep	Input	Command	Output
	second array element. The program displays the rank of the next array element in the X register and the value of the array element in the Y register.			14.
	Repeat the above step to obtain the remaining ranks. When you are done, the program pauses to display RESTORE before it restores the original values of the array you entered. When the program is done restoring the original array it pauses to display the message DONE.			

The next table shows the numeric ranks obtained by the program:

Index	Array Value	Rank
1	82	6
2	67	14
3	91	3
4	98	1
5	74	11
6	52	15
7	86	4
8	95	2
9	79	9
10	78	10
11	84	5
12	80	8
13	69	13
14	81	7
15	73	12

Spearman Rank Program NCS_SPEARANK1

Overview

The NCS_SPEARANK1 program calculates the Spearman rank correlation coefficient for an array of paired ranked values.

The program offers the following operations:

- Add paired rank values.
- Delete paired rank values.
- Edit paired rank values.
- Swap paired rank values.
- View specific paired rank values.
- View the values of the paired rank array.
- Calculate the Spearman rank correlation coefficient and related statistics.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Given an array of n paired rank values (x,y), the difference for each paired ranks is:

$$D_i = x_i - y_i$$

The Spearman rank correlation coefficient is obtained using:

$$r_s = 1 - \frac{6 \sum_{i=1}^{n} D_i^2}{n(n^2-1)}$$

To test the null hypothesis:

H_0: x and y are independent

We calculate an approximation to the inverse normal statistic Z (for $n \geq 10$) using the following equation:

$$Z = r_s \sqrt{n-1}$$

If the calculated Z does not exceed Z_α then we cannot reject the null hypothesis that states x and y are independent.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a pair of rank values.		XEQ A	X?		
	Enter a value for x.	x	R/S	Y?		
	Enter a value for y.	y	R/S	Number of data points		
	Enter the next paired rank values. Repeat this step until you have entered all of the paired rank values.	x	R/S	X?		
3	Edit a paired rank values.		XEQ E	I?		
	Enter the index for the value.	Index	R/S	X?		
	View the value and optionally enter a replacement value for x.	x	R/S	Y?		
	View the value and optionally enter a replacement value for y.	y	R/S	[x,y]		
4	View the data while pausing at each value. The program displays the values in the table by columns.		XEQ V	I?		
	Enter the index for the value.	Index	R/S	[x,y]		
5	View the data while pausing at each value.		XEQ W	[x,y]		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
6	Calculate the degrees of freedom, Spearman rank correlation coefficient, and Z value.		XEQ C	Z	rs	Degrees of freedom
7	Display the program name. The program pauses to display the program name.		XEQ U	NCS-SPEARANK1		

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a new data point	C	Count for number of data points	1	Index invalid when set
D	Delete an existing data point	D	Sum of squared differences in the values of paired ranks		
E	Edit a data point	R	Spearman rank correlation coefficient		
V	View a data point	Z	Inverse normal pdf		
W	View all data points	X	Rank of x		
I	Initialize	Y	Rank of y		
S	Swap two data points	I	Index, used		
C	Calculate statistics	J	Index, used		
U	Display program name	1	Data point #1		
Labels for internal use		2	Data point #2		
H	Validate indices and display error message if an index is invalid	3	Data point #3		
G	Prompt for x and y values	…			
R	Display the message INVALID DATA	#C	Data point #C		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize data points
I002	0	
I003	STO C	
I004	RTN	
A001	LBL A	Add a new data point
A002	XEQ G	Get the data point
A003	1	Increment data point counter
A004	RCL+ C	
A005	STO C	
A006	STO I	
A007	R↓	
A008	STO (I)	Store the new data point
A009	R↑	
A010	RTN	
E001	LBL E	Edit a data point
E002	INPUT I	Get the index of the data point to overwrite
E003	XEQ H	Validate the index
E004	XEQ G	Get the new data point
E005	STO (I)	
E006	RTN	
D001	LBL D	Delete an existing data point
D002	INPUT I	Get the index of the data point to delete
D003	XEQ H	Validate the index
D004	RCL C	
D005	x=y?	Is the data point to delete the last one?
D006	GTO D020	Skip loop to overwrite data point
D007	RCL I	
D008	1	
D009	+	
D010	STO J	
D011	RCL (J)	Start loop to overwrite deleted data point
D012	STO (I)	
D013	1	
D014	STO+ I	
D015	STO+ J	
D016	RCL I	
D017	RCL C	
D018	x<y?	Continue overwriting?
D019	GTO D011	Resume with next iteration
D020	1	Decrement data point counter
D021	STO– C	

Line	Command	Comments
D022	RCL C	
D023	RTN	
H001	LBL H	Validate an index
H002	CF 1	
H003	RCL C	
H004	x↔y	
H005	x>y?	Index exceed data point counter?
H006	SF 1	
H007	x≤0?	Index is 0 or less?
H008	SF 1	
H009	FS? 1	Invalid index?
H010	GTO R001	Display error message
H011	RTN	
S001	LBL S	Swap data points
S002	INPUT I	
S003	XEQ H	Validate index
S004	INPUT J	
S005	XEQ H	Validate index
S006	RCL (I)	Recall data points and place them on the stack
S007	RCL (J)	
S008	STO (I)	Store data point in swapped indices
S009	x⇆y	
S010	STO (J)	
S011	RTN	
V001	LBL V	View a data point
V002	INPUT I	Get the index of the data point to view
V003	XEQ H	Validate index
V004	FS? 1	Index out or range?
V005	GTO R001	
V006	RCL (I)	Recall data point
V007	STOP	
V008	1	
V009	RCL I	
V010	+	
V011	STO I	
V012	GTO V003	
W001	LBL W	View all data points
W002	1	
W003	RCL C	
W004	1E3	
W005	÷	
W006	+	
W007	STO I	Store loop index
W008	RCL I	Start loop to view data points

Line	Command	Comments
W009	RCL (I)	
W010	PSE	
W011	ISG I	
W012	GTO W008	Resume to next loop iteration
W013	RTN	
C001	LBL C	Calculate the rank statistics
C002	1	
C003	RCL C	
C004	1E3	
C005	÷	
C006	+	
C007	STO I	Initialize loop variable
C008	0	
C009	STO D	Initialize sum of squared rank differences
C010	RCL (I)	
C011	[1,0]	
C012	×	Get x(I)
C013	RCL (I)	
C014	[0,1]	Get y(I)
C015	×	
C016	-	Calculate difference in rank
C017	x^2	Square the difference
C018	STO+ D	Update sum of squared differences
C019	ISG I	
C020	GTO C010	
C021	RCL D	
C022	6	
C023	×	
C024	RCL C	
C025	÷	
C026	RCL C	
C027	x^2	
C028	1	
C029	-	
C030	÷	
C031	+/-	
C032	1	
C033	+	Calculate the Spearman rank correlation coefficient
C034	STO R	
C035	RCL C	
C036	1	
C037	-	
C038	√x	

Line	Command	Comments
C039	×	Calculate Z
C040	STO Z	
C041	SF 10	
C042	DF,RS,Z	This is a "tag" created using EQN button
C043	PSE	
C044	CF 10	
C045	RCL C	
C046	1	
C047	-	Calculate the degrees of freedom
C048	RCL R	Recall rs
C049	RCL Z	Recall Z
C050	RTN	Display the results
G001	LBL G	Get a data point
G002	INPUT X	Prompt user for x value
G003	INPUT Y	Prompt user for y value
G004	[X,Y]	Create data point. Enter this statement using the EQN key.
G005	RTN	
R001	LBL R	Display INVALID DATA message
R002	CF 1	
R003	ACOS(2)	Enter this statement using the EQN key.
R004	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-SPEARANK1	This is a "tag" created using EQN button
U004	PSE	Display program name
U005	CF 10	
U006	RTN	

Example

Let's look at an example that exercises various routines in the NCS_SPEARANK1 program. The next table contains the raw observations and their ranks. You can obtain the rank values using the program NCS_RANKER1. Since the NCS_RANKER1 program generates the ranks for each variable, you have to separately key in the each set of ranks in the NCS_SPEARANK1 program.

Student ID	English Grade E	French Grade, F	x = Rank of E	y = Rank of F
1	82	81	6	7
2	67	75	14	11
3	91	85	3	4
4	98	90	1	2
5	74	80	11	8
6	52	60	15	15
7	86	94	4	1
8	95	78	2	9
9	79	83	9	6
10	78	76	10	10
11	84	84	5	5
12	80	69	8	13
13	69	72	13	12
14	81	88	7	3
15	73	61	12	14

After you enter the above data, calculate the degrees of freedom, Spearman rank correlation coefficient, and Z value. Use the value of Z to check the null hypothesis at a 95% confidence level. If you enter erroneous values, execute the XEQ E command to edit these values. You can also view the data using the command XEQ W before (or even after) you calculate the statistical results.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Enter the data.		XEQ A	X?
	Enter the first value of x.	6	R/S	Y?
	Enter the first value of y.	7	R/S	1.00
	Repeat the above steps to enter the rest of the paired rank values.			
3	Calculate degrees of freedom, Spearman rank correlation coefficient, and Z value.		XEQ C	 Displays the Spearman

Step #	Step/Substep	Input	Command	Output
				rank correlation coefficient, and Z value
	View the degrees of freedom.		R↓	14.00 0.76 Displays the degrees of freedom, Spearman rank correlation coefficient

Since the calculated Z value of 2.85 exceeds the value of $Z_{0.05}$ (equal to 1.65) we cannot accept the null hypothesis, stating that there paired rank values are independent.

Appendix A contains a new approximation that I recently developed for inverse normal function. Using the approximation for Z, shown in the appendix, you get Z = 1.61 for a significance level of 0.05.

Unpaired Data T-Test Program NCS_UNPAIRED1

Overview

The NCS_UNPAIRED1 program implements the Student-t test for the means of unpaired data. The program allows you to enter multiple groups of unpaired data and then select any two groups to test whether or not they have the same statistical means. This feature allows you to compare the means of multiple unpaired groups without rekeying in the data for each comparison.

The program does not store the raw data for each set. Doing so and offering the expected features of adding, editing, and deleting data points can be taxing on the calculator resources. Instead, the program stores the statistical summations and supports the following operations:

1. Adding and deleting data within the currently edited group.
2. Ending the work on a data group and starting a new data group. The program can support as many groups as allowed by the number of unnamed memory registers. The program stores the number of data points, the sum of observations, and the sum of observations squared for each data group using a 3-D vector.
3. Recalling a previously stored data group to add or delete observations. This feature gives flexibility to the user by not *sealing* the data once you move on to another data group.
4. Selecting any two data groups to calculate the Student-t statistic that is used to test the null hypothesis.

Brief Theory

This section briefly lists the statistical variables and equations used in the calculations.

Give data groups x and y of unequal number of data elements. The null hypothesis is that the the difference between the means of the two data groups is the value d:

$H_0 : \mu_1 - \mu_2 = d$

The difference for each paired observations is:

$$D_i = x_i - y_i$$

The mean for each data set is:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n1} x_i$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n2} y_i$$

The Student-t statistic is calculated using the following equation:

$$t = A / (B \cdot C)$$

Where:

$$A = \bar{x} - \bar{y} - d$$

$$B = \sqrt{\frac{1}{n1} + \frac{1}{n2}}$$

$$C = \sqrt{\frac{\sum x^2 - n1\,\bar{x} + \sum y^2 - n2\,\bar{\bar{y}}}{n1 + n2 - 2}}$$

If the calculated t statistic does not exceed the value of $t_{\alpha/2,n1+n2-2}$ then we cannot reject the null hypothesis stating that the paired data have a difference in mean values equal to d.

Instructions

The following table shows the steps involved in using the various parts of the program. The commands show the label truncated to the first letter when step 001 is implied. Thus XEQ A, for example, is the same as typing XEQ A001 or XEQ A [Enter]. This convention is used throughout this document.

When it comes to the output, the program pauses to display an output tag for each output, before showing the numerical values of that output. The output usually displays results in pairs to take advantage of the double display lines of the calculator. Using output tags labels the output and makes the program more user-friendly. In addition, when a step mentions viewing two output values, call them A and B, then A appears in the Y stack register and B appears in the X stack register. Likewise, when a step mentions the output of three values, call them A, B, and C, then A is placed in the Z stack register, B appears in the Y stack register, and C appears in the X stack register. Since the calculator displays the Y and X stack registers, the value of A is not immediately visible. You will need to use the R↓ key to view that value.

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
1	Initialize the program.		XEQ I	0		
2	Add a new data point.		XEQ A	X?		
	Enter a value for x.	x	R/S	Number of data points		
	Enter another value.		R/S	X?		
	Resume with the steps above.					

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
3	Delete a data point.		XEQ D	X?		
	Enter a value for x.	x	R/S	Number of data points		
4	End the data entry for the current group		XEQ E	[n, Σx, Σx^2]		
	Resume with step 2 to enter new data.					
5	Recall summations of previous data group.		XEQ R	I?		
	Enter the index of the data point to view. If you enter an invalid index the program displays the message INVALID DATA.	Index	R/S			
	You can resume at step 2 to add more data and/or step 3 to delete data. When you are done, execute step 4.					
6	To calculate the Student-t and the degrees of freedom.		XEQ C	I?		
	Enter the index of the one group. If you enter an invalid index (greater than the number of actual sets), the program displays the INVALID DATA error message. Entering an index of 0 or less has unpredictable effects!	Index	R/S	J?		
	Enter the index of the other group. If you enter an invalid index (equal to the index for the other group, or greater than the number of actual sets), the program displays the INAVLID DATA	Index	R/S	D?		

Step #	Step/Substep	Input	Command	Output X	Output Y	Output Z
	error message. Entering an index of 0 or less has unpredictable effects!					
	Enter the value for the tested mean difference d.	d	R/S	Degrees of freedom	Student-t	
7	Display the program name. The program pauses to display the program name.		XEQ U	NCS-UNPAIRED1		

Program Resources

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
A	Add a data point to the current group	A	n1	1	Set = Recall group mode
C	Calculate Student-t and degrees of freedom for 2 groups	B	Mean of x		
D	Delete a data point from current group	C	Σx^2		
E	End the data entry for the current group	D	difference between mean d		
I	Initialize the program	E	n2		
R	Recall summations of previous data group	F	Mean of y		
U	Display the program name	G	Σy^2		
		H			
Internally used labels		I	Index of stat data for x		
P	Parse data recalled from unamed variables	J	Index of stat data for y		
		K			
		L			
		M			
		N	Number of data groups		
		O			
		P			
		Q			

Labels	Purpose	Memory Registers	Purpose	Flags	Purpose
		R			
		S			
		T			
		U			
		V			
		W			
		X			
		Y	Used		
		Z	Used		
		1	[n, Σx, sum of Σx^2] for group 1		
		2	[n, Σx, sum of Σx^2] for group 2		
		…			
		n	[n, Σx, sum of Σx^2] for group n		

Program Listing

Line	Command	Comments
I001	LBL I	Initialize the program
I002	ClΣ	
I003	0	
I004	STO N	Initialize data group counter
I005	CF 1	Clear recall data group mode
I006	RTN	
A001	LBL A	Add data point to the current data group
A002	INPUT X	
A003	RCL X	
A004	Σ+	
A005	STOP	
A006	GTO A002	
D001	LBL D	Delete a data point from the current data group
D002	INPUT X	
D003	RCL X	
D004	Σ-	
D005	RTN	
E001	LBL E	End the data entry for the current data group
E002	FS? 1	Is recalled data group mode on?
E003	GTO E008	
E004	1	
E005	STO+ N	

Line	Command	Comments
E006	RCL N	
E007	STO I	
E008	[n, Σx, Σx^2]	
E009	STO (I)	
E010	ClΣ	Clear summation registers block
E011	CF 1	Clear recalled data group mode
E012	RTN	
P001	LBL P	
P002	STO Z	
P003	[0,0,1]	
P004	×	
P005	RCL Z	
P006	[0,1,0]	
P007	×	
P008	RCL Z	
P009	[1,0,0]	
P010	×	
P011	RTN	
C001	LBL C	Calculate statistics
C002	INPUT I	Prompt user for index of first data group
C003	RCL I	
C004	RCL N	
C005	x<y?	Validate input
C006	ASIN(2)	Generate INVALID DATA message
C007	INPUT J	Prompt user for index of second data group
C008	RCL J	
C009	RCL N	
C010	x<y?	Validate input
C011	ASIN(2)	Generate INVALID DATA message
C012	RCL I	
C013	RCL J	Check if both indices I and J are equal
C014	x=y?	
C015	ASIN(2)	Generate INVALID DATA message
C016	INPUT D	Prompt user for the tested difference in the means
C017	STO D	
C018	RCL (I)	
C019	XEQ P	
C020	STO A	Store n1
C021	÷	Calculate mean of x
C022	STO B	Store mean of x
C023	R↓	
C024	STO C	Store Σx^2
C025	RCL (J)	

Line	Command	Comments
C026	XEQ P	
C027	STO E	Store n2
C028	÷	Calculate mean of y
C029	STO F	Store mean of y
C030	R↓	
C031	STO G	Store Σy^2
C032	RCL C	Start calculating Student-t
C033	RCL A	
C034	RCL B	
C035	x^2	
C036	×	
C037	-	
C038	RCL G	
C039	+	
C040	RCL E	
C041	RCL F	
C042	x^2	
C043	×	
C044	-	
C045	RCL A	
C046	RCL E	
C047	+	
C048	2	
C049	-	
C050	STO Y	
C051	÷	
C052	√x	Finish calculating second square root term
C053	1/x	
C054	STO Z	
C055	RCL A	
C056	1/x	
C057	RCL E	
C058	1/x	
C059	+	
C060	√x	Finish calculating first square root term
C061	STO÷ Z	
C062	RCL B	
C063	RCL F	
C064	-	
C065	RCL D	
C066	-	Calculate Xmean - Ymean - d
C067	STO× Z	Finish calculating Student-t
C068	SF 10	

Line	Command	Comments
C069	T,DF	This is a "tag" created using the EQN button
C070	PSE	
C071	CF 10	
C072	RCL Z	Recall Student-t
C073	RCL Y	Recall degrees of freedom
C074	RTN	
R001	LBL R	Recall an existing data group
R002	FS? 1	Already editing another recalled group?
R003	GTO R006	
R004	XEQ E	
R005	GTO R009	
R006	[n, Σx, Σ x^2]	
R007	STO (I)	
R008	ClΣ	
R009	INPUT I	
R010	SF 1	Set group recall mode
R011	RCL (I)	
R012	XEQ P	Parse recalled data group
R013	-27	
R014	STO J	
R015	R↓	
R016	STO (J)	Store n
R017	R↓	
R018	1	
R019	STO- J	
R020	R↓	
R021	STO (J)	Store sum of x
R022	R↓	
R023	2	
R024	STO- J	
R025	R↓	
R026	STO (J)	Store sum of x^2
R027	RTN	
U001	LBL U	Display program name
U002	SF 10	
U003	NCS-UNPAIRED1	This is a "tag" created using the EQN button
U004	PSE	
U005	CF 10	
U006	RTN	

Example

Let's look at an example uses the routines in the NCS_UNPAIREDT1 program. The next table contains unpaired data points that we want to test for equal statistical means (that is, d = 0).

X	y
79	91
84	103
108	90
114	113
120	108
103	87
122	100
120	80
	99
	84

After you enter the above data, perform the following steps:

1. Calculate degrees of freedom and Student-t statistic.
2. Test the null hypothesis at a 95% confidence level.

The following table shows the steps using the FIX 2 display mode.

Step #	Step/Substep	Input	Command	Output
1	Initialize the program.		XEQ I	0
2	Add the first data point in the first data group.		XEQ A	X?
	Enter a value for x.	79	R/S	1.00
	Repeat the above steps for the rest of the values in the first data groups.			
3	End the first data group.		XEQ E	1.00 [8.00,85.00,92.2
4	Add the first data point in the second data group.		XEQ A	X?
	Enter a value for x.	91	R/S	1.00
	Repeat the above steps for the rest of the values in the second data groups.			
5	End the second data group.		XEQ E	2.00 [10.00,92.500,88,

Step #	Step/Substep	Input	Command	Output
6	Calculate statistical results.		XEQ C	I?
	Enter the index for group 1.	1	R/S	J?
	Enter the index for group 2.	2	R/S	D?
	Enter the difference (between the means) of 0.	0	R/S	1.73 16.00 Displays the values for the Student-t and the degrees of freedom.

Since the calculated t value of 1.73 does not exceed the value of $t_{16,0.025}$ of 2.12 we cannot reject the null hypothesis stating that the unpaired samples have a zero difference in means.

Appendix A contains a new approximation that I recently developed for inverse Student-t function. The appendix also contains tables for selected values.

Appendix A: Inverse Probability Distribution Functions

This appendix offers information related to common inverse probability distribution functions. The functions include:

- The inverse normal probability distribution function
- The inverse student-t probability distribution function
- The inverse Chi-square probability distribution function
- The inverse F probability distribution function

The Inverse Normal Probability Distribution Function

You can calculate the inverse Normal value using well known approximations. Recently, I was able to obtain the following approximation:

$Qinv = \sqrt[3]{[x^3 / (0.063493571 - 0.198675274\, x^2 - 0.017195871\, x^3)]}$

Where $x = 0.5 - \alpha$, Qinv is the inverse normal distribution, and α is the significance level in decimals.

The Inverse Student-t Probability Distribution Function

You can also calculate the inverse Student-t value using well known approximations. Recently, I was able to obtain a set of approximations that fits the inverse Student-t and the degrees of freedom using the following model:

$Tinv = \exp(A + B / df + C / df^2)$

The following table shows the values for the constant A, B, and C for different values of the significance level α:

Significance Level α	A	B	C
0.200	0.248069936	0.660674	0.226537
0.150	0.364320592	0.767873	0.308868
0.100	0.497661825	0.925738	0.445297
0.050	0.672951400	1.208789	0.734348
0.025	0.807141675	1.503440	1.093993

The above family of equations is easy to program in any programmable calculator.

Following a more traditional approach, here is the table for the inverse Student-t probability distribution function. The last row of the table contains values for the inverse normal probability distribution function.

Degrees of Freedom	α = 0.100	α = 0.050	α = 0.025	α = 0.010
1	3.078	6.314	12.706	31.821
2	1.886	2.920	4.303	6.965
3	1.638	2.353	3.182	4.541
4	1.533	2.132	2.776	3.747
5	1.476	2.015	2.571	3.365
6	1.440	1.943	2.447	3.143
7	1.415	1.895	2.365	2.998
8	1.397	1.860	2.306	2.896
9	1.383	1.833	2.262	2.821
10	1.372	1.812	2.228	2.764
11	1.363	1.796	2.201	2.718
12	1.356	1.782	2.179	2.681
13	1.350	1.771	2.160	2.650
14	1.345	1.761	2.145	2.624
15	1.341	1.753	2.131	2.602
16	1.337	1.746	2.120	2.583
17	1.333	1.740	2.110	2.567
18	1.330	1.734	2.101	2.552
19	1.328	1.729	2.093	2.539
20	1.325	1.725	2.086	2.528
21	1.323	1.721	2.080	2.518
22	1.321	1.717	2.074	2.508
23	1.319	1.714	2.069	2.500
24	1.318	1.711	2.064	2.492
25	1.316	1.708	2.060	2.485
26	1.315	1.706	2.056	2.479
27	1.314	1.703	2.052	2.473
28	1.313	1.701	2.048	2.467
29	1.311	1.699	2.045	2.462
30	1.310	1.697	2.042	2.457
31	1.309	1.696	2.040	2.453
32	1.309	1.694	2.037	2.449
33	1.308	1.692	2.035	2.445
34	1.307	1.691	2.032	2.441
35	1.306	1.690	2.030	2.438
36	1.306	1.688	2.028	2.434
37	1.305	1.687	2.026	2.431
38	1.304	1.686	2.024	2.429
39	1.304	1.685	2.023	2.426
40	1.303	1.684	2.021	2.423

Degrees of Freedom	α = 0.100	α = 0.050	α = 0.025	α = 0.010
50	1.299	1.676	2.009	2.403
60	1.296	1.671	2.000	2.390
70	1.294	1.667	1.994	2.381
80	1.292	1.664	1.990	2.374
90	1.291	1.662	1.987	2.368
100	1.290	1.660	1.984	2.364
Infinity	1.282	1.645	1.960	2.326

The Inverse Chi-square Probability Distribution Function

You can determine the value of chi-square distribution from tables or well know approximations. Recently, I developed the following approximation:

$$\chi^2 = (2.24636214365125 — 3.072059274\ \alpha^{1/3} + 1.008402684\ \sqrt{df})^2$$

Where α is the significance level (and 100(1-α)% is the confidence level) and df is the degrees of freedom. The above approximation is easy to use with any programmable calculator.

Following a more traditional approach, here is the table for the inverse chi-square probability distribution function.

Degrees of Freedom	α = 0.100	α = 0.050	α = 0.025	α = 0.010
1	2.706	3.841	5.024	6.635
2	4.605	5.991	7.378	9.210
3	6.251	7.815	9.348	11.345
4	7.779	9.488	11.143	13.277
5	9.236	11.070	12.833	15.086
6	10.645	12.592	14.449	16.812
7	12.017	14.067	16.013	18.475
8	13.362	15.507	17.535	20.090
9	14.684	16.919	19.023	21.666
10	15.987	18.307	20.483	23.209
11	17.275	19.675	21.920	24.725
12	18.549	21.026	23.337	26.217
13	19.812	22.362	24.736	27.688
14	21.064	23.685	26.119	29.141
15	22.307	24.996	27.488	30.578
16	23.542	26.296	28.845	32.000
17	24.769	27.587	30.191	33.409
18	25.989	28.869	31.526	34.805
19	27.204	30.144	32.852	36.191
20	28.412	31.410	34.170	37.566
21	29.615	32.671	35.479	38.932
22	30.813	33.924	36.781	40.289

Degrees of Freedom	α = 0.100	α = 0.050	α = 0.025	α = 0.010
23	32.007	35.172	38.076	41.638
24	33.196	36.415	39.364	42.980
25	34.382	37.652	40.646	44.314
26	35.563	38.885	41.923	45.642
27	36.741	40.113	43.195	46.963
28	37.916	41.337	44.461	48.278
29	39.087	42.557	45.722	49.588
30	40.256	43.773	46.979	50.892
31	41.422	44.985	48.232	52.191
32	42.585	46.194	49.480	53.486
33	43.745	47.400	50.725	54.776
34	44.903	48.602	51.966	56.061
35	46.059	49.802	53.203	57.342
36	47.212	50.998	54.437	58.619
37	48.363	52.192	55.668	59.893
38	49.513	53.384	56.896	61.162
39	50.660	54.572	58.120	62.428
40	51.805	55.758	59.342	63.691
50	63.167	67.505	71.420	76.154
60	74.397	79.082	83.298	88.379
70	85.527	90.531	95.023	100.425
80	96.578	101.879	106.629	112.329
90	107.565	113.145	118.136	124.116
100	118.498	124.342	129.561	135.807

The Inverse F Probability Distribution Function

You can look up the value of F in statistical tables, use traditional approximations, or use the following approximation that I recently developed for $df_1 > 4$ and $df_2 > 4$:

$$\begin{aligned} F(\alpha, df1, df2) = [&1.213555792 - 0.974834275\,\alpha + 2.014853337\,\alpha^2 \\ &- 0.829609001 / df_1 - 0.93038369 \ln(\alpha) / df_1 \\ &- 1.927707851 / df_2 - 2.302124211 \ln(\alpha) / df_2 \\ &+ 0.875932791 / \sqrt{(df_1 \times df_2)} \\ &+ 0.717398799 \ln(\alpha) / \sqrt{(df_1 \times df_2)}]^2 \end{aligned}$$

Despite the use of nine constants, the above approximation is much simpler than the legacy approximation for the inverse F probability distribution function.

Here is a partial table for the inverse F probability distribution function, for the 0.05 significance level:

	Inverse F for α = 0.05									
df1 / df2	1	2	3	4	5	6	7	8	8	10
1	161.45	199.50	215.71	224.58	230.16	233.99	236.77	238.88	238.88	241.88
2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.37	19.40
3	10.13	9.55	9.28	9.12	9.01	8.94	8.89	8.85	8.85	8.79
4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.04	5.96
5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.82	4.74
6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.15	4.06
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.73	3.64
8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.44	3.35
9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.23	3.14
10	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.07	2.98

Here is a partial table for the inverse F probability distribution function, for the 0.025 significance level:

	Inverse F for α = 0.025									
df1 / df2	1	2	3	4	5	6	7	8	8	10
1	647.79	799.50	864.16	899.58	921.85	937.11	948.22	956.66	956.66	968.63
2	38.51	39.00	39.17	39.25	39.30	39.33	39.36	39.37	39.37	39.40
3	17.44	16.04	15.44	15.10	14.88	14.73	14.62	14.54	14.54	14.42
4	12.22	10.65	9.98	9.60	9.36	9.20	9.07	8.98	8.98	8.84
5	10.01	8.43	7.76	7.39	7.15	6.98	6.85	6.76	6.76	6.62
6	8.81	7.26	6.60	6.23	5.99	5.82	5.70	5.60	5.60	5.46
7	8.07	6.54	5.89	5.52	5.29	5.12	4.99	4.90	4.90	4.76
8	7.57	6.06	5.42	5.05	4.82	4.65	4.53	4.43	4.43	4.30
9	7.21	5.71	5.08	4.72	4.48	4.32	4.20	4.10	4.10	3.96
10	6.94	5.46	4.83	4.47	4.24	4.07	3.95	3.85	3.85	3.72

Here is a partial table for the inverse F probability distribution function, for the 0.01 significance level:

	Inverse F for α = 0.01									
df1 / df2	1	2	3	4	5	6	7	8	8	10
2	98.50	99.00	99.17	99.25	99.30	99.33	99.36	99.37	99.37	99.40
3	34.12	30.82	29.46	28.71	28.24	27.91	27.67	27.49	27.49	27.23
4	21.20	18.00	16.69	15.98	15.52	15.21	14.98	14.80	14.80	14.55
5	16.26	13.27	12.06	11.39	10.97	10.67	10.46	10.29	10.29	10.05
6	13.75	10.92	9.78	9.15	8.75	8.47	8.26	8.10	8.10	7.87
7	12.25	9.55	8.45	7.85	7.46	7.19	6.99	6.84	6.84	6.62
8	11.26	8.65	7.59	7.01	6.63	6.37	6.18	6.03	6.03	5.81
9	10.56	8.02	6.99	6.42	6.06	5.80	5.61	5.47	5.47	5.26
10	10.04	7.56	6.55	5.99	5.64	5.39	5.20	5.06	5.06	4.85

Document Update History

Version	*Date*	*Update*
1.00	9/1/2007	Initial document release.