Supplementary Guide

instat
instat
instat

instat

instat
instat
instat
instat
instat

a statistics package for the BBC micro

Statistical Services Centre

UNIVERSITY of READING

INSTAT SUPPLEMENTARY GUIDE

INSTAT SUPPLEMENTARY GUIDE

PART I   SUPPLEMENTARY INTRODUCTORY GUIDE

PART II   SUPPLEMENTARY REFERENCE GUIDE

INSTAT SUPPLEMENTARY GUIDE

INSTAT SUPPLEMENTARY GUIDE

PART I SUPPLEMENTARY INTRODUCTORY GUIDE

CONTENTS

INSTAT SUPPLEMENTARY GUIDE

PART II SUPPLEMENTARY REFERENCE GUIDE

INDEX

PART I SUPPLEMENTARY INTRODUCTORY GUIDE

1 INTRODUCTION

The first release of INSTAT has been available for about two years and has found its way into a wide variety of sites. INSTAT is being used in industry, Government departments, medical and agricultural research establishments, universities, polytechnics and other colleges. We are delighted to note that schools are also showing an interest in INSTAT.

Our original aim was to provide facilities on the BBC Micro similar to those offered by mainframe statistics packages. Of course, given the limitations of a small computer, this was a rather ambitious goal. On the other hand, we feel that we have had some success in exploiting the strengths of the BBC Micro, and in particular BBC BASIC, in order to fulfill this aim.

The feedback that the development team has received from users and reviewers has been most encouraging, and we have consequently decided that the effort required to produce this upgrade is more than justified. The upgrade consists of some new facilities and some changes and enhancements to existing facilities. All bugs that we know of have been fixed (we hope).

Many of the changes that have been introduced in this upgrade are the result of suggestions from users. We wish to thank all who have fed us new ideas and pointed out bugs and inadequacies in INSTAT. In order to keep abreast of our users' needs, we depend rather heavily on this kind of feedback, and we hope that users will not hesitate to keep us supplied with new ideas and, of course, criticism.

This Supplementary Guide is intended to accompany the existing Introductory and Reference Guides, not to replace them. It describes the new facilities that have been added to INSTAT and those existing facilities which have been enhanced. The next editions of the INSTAT User Guides, when available, will incorporate all of the new features and amendments. The Supplementary Guide is in two parts, corresponding to the two parts of the existing User Guide. The first part gives informal descriptions of the new facilities with examples of their use, while the second part is the Supplementary Reference Guide.

Development of INSTAT continues. A major new departure is the development of an MS-DOS version of INSTAT to run on the IBM PC and compatible machines. Work on this version is now well advanced and we hope to release it in the very near future.

2: ROW STATISTICS

The fundamental data structure in an INSTAT worksheet is the column of data. Almost all commands that manipulate data, or perform calculations on data, operate on columns.

Situations are not uncommon, however, in which we would like to have summary statistics of *rows* of data as well as columns. Presumably, if we had wanted to perform operations only on rows, and not on columns, then we would have entered the data into the worksheet the other way around. So we are considering those situations where most of our data manipulation will operate on columns, but we would also like to have some facilities for working with rows.

Many row operations are already available in INSTAT with simple calculations on columns. The sum of columns X1, X2 and X3, for example, can be saved in X8 by

: *X8 = X1+X2+X3*

However, not many summary statistics are easily available by direct calculation. Consider, for example, the operations that would be required to produce a column of standard deviations of the rows of ten columns.

The new ROW command has been provided for such situations and its use is shown in Figure 2.1. For example, suppose the data in X7 to X10 give the results for 6 shildren in 4 tests. Then summary statistics about each **test** are normal column operations. We may also wish to find the mean and standard deviation for each **child** over the 4 tests. The command

: *ROW X7-X10; MEA X11; SDE X12*

gives the 6 means in X11 and the corresponding standard deviations in X12.

The subcommands currently available are ;COUnt, ;MEAn , ;MINimum, ;MAXimum, ;SDEviation and ;SUM.

Figure 2.1 The calculation of ROW statistics

```
: disp x7-x10

Row        X7        X8        X9        X10

  1        63        20        19         45
  2        77        26        23         38
  3        61        17        13         49
  4        73        22        27         56
  5        45        22        15         60
  6        62        14        17         55
: row x7-x10; mean x11; sdev x12
: disp x11 x12

Row       X11       X12

  1     36.75    21.235
  2        41     24.86
  3        35    23.664
  4      44.5    24.201
  5      35.5    20.761
  6        37     25.02
```

3: FREQUENCY DISTRIBUTIONS WITH THE GROUP COMMAND

Sometimes we want to group the values of a variate into a frequency distribution. The reason for doing this is often to compare the observed frequencies with the expected frequencies obtained by assuming a particular theoretical probability distribution. The expected frequencies are easily obtainable, for a variety of distributions, from the FREquencies command. But getting a column of observed frequencies of the same format as these expected frequencies has hitherto been rather a cumbersome procedure. Two methods for deriving the observed frequency distribution were described in the INSTAT User Guide, Section 14.5. Neither of these methods is particularly satisfactory, and for this reason a new command GROup has been devised.

GROup takes the data in a column and outputs (and optionally saves) a column of frequencies. The command can be used in two ways:

(i) to produce frequencies corresponding to contiguous intervals
(ii) to produce frequencies of occurrences of actual values.

The first usage is intended for continuous variates, while the second is likely to be more useful for discrete data.

When the GROup command is used without subcommands it simply counts the occurrences of each different value in the data, see Figure 3.1. If the ;VALues subcommand is used then the number of occurrences at each of the specified values is given.

The subcommand ;BOUndaries specifies the boundaries between successive intervals for continuous data. This is illustrated in Figure 3.2. In the next section, Figure 4.4 takes this example further, and shows how to use the columns of frequencies obtained from GROup, together with the new CHIsquare command, to test whether a set of data could be from a normal distribution.

Figure 3.1 which illustrates the use of the GROup command for discrete data, also shows the effect of the ;PERcentages subcommand. Percentage and cumulative percentage frequencies are only displayed and not saved. Should they be required for further calculations, however, it is easy to derive them from the frequencies using calculations on columns. The CUSUM function (see CALculate in the Reference Guide) could be used for cumulative frequencies.

Figure 3.1 Using the GROup command with discrete data

```
: disp x7

X7
        2       1       2       3       2       5
        1       3       7       5       5       1
        3       1       1       4       6       2
        8       3       1       1       4       3
        0       3       3       2       6       1
        1      10
:
: group x7

Values      Count
-----------------
  0           1
  1           9
  2           5
  3           7
  4           2
  5           3
  6           2
  7           1
  8           1
 10           1
-----------------
Total        32
:
: enter x8
data  1: (0]10)
data 12: eod
:
: group x7; values x8; percents

Values      Count        %       Cum. %
----------------------------------------
  0           1        3.13       3.13
  1           9       28.13      31.25
  2           5       15.63      46.88
  3           7       21.88      68.75
  4           2        6.25      75.00
  5           3        9.38      84.38
  6           2        6.25      90.63
  7           1        3.13      93.75
  8           1        3.13      96.88
  9           0        0.00      96.88
 10           1        3.13     100.00
----------------------------------------
Total        32      100.00
```

Figure 3.2 GROuping continuous data

```
: disp x5; FIX 0
  X5
 260.    350.    176.    288.    249.    160.    218.    348.
 111.    316.    214.    195.    118.    230.    257.    155.
  74.    192.    161.    184.     87.    110.    416.    357.
 403.     40.    179.    219.    175.    273.    421.     19.

: ? min(x5)
 18.701

: ? max(x5)
  421.3

: ? count(x5)
     32

: ent x15
data  1: (50]400!50)
data  9: eod

: disp x15

X15
      50         100         150         200         250         300
     350         400
:
: group x5 x16; boundaries x15

Values       Count
------------------
<= 50           2
to100           2
to150           3
to200           9
to250           5
to300           4
to350           2
to400           2
 >400           3
------------------
Total          32
```

4: CHI-SQUARE TESTS

INSTAT now has a CHIsquare command which can be used both for performing tests of independence in two-way contingency tables and for goodness-of-fit tests.

4.1 Contingency Tables

There are two ways to input a contingency table for the CHIsquare command - the cell counts can either be arranged in a number of columns, or in a single column. The example in Figure 4.1 shows how the CHIsquare command is used in the former case. The columns X7, X8 and X9 represent the columns of the contingency table. After the CHIsquare test, the PRObability command can be used, if necessary, to find the exact significance level.

Figure 4.1 The CHI-square test for a 4 by 3 contingency table

```
: disp x7-x9
 Row          X7          X8          X9

  1           75          60          65
  2          160         115         175
  3          100          65         135
  4           15          10          25

:
: chisq x7-x9
   Row   Observed   Expected    Chi
    1       75        70.0     -0.6
    2      160       157.5     -0.2
    3      100       105.0      0.5
    4       15        17.5      0.6
    5       60        50.0     -1.4
    6      115       112.5     -0.2
    7       65        75.0      1.2
    8       10        12.5      0.7
    9       65        80.0      1.7
   10      175       180.0      0.4
   11      135       120.0     -1.4
   12       25        20.0     -1.1
Chi-square value 11.0 with   6 d.f.
:
: prob 11.0; chi 6

 Chi-square  dist. with 6 d.f.
 Probability > 11 = 0.0884
```

There are subcommands which can be used to convert the table into a single column, and to save factors representing the corresponding row and column levels. To do this, the command could be

```
: chisq x7-x9; counts x15; rows x16; cols x17
```

The resulting column of counts, X15, is then saved as a 'table' column with X16 and X17 the associated row and column factors.

Now suppose that the same table was originally entered as a single column X10, with factor columns X11 and X12 indexing the rows and columns, respectively, of the table. Figure 4.2 shows the use of the CHIsquare command when the data are stored in this form. The example also illustrates the ;FVAls and ;RESids subcommands for saving fitted (or 'expected' values) and standardised residuals, respectively. The standardised residuals are defined by

st. res. = (observed - fitted) / √(fitted)

These residuals are just the 'chi' values displayed as the final column in the output.

Figure 4.2 An alternative layout for a contingency table

```
: disp x10-x12

Row        X10        Row        Col

  1         75          1          1
  2        160          2          1
  3        100          3          1
  4         15          4          1
  5         60          1          2
  6        115          2          2
  7         65          3          2
  8         10          4          2
  9         65          1          3
 10        175          2          3
 11        135          3          3
 12         25          4          3

: factor 'Row 4 'Col 3
:
: chisq x10; fac 'Row 'Col; fvals x13; resids x14
   Row   Observed   Expected   Chi
   1        75         70.0    -0.6

     .....  same as Figure 4.1  .....

   12       25         20.0    -1.1
Chi-square value 11.0 with    6 d.f.
```

Notice that the output from the CHIsquare command always displays the counts and expected values as columns. If tabular format is preferred, for the data or the results, then the PREsent command can be used. Figure 5.3 shows how this might be done for the same example.

Figure 4.3 Presenting the results from a chi-square test in a 2-way table

```
: name x13 'Fitted x14 'Resids
: present x13; row x11; col x12; width 12

  Table for Fitted: Col BY Row

        Col             1            2            3
---------------------------------------------------
        Row ---------------------------------------
          1 :          70           50           80
          2 :       157.5        112.5          180
          3 :         105           75          120
          4 :        17.5         12.5           20

: present x14; row x11; col x12; fix 2; width 12

 Table for Resids: Col BY Row

        Col             1            2            3
---------------------------------------------------
        Row ---------------------------------------
          1 :       -0.60        -1.41         1.68
          2 :       -0.20        -0.24         0.37
          3 :        0.49         1.15        -1.37
          4 :        0.60         0.71        -1.12
```

Suppose that the column of counts (X10 in the example) was derived from the TABle command, with associated factor columns X11 and X12 (using the ;ASSoc subcommand with TABle). Then the CHIsquare command can be simplified to

```
: chisq x10
```

The reason this works is that X10 would have been saved as a 'table' column in the worksheet, and INSTAT will 'know' which factor columns are implicitly used to index its rows and columns.

A final point concerning contingency tables is that the continuity correction (due to Yates, 1934), which is only appropriate for 2 by 2 tables, is available with the ;CONtinuity subcommand. Recent research has cast some doubt on the use of the correction, and it should be remembered that when it is used, the test is likely to be conservative - the reported significance level is probably higher than the actual level.

4.2 Goodness-of-Fit Tests

Another common use of chi-square is to test the fit of a probability distribution to a sample of data. The FREquencies command can be used to derive expected frequencies for a number of commonly used distributions, and the new GROup command, described in the previous section, facilitates the calculation of observed frequencies. The CHIsquare command tests for goodness-of-fit between a column, Xn, of observed frequencies and a column of expected frequencies, by means of the subcommand ;EXPected Xm p. Here, p is the number of parameters estimated from the data in deriving the expected frequencies.

To illustrate the test, we continue the example begun in Figure 3.2. There, we used the GROup command to produce a column, X16, of frequencies derived from the data in X5, using interval boundaries in X15. Figure 4.4 shows how to obtain the expected frequencies corresponding to the normal distribution with the same mean and standard deviation as the data, and test the goodness-of-fit with the CHIsquare command.

Before doing the chi-square test, we note from the output of the FREquencies command that there are some small expected frequencies. The output suggests that we should combine rows 1 and 2 and rows 8 and 9. This can be accomplished with a subcommand for CHIsquare, as is shown in the example.

Figure 4.4 Using the CHIsquare command for a goodness of fit test

```
: k1=mean(x5)
: k2=sdev(x5)
: freq 32 x15 x17; normal k1 k2

 Normal dist.  Mean 217.3 and s.d. 106.4
    Values              Expected Frequency
          <=   50                1.86
  50      to  100                2.47
 100      to  150                4.11
 150      to  200                5.50
 200      to  250                5.93
 250      to  300                5.14
 300      to  350                3.59
 350      to  400                2.02
          >   400                1.38
: chisq x16; expected x17 2; group 1 2; group 8 9
   Row   Observed   Expected   Chi
1  - 2      4          4.3      0.2
   3        3          4.1      0.5
   4        9          5.5     -1.5
   5        5          5.9      0.4
   6        4          5.1      0.5
   7        2          3.6      0.8
8  - 9      5          3.4     -0.9
Chi-square value 4.4 with   4 d.f.
```

5: REGRESSION AND CORRELATION

There are some enhancements to INSTAT's facilities for regression and correlation, and these are described in this section. The program has been modified so that regression and correlation take account of missing values. Other aspects of INSTAT's treatment of missing values are dealt with in Section 7.

5.1 The INDicator Command

The INDicator command produces indicator or 'dummy' variables from a factor column, which are intended to be included in the TERms list when regression models with the factor are to be fitted. The number of dummy variables would normally be equal to the number of levels of the factor, and each one takes the value 1 when the factor has the corresponding level, and 0 otherwise.

Usually the reason for including a factor in a regression model is to investigate differences in the regression equations for the groups defined by the factor levels. This is what many textbooks refer to as 'comparison of regressions'. In order to fit a model where each group has a different slope, it is necessary to fit terms like the dummy variables, except that, where a dummy variable takes the value 1, they take the value of the variate.

These variables could be produced by multiplying the dummy variables by the variate, but the new subcommand ;VARiate for the INDicator command makes it a little easier. The example on comparison of regressions that was used in the INSTAT Introductory Guide, Section 13.5, can be used to illustrate its use. This example is based on data in the 'SURVEY' worksheet, and is re-worked with the new facilities in Figures 5.1 to 5.3. Compare this example with Figure 13.4 in the Introductory Guide, pp 105-6.

Figure 5.1 Use of the INDicator command

```
: open @survey
Rice Survey Data

: indic x5 x11-x13
: indic x5 x14-x16; variate x4
: terms x4 x6 x11-x16
: yvar x6
```

5.2 Regression without the Constant Term

It is now possible to fit a multiple regression model without the constant term. For this purpose, the FIT and REFit commands each have a ;NOConstant subcommand. To restore the constant term to the current regression model, the REFit command has an additional subcommand ;CONstant. Note that DROp and ADD cannot be used to remove or put back the constant term.

The main reason for fitting a regression model without the constant term (or overall mean) is normally to simplify the interpretation of the parameter estimates and their standard errors when fitting models with factors. Typical situations of this kind are the comparison of regressions and the analysis of covariance.

Figure 5.2 is a continuation of the previous example. As in the INSTAT Introductory Guide, we want to fit three parallel regression lines. Columns X11-X13 are the dummy variables for the factor X5, so fitting these terms together with the variate X4 (fertiliser) constitutes the model for three parallel regressions. Note that, without the constant term, the estimates are the three intercepts and the common slope. Thus, this shows directly that the estimated yield with no fertilizer is about 26 cwt/acre units for the 'traditional' varieties, compared with 36 for 'old improved' and 48 for 'new improved'. Each cwt of fertilizer per acre increases the estimated mean yield by 5 cwt/acre. These results can be compared with those in Figure 5.3 where individual lines are fitted to the three groups.

<u>Figure 5.2 Fitting three parallel lines to the yields in the @SURVEY file</u>

```
: fit x11-x13 x4; noconst

ANOVA for regression of Yield
on Fert X11 X12 X13
--------------------------------------
Source        df      SS         MS
--------------------------------------
Regression     4  63506.6     15876.6
Residual      32  732.283     22.8838
--------------------------------------
Total         36  64238.9
--------------------------------------
(No constant term)

: est

REGRESSION COEFFICIENTS

Y-variate: Yield
--------------------------------------
Param.  Estimate   SE          t
--------------------------------------
Fert    5.2643     0.95549     5.51
X11     47.755     3.216       14.85
X12     35.687     2.1168      16.86
X13     25.964     1.436       18.08
--------------------------------------
```

5.3 Detection of Collinearity

The detection of correlated regressor variables has been improved in this release. As the FIT (or ADD) command processes the terms being fitted, it checks the R-squared value that would result from regressing each variable in turn on the variables currently in the model. If this R-squared exceeds 99.99%, then the term is not included. The program actually does more. In principle, it is possible that once a new term is included, one of the terms already in the model might now fail the test for inclusion. The program therefore checks all R-squared values that would result from regressing each term already in on the other terms, including the proposed new one.

From the user's point of view, the major difference is that if a group of terms are being fitted together, the collinearity tests are applied to each variable in turn. Only those terms which fail the test are rejected, and the others are fitted. Previously, the whole lot would have been rejected. To continue with the above example, Figure 5.3 demonstrates the point. Suppose that we now want to fit regression lines with different slopes. It should not be possible to fit all of X14, X15 and X16 while X4 is in the model because X14 + X15 + X16 = X4. Thus when we

: *ADD X14-X16*

the first two terms are fitted, but not the last. Note that the change in regression S.S., and the F-ratio, can still be used to test the significance of the difference in slopes.

The model in Figure 5.3 with terms X11-X13 (intercepts) and X14-X16 (slopes), without the constant term and without X4, has the same residual sum of squares since it is just a reparametrisation of the earlier model, but the estimates and their standard errors are easier to interpret.

Figure 5.3 Further analysis of the rice yields - fitting three separate lines

```
: add x14-x16
  Correlated X-vars: X16 not fitted

Residual S.S.          = 718.024
Residual d.f.          = 30
Increase in Reg.S.S.   = 14.2587
Increase in Reg.d.f.   = 2
F-ratio for change     = 0.298
on (2,30) d.f.

: fit x11-x16; noconst

ANOVA for regression of Yield
on X11 X12 X13 X14 X15 X16
--------------------------------------
Source       df     SS         MS
--------------------------------------
Regression    6  63520.8     10586.8
Residual     30  718.024     23.9341
--------------------------------------
Total        36  64238.9
--------------------------------------

(No constant term)

: est

REGRESSION COEFFICIENTS

Y-variate: Yield
--------------------------------------
Param.  Estimate    SE           t
--------------------------------------
X11     55.1        22.151      2.49
X12     34.194      2.9932     11.42
X13     26.401      1.6141     16.36
X14     2           9.7845      0.20
X15     6.0697      1.483       4.09
X16     4.6938      1.3106      3.58
--------------------------------------
```

5.4 Weighted Regression

The TERms command has a new subcommand ;WEIghts Xm, after which all subsequent regression models will be fitted by weighted least squares, with weights in Xm. Weights must be non-negative. For example,

: *TERMS X11-X16 X6 X4; WEIGHTS X18*

The only way to refit a regression model without weights, or with a new column of weights, is to start afresh from the TERms command.

Zero weights are allowed, and their effect is to delete zero-weighted cases from the regression. This is a useful way of dealing with outliers without having to alter the data in any permanent way. Total and residual degrees of freedom are adjusted for zero-weighted cases. It is also possible to recode cases that are to be removed as missing values (using the RECode command for instance). The effect on the regression is the same as zero-weighting.

5.5 Correlation

The only new feature of CORrelation is the ability to use zero-one weights with the TERms command, as described above, in order to remove cases from the calculations. Note that, in this case, the weights can *only* be zero or one. If TERms was executed with other values in the ;WEIghts column, then weighted correlations are not allowed. However, even if TERms were weighted, the CORrelate command can still be used for two columns, and the result will be the ordinary correlation coefficient. This is because it is only when CORrelate has more than two arguments that the SSP matrix produced by TERms is used.

To summarise,

```
: terms x3-x9
: corr x4 x6 x8 x9
```

is still acceptable.

```
: terms x3-x9; weights x12
: corr x4 x6 x8 x9
```

is acceptable provided x12 consists only of 0's and 1's, and then the zero-weighted cases are left out.

```
: corr x4 x6
```

is always acceptable whether TERms have been declared or not, with or without weights, and if there are weights, they are ignored.

5.6 Missing Values

If there are any missing values in the columns specified in TERms, then they are taken into account in all subsequent regression and correlation calculations (provided MISsing was ON when the TERms command was executed - see Section 7).

An important point to note is that if there is a missing value in a particular row in *any one* of the variables (or in the weights column, if there is one), then that entire row is omitted from all subsequent regression models. This remains true even if the particular column with missing values is not included in the regression model. For example, suppose there is a missing value in row 8 of column X12. Then, after the commands

: *terms x8-x14 x16*
: *fit x8-x10*

row 8 will not be included in the regression calculations, even though X12 is not in the model.

The same applies to correlations, so that

: *corr x8-x10*

would not use row 8. Note, however, that doing the correlations one at a time, which does not require TERms, *would* use row 8, for example

: *corr x8 x9* : *corr x8 x10*

Of course,

: *corr x8 x12*

omits row 8 because the missing value is in column X12.

Finally, a word about fitted values and residuals when there are missing values. If there are missing values in any regressor variable (or in the weights), then the corresponding fitted values and residuals are set to missing values also. However, if there are missing values only in the y-variate, then the fitted value is calculated, though the residual is set to missing.

6: HIGH RESOLUTION PLOTTING

This section is relevant only for users who run INSTAT on a system with sufficient memory to support INSTAT's high resolution graphics.

Some new facilities have been added and there are some changes to the way existing commands work.

6.1 Reference Lines

Two new subcommands, ;HREf and ;VREf, which can be used with both PLOt and REPlot, now make it easy to put horizontal and vertical reference lines on a graph. A plot of residuals, for example, can be easier to interpret if there is a horizontal line through 0. There are many other applications and one example is illustrated in Figure 6.1. The data for this plot are total amounts of rainfall in April over a number of years. Reference lines are drawn at the mean and one S.D. on either side of the mean.

The arguments for ;HREf and ;VREf can be numbers, constants (Kn) and columns (Xm) in any mixture. Thus, for example, all of the following are acceptable:

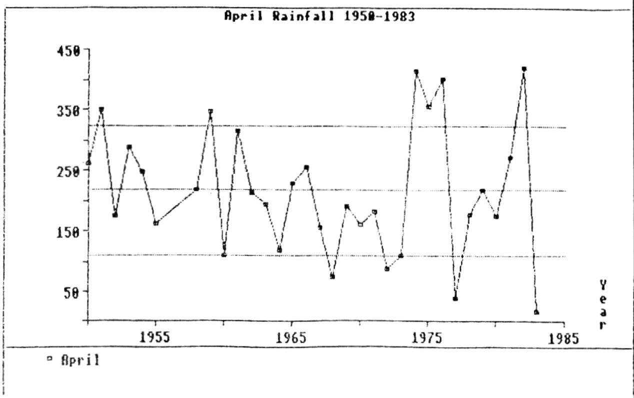
```
: plot x2 x4 x5; href 0 5 10
: plot x2 x4 x5; href k2 k1 x17
: replot; href -10 k2 x17 k1 10
```

etc.

For those who like grids on their graphs, the ;HREf and ;VREf subcommands can be used to draw them.

Figure 6.1 Use of a new plotting subcommand to draw reference lines

```
: k4=mea(x5): k5=k4-sde(x5): k6=k4+sde(x5)
: lin x5 1 1 1: sym x5 1 1 1
: plot x5 x1; href k4-k6; tit "April Rainfall 1950-1983"
```



6.2 Needle Plots

The LINe command, in the first release of INSTAT, was used to produce line plots. Data points could be joined together by straight line segments, with or without symbols. The old usage of the LINe command is still available, but it can now also be used to draw plots of data with a vertical line drawn from each data point to the x-axis. These are sometimes called 'needle plots'. Again, the points may be plotted with or without symbols together with the lines.

To produce needle plots, the first argument of LINe should be set to 2. As before, if this argument is 1, ordinary line plots are produced, and if it is 0 (in which case the second and third arguments are not required), lines are 'switched off'.

The second and third arguments of LINe still operate as before - the second sets the line style, and the third sets the colour. Line style 3 is not useful for needle plots, and is only suitable for function plots.

Figure 6.2 illustrates how the new LINe command can be used for displaying discrete probability distributions. The PRObability command is used to produce the probabilities for two binomial distributions and they are displayed as line graphs on the same plot.

6.3 Default Axes

The DEFault command was intended for setting default x- and/or y-axis ranges which would remain in force for all subsequent plots. However, the command suffered from the drawback of not being able to 'undo' axis range settings and revert to automatic scaling of axes. This problem has been fixed by now allowing the command DEFault without subcommands. This has the required effect.

Another change in the use of DEFault concerns function plots. Previously, a command such as

```
: PLOT S4
```

would not work without an ;XAXis subcommand to specify the x-range. It is now possible to set the x-axis range using DEFault and then subsequent function plots do not need the ;XAXis subcommand. For example,

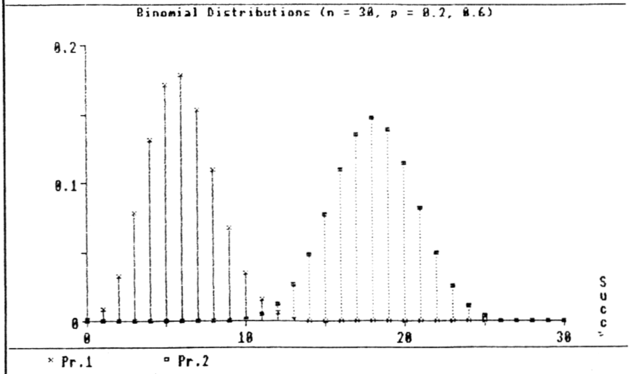
```
: default; xaxis -5 10
: plot s4

: plot s4 s6
```

will now work.

Figure 6.2 Plotting binomial probabilities to illustrate needle plots

```
: ent x15
data  1: (0]30)
data 32: eod
: prob x15 x16; bin 30 0.2
... output suppressed ...
: prob x15 x17; bin 30 0.6
... output suppressed ...
: name x15 'Succ: name x16 'Pr.1 x17 'Pr.2
: line x16 2 1 1: sym x16 5 1 1
: line x17 2 2 1: sym x17 1 1 1
: plot x16 x17 x15
```



6.4 Missing Values

In most cases, if a column contains missing values, they are dealt with in the obvious way when plotting that column. The missing values are simply left out. If the plot is a line plot, the data points on either side of the missing value are *not* joined up by a line segment.

For plots using the ;BY or ;WEIght subcommands, if the factor column contains a missing value, then corresponding data points are not plotted.

7: MISSING VALUES

Most commands in INSTAT now cope with missing values. Each worksheet may have its own missing value codes (up to three are allowed). If a worksheet has no missing values then the flag MISsing can be turned OFF and no checks for missing data are made. This slightly increases the speed with which commands are processed.

7.1 Specifying the missing value codes

The default missing value code is -9999. The CONfigure command can be used to change this default. The missing value codes can also be specified for an individual worksheet, when it is CREated. For example

: *CREate @TEST*; *MISsing* 888 999

creates a worksheet of default size with two distinct missing value codes. There is usually little reason for specifying more than one missing value code as most commands just check whether an observation is missing. The missing value codes can be re-specified when a worksheet is OPEned, for example,

: *OPEn @TEST*; *MISsing* 9999

Note that a change in missing value codes does not change any data that are already in a worksheet. They retain the value of the previous missing value code but will not be treated as missing values. They can, of course, be RECoded if necessary.

Normally only a single missing value code is used. The provision of 3 codes is primarily intended for specialist applications written by users.

7.2 Entry and display of data

When data are input with the REAd, INSert or ENTer commands, missing values are entered either by typing the actual number or by typing * if the first missing value code is to be input (** and *** represent the second or third missing codes). For example

```
: ENTer X1
data 1 : 14  17  *  12  25
data 6 :
```

The RECode command can convert a range of values to missing in the same way, for example,

```
: RECode  X7  X8
data : 201  500  *
```

recodes into X8 data from X7 within the range 201 - 500 as missing, and copies the other data.

Information about which observations are missing is given by the INF command with the subcommand ;MISsing. The DISplay and PREsent commands show missing observations as * (or **, or ***). These are currently the only commands that differentiate between the three missing value codes.

Figure 7.1 gives an example to show some of the basic facilities to specify and enter missing data.

Figure 7.1 Operations with Missing Values

```
: CREate @TEST;MISsing 999  -9999
Enter title for worksheet (or RETURN).
Title : Simple worksheet to demonstrate facilities for missing
        values
: ENTer X1
data 1: 2  6  999  14
data 5: *  **  25
data 8:
:
: DISplay X1
X1
        2        6        *        14        *        **
       25
:
: MISsing OFF
: DISplay X1
X1
        2        6       999       14       999     -9999
       25
: ?M(0)
     999
: MISsing ON
: ?M(0)
     *
:
: ENTer X2
data  1:  (1]7)
data  8:
: RECode X2 X3; DATA 4.5  7  *
:
: ?X3
1        2        3        4        *        *        *
:
: DEScribe X1
Column                  X1

No. of observations     7
No. not missing         4
Minimum                 2
Maximum                 25
Range                   23
Mean                    11.75
Std. deviation          10.145
:
```

7.3 Calculations involving missing values

When a worksheet is used, its first missing value code is stored in a variable called M(0). (Similarly M(1) and M(2) hold the other two codes). The missing value codes can be used in any of the commands that involve calculations. For example

```
: ?M(0)
: SELECT X1 into X5; IF X1<M(0)-1
: X8 = -M(0)*(X7>10)-X7*(X7<=10)
```

The last example recodes all values in X7 to missing if they are greater than 10. (Remember that TRUE is -1 on the BBC Micro, hence the minus sign in this formula.) This calculation could equally be done with the RECode command. We often use the form above, though calculations involving logical expressions take some getting used to.

7.4 Analysis with missing values

Most commands cope with missing values in the obvious way by leaving out the corresponding observations. There are sometimes complications. The easiest way of resolving any doubts about how a particular command handles missing values is to set up a simple worksheet and try the command. For example, when a column is SORted, the missing values are put at the end of the column whatever their actual values.

Sections 5 and 6 of this Guide explain how missing values are handled in PLOtting and in regression. The TABle command is the only one for which we have added an extra subcommand ;MISsing to handle the different situations. Figure 7.2 illustrates the problem. There is no difficulty with the simple use of a TABle to give counts, e.g.

```
: TABle X4X5
```

Here any observation where either X4 or X5 are missing is merely omitted. However the TABle command can also give statistics on one or more columns, e.g.

```
: TABle X4X5;COUnt X7;MEAn of X6 to X8
```

If there are observations missing in X6, the corresponding mean is not given but is coded as missing. The MISsing subcommand, shown in Figure 7.2, caters for the situation where the means are all required and the counts should only be for observations where nothing is missing.

Finally, one or two commands in this upgrade do not yet handle missing values. In particular missing values are not permitted with the ANOVA command. The necessary code for ANOVA is available but is sufficiently extensive as to be impractical on machines without a second processor.

Figure 7.2 Missing values and the TABle command

```
: READ X4-X6
data  1: 1  1  46
data  2: *  1  17
data  3: 2  1  19
data  4: 1  2  24
data  5: 2  1  14
data  6: 2  2  20
data  7: 1  1   *
data  8: 1  1  35
data  9:
:
: FACtor  X4 2 X5 2
: TABle  X4 X5

   X4     X5     COUNT
  LEVEL  LEVEL
    1      1       3
    1      2       1
    2      1       2
    2      2       1

     1 observation(s) with missing levels
: NOTE  Missing factor in X4(2)
:
: TABLE X4 X5 ;COUNT X7 ;MEA X6 X8

   X4     X5     COUNT    MEAN
  LEVEL  LEVEL             X6
                  (X7)    (X8)
    1      1       3        *
    1      2       1       24
    2      1       2      16.5
    2      2       1       20

     1 observation(s) with missing level(s)
: NOTe  Mean for level (1,1) given as missing because
: NOTe  one of the 3 observations at that level is missing
: REMove X8X7
: TABle X4 X5 ;COUnt X7 ;MEAn X6 X8 ;MISsing X6

   X4     X5     COUNT    MEAN
  LEVEL  LEVEL             X6
                  (X7)    (X8)
    1      1       2      40.5
    1      2       1       24
    2      1       2      16.5
    2      2       1       20

     1 observation(s) with missing level(s)
plus      1 observation(s) missing from X6
: NOTe With ;MIS X6 the count for level (1,1) is reduced
: NOTe to 2 and mean is from these 2 values
```

8: MISCELLANEOUS

There are a number of improvements that do not warrant a complete section in this guide.

8.1 Coded tables

A facility for CODed tables has been added to the PREsent command. Figure 8.1 gives a simple example. Here the residuals in a 2 way ANOVA have been PREsented with all values between -1 and +1 CODed to ".". This focuses attention on the values that are outside this range.

Figure 8.1 Example of coded tables

```
: OPEN @EXPERI : YVA X1 : FAC X2 4 X3 3
: ANO X2 X3 ; RESiduals X4
: PREsent X4 ;ROW X2 ;COL X3 ; FIX 2 ; CODe -1 1 "."

   Table for X4: Treat BY Block

  Treat         1       2       3
  ------------------------------
  Block ----------------------
      1 :       .       .       .
      2 :       .       .       .
      3 :       .       .    1.32
      4 :    1.05       .   -1.36
```

The ;CODe subcommand can be used repeatedly, for example

```
: PREsent X1; ROW X2; COL X3; CODe 0 20 "low";
              CODe 20.1 50 "medium"; CODe 50.1 100 "high"
```

8.2 The OSCli command

A new OSCli command has been added to pass star commands to the operating system. As in BBC Basic, the command

```
: OSC "CAT"
```

is the same as *CAT. However it is possible to type more than one command on a line with the OSCli command, for example,

```
: OSC "CAT" :INF
```

whereas a star command passes the whole remainder of the line to the operating system. This means that the OSCli command can be part of a macro. We required this facility to produce a batch of plots on the printer. The following macro, in S1, was used to produce and dump 20 plots of X2 v X1, X3 v X1 etc.

```
: ENT S1
S1: PLO X%1 X1 : OSC "DEPSON" : %1 = %1+1
: %1 = 2
: USE S1; REPeat 20
```

8.3 Data Management with the COPy Command

The basic syntax of the new COPy command is

: COPy Xn1 Xn2 ... into Xm1 Xm2 ...

and, used like this without subcommands, simply makes copies of the columns Xn1, Xn2, ... in new columns Xm1, Xm2, ...

COPy therefore seems to offer a mild advantage over the alternative sequence of commands

: Xm1 = Xn1 : Xm2 = Xn2 : ... etc

in that there is perhaps a little less typing to do.

The real power of the command, however, lies in its one and only subcommand ;MATch Xs Xt. Here, Xs and Xt are factor (or interaction) columns. The effect of this is to copy values from Xn1, Xn2, ... only when Xt matches Xs. Missing values are put into the 'target' columns Xm1, Xm2, ... when there is no match.

To see how this facility might be useful, consider the following simple example, illustrated in Figure 8.2. A survey of households includes some variables pertaining to the household itself, and others concerned with individuals within households. (There can be different numbers of individuals in each household.) For instance, data at the household level could include 'type of house', 'rented or owner-occupied', etc., while data on individuals could be age, sex, education, etc. The data resulting from this survey is thus an example of a hierarchical data structure with two levels.

Suppose we have a factor column X3 which simply indexes the households, and, to keep things simple, suppose there are just 6 households. Suppose also that the total number of individuals is 20, and that X4 is a factor column of length 20 and with 6 levels, indicating the household to which each individual belongs.

Columns X1 and X2 contain data at the household level, and are therefore of length six. If we are to analyse the data at the level of individuals, it may be necessary to have copies of the household data (from X1 and X2) for each individual. This is the purpose of COPy with the ;MATch subcommand, and the result from the command

: *COPY X1 X2 into* X5 X6; *MATch* X3 X4

is shown in Figure 8.2. Columns X5 and X6 contain copies of the household data from X1 and X2, respectively, each individual having his appropriate household data.
If X4 had values other than 1 to 6, then missing value codes would have been inserted in those rows of X5 and X6. INSTAT's new facilities for missing values are dealt with in Section 7 of this Supplementary Guide.

Figure 8.2 Using the COPy command with hierarchical data

```
: factor x3 6 : factor x4 6 : display x1-x3

Row        X1        X2        X3

  1        1.7       21.4       1
  2        2.3       16.6       3
  3        1.9       10.7       2
  4        2.2       13.7       5
  5        2.8       18.4       4
  6        2.1       17.2       6
: display x4

X4
        3       2       4       4       6       1
        1       5       3       2       1       5
        6       6       2       5       5       1
        1       3
:
:
: copy x1 x2 into x5 x6; match x3 x4
: display x1-x6; wid 6

Row      X1      X2      X3      X4      X5      X6
  1      1.7     21.4    1       3       2.3     16.6
  2      2.3     16.6    3       2       1.9     10.7
  3      1.9     10.7    2       4       2.8     18.4
  4      2.2     13.7    5       4       2.8     18.4
  5      2.8     18.44   4       6       2.1     17.2
  6      2.1     17.2    6       1       1.7     21.4
  7                              1       1.7     21.4
  8                              5       2.2     13.7
  9                              3       2.3     16.6
 10                              2       1.9     10.7
 11                              1       1.7     21.4
 12                              5       2.2     13.7
 13                              6       2.1     17.2
 14                              6       2.1     17.2
 15                              2       1.9     10.7
 16                              5       2.2     13.7
 17                              5       2.2     13.7
 18                              1       1.7     21.4
 19                              1       1.7     21.4
 20                              3       2.3     16.6
```

The example shows how COPy with ;MATch can be used to 'expand' columns of data from a higher level (households) to a lower level (individuals). The reverse operation of 'collapsing' or summarising data from the 'individual' level to the 'household' level is also often needed. This sort of operation is precisely what the STAtistics command does. To continue the same example, if X9 contains data on individuals age, then the mean age per household and the age of the youngest person can be saved in X10 and X11 by

: *STAts X9; BY X4; MEANS X9 X10; MIN X9 X11*

The STAtistics command gives a wide variety of summary measures that are useful in different circumstances.

In some studies, hierarchical data arise simply because results are available more frequently for one group of variables than for another. For example, when analysing climatic data, rainfall values may be available on a daily basis while only monthly averages are available for temperature and evaporation. Suppose column X1 contains 365 daily rainfall amounts, while X2 contains just the 12 values of the average daily evaporation in each month. Then, after setting up suitable factor columns, the STAtistics command could be used to give the rainfall totals for each of the 12 months, while the COPy command could expand the evaporation column, X2, to give a value for each day.

8.4 Changes in the GAMma and STAtistics commands

A ;REStrict subcommand has been added to the GAMma, and STAtistics command, and it is also in the ROW command. It enables data values outside the specified range to be omitted. For example, if a gamma distribution is to be fitted to a column of rainfall data, X1, the data must all be greater than zero. If there are a few zeros then a simple way to fit a gamma distribution to the non-zero amounts is by

: *GAMma X1; REStrict 0.01*

This omits all values less than 0.01 from the analysis. Similarly

: *GAMma X1; REStrict 0.01 998*

omits all values outside the range 0.01 to 998.

The STAtistics command has another new facility that was also required when processing climatic data. The syntax

: *STAtistics X1-X10; BY X11; SUM to X21-X30*

is now permitted. This is quite powerful. If X1 to X10 are of length 366 containing 10 years of daily rainfall records and X11 is a factor column with 12 levels, which takes the value 1 for the first 31 values (January) etc., then the command above produces the 10 years of monthly totals in X21 to X30.

8.5 The output of ASCII files

It is sometimes useful to output data from a worksheet so that they can be transferred to another package or computer. The OUTput command is available for this task. It is mentioned here because, although unchanged in this update, it is not covered in the Introductory Guide, though it is fully specified in the Reference Guide on page 59.

The simplest use of the OUTput command is just

:OUTput xi....xj

This outputs the data from each column in turn to the file TEMP. This file can then be edited with a word processor. To check its contents, type

*TYPE TEMP

either within or outside INSTAT.

The OUTput command has a number of subcommands. Figure 8.3 gives two examples of its use. In the first, data from the @SURVEY file are output by case to a named file. Note that strings S1, S2.... can also be OUTput. In Figure 8.3 this is used merely to give descriptive information, but could also give the appropriate control information so the resulting file can be read directly into another package. The second example uses INSTAT's facilities to generate data from a specified distribution. These data could form the basis for a teaching practical where pupils each take a different subset of the data. For this they need to be on suitable slips of paper or card. This type of formatting is easy with Basic and the resulting file from INSTAT is in a form to become the data lines in such a program.

It is also sometimes useful to save results in a file so they can be edited and printed later, if necessary. This is done with the star command, *SPOOL. It is used as follows:

```
: *SPOOL filename          this opens the file on the disc to
:                          receive the output.
:
: INSTAT commands          the screen output is now copied to
:                          this file.
:
: *SPOOL                   this closes the spool file.
```

The filename does not need the @ in front because *SPOOL is an operating system rather than an INSTAT command. The resulting file can now be listed with

*TYPE filename

and can be edited with any word processor. We have trapped some of the screen display that would be inconvenient in a spooled file, for example the output from a DISplay command will appear without any of the screen prompts to press the shift key for a new page.

Figure 8.3 Examples of the OUTput command

```
: OPEN @ SURVEY
  Rice Survey Data
:
: ENTER S1
S1 : NOTE Cols are village no., size of field, fertilizer,
                                  variety and yeild
:
: OUTput S1 X1 X3 - X6; CASe; FILe @SURVOUT
:
:
 NOTe Cols. are village no., size of field, fertilizer,
                              variety and yield
      1        2      2.5      2      53.6
      1        5      1.5      2      44.6
      1        .       .       .       .
      1        .       .       .       .
      2       1.5      2       2      40.4         etc.
: CLOSe
: NOTe - Now the second example.
: CRE @TEACH; COL5 100; STR 5
 Title :Demonstration of output of generated data

: GENerate 100 X1; NOR 100 15
  120.6263 101.2284 94.4574 5061 94.6556 83.7395 102.2136
  113.4953 122.7536 94.2903 83.43 55 85.3468 101.1711
  86.2071 4.2694 95.8926 78.8182 109.7130 ......

: GENe 100 X2; NOR  0  5
  4.1035 5.1484 -0.5092 -7.6110 6404 -1.8990 -3.6090 7.6023
  0.4455 -1.6921 -3.2990 -11.4515 6.8700 20 4.9826 4.1139
  -7.1287 -2.9423 ......

: X3 = 40 + 0.5* (X1 - 100) + X2

: ENTER S1 S2
  S1 : REM Column 1 is the mark of the child in a test
  S2 : REM Column 2 is their IQ

: OUTput S1 S2 X3 X1 ; CASe ; FIX 0 ; BASic 1000
  1000 REM Column 1 is the mark of the child in a test
  1010 REM Column 2 is their IQ
  1020DATA   54.,  121.
  1030DATA   46.,  101.
  1040DATA   37.,   94.
  1050DATA   28.,   91.
  1060DATA   54.,  113.
  1070DATA   35.,   84.
  1080DATA   54.,  114.
  1090DATA   48.,  117.
      .        .      .
      .        .      .
      .        .      .
```

8.6 The IMPort and EXPort Commands

The IMPort and EXPort commands have been written so data can be transferred between an INSTAT worksheet and either of two commonly used database packages for the BBC Micro, Masterfile II and QUEST.

The full documentation for these commands is in Appendix 2 of the INSTAT Schools Guide and Chapter 14 of the INSTAT Programmers Guide.

8.7 Other changes

There are numerous other minor changes, some trap errors more effectively, others give more meaningful error messages. The speed of some of the commands, particularly PREsent and SCAtterplot has been improved. Last, but certainly not least, the HELp facilities have been rewritten and are considerably extended. They also display in either 40 or 80 columns depending on the current screen MODe.

INSTAT SUPPLEMENTARY GUIDE

PART II REFERENCE

INTRODUCTION

The Reference Section of the Supplementary Guide updates the current version of the INSTAT User Guide, Part II.

In the following pages, those commands which are new to this version have their own new reference pages. Where a command in the old version has undergone substantial modification, for instance by the addition of new subcommands, then the corrected pages are given here. A note of the commands that have changed could be made in your Part II User Guide.

One or two minor changes, errors and omissions have been thought to be too small to merit a whole new page. For instance, according to the old Reference Guide, the OUTput command seems to have a ;FORmat subcommand. There is no such subcommand. On the other hand, the RECode command has a ;DATa subcommand which is not mentioned in the current Guide. Its usage and effect are the same as the ;DATa subcommand for REAd. With the REAd command, data may extend over several lines if the continuation symbol, &, is the last character on any incomplete data line.

There is an error in the description of the PRObabilities command in the Reference Guide. The probabilities given for continuous distributions are, by default, the 'upper tail' probabilities and not the cumulative distribution function, as stated. A new subcommand ;LESs has been added, whose effect is to produce the 'lower tail' probabilities instead.

Another relatively minor addition which has not earned new pages in the Reference Guide is the ;REStrict subcommand. This subcommand has been added to the STAtistics and GAMma commands. It is also available with the new ROW command, and its usage is described under that command. Its effect is to truncate the data so that calculations are based on a restricted range of values.

CHIsquare New Command

Syntax:

: CHIsquare Xn
: CHIsquare Xn1 Xn2 ...

Subcommands:

; COLumns Xm
; CONtinuity correction
; COUnts Xm
; EXPected Xm n
; FACtors Xn1 Xn2
; FValues Xm
; GROup n1 n2
; RESiduals Xm
; ROWs Xm

Purpose

Calculates chi-square statistics either for contingency tables or for goodness-of-fit tests.

If the counts of an r by c contingency table are arranged in columns Xn1, Xn2, ... Xnc (where the columns are each of length r), then : CHIsquare Xn1 Xn2 ... Xnc gives the chi-square test for independence. With this usage of the command, the subcommand ;COUnts Xm can be used to save the original counts in a single column Xm. Similarly, the subcommands ;ROWs Xm1 and ;COLumns Xm2 will create 'factor' columns to index rows and columns associated with the ;COUnts column. The counts column Xm is then a 'table' column (just as though the TABle command had been used with subcommands ;COUnts Xm ;ASSociated Xm1 Xm2).

For a 2 by 2 table, the subcommand ;CONtinuity uses the usual continuity correction.

Alternatively, the counts of the contingency table may be stored in a single column Xn to begin with, the rows and columns being indexed by factor columns Xn1 and Xn2, respectively. The chi-square test is then given by : CHIsquare Xn; FACtors Xn1 Xn2.

If the counts column was produced by the TABle command, then the ;FACtors subcommand is redundant. The chi-square test is obtained directly. For example, if the counts were produced by

: *TABle X1 X3; COUnts X8*

then the chi-square test is given by : *CHIsquare* X8.
Expected, or 'fitted' counts are saved in Xm1 by the subcommand ;FVAlues Xm1. For a contingency table, these are the expected values of the cell entries assuming the null hypothesis of independence to be true.

Standardised residuals are saved in Xm2 by ;RESiduals Xm2. The residuals are defined by

St. resid. = (Observed - Expected) / SQR(Expected).

To use the CHIsquare command for a goodness-of-fit test, a column, Xm, containing expected frequencies must first be calculated, possibly by means of the FREquencies command. If the observed frequencies are in Xn (which must have the same length as Xm), then the command is : CHIsquare Xn; EXPected Xm p, where p is the number of parameters that have been estimated from the data in Xn in order to derive the expected frequencies. In the event that there are small expected frequencies, adjacent rows may be combined by using the ;GROup n1 n2 subcommand. Rows between n1 and n2 (inclusive) would then be combined.

Remarks

The output of the CHIsquare command for a contingency table shows the observed and expected frequencies in the form of a one-way table. The PREsent command could be used to display the data, fitted values and residuals in tabular form.

For goodness-of-fit tests, a check is made of the sums of the observed and expected frequencies. If they are not equal, the user is asked whether or not to continue.

Examples

Contingency tables:

```
: CHI X3-X6
: CHIsq X3-X6; COUnts X8; ROWs X11; COLs X12

: table x2 x4; counts x12; assoc x15 x16
: chisq x12; fvals x20; resids x21
: present x12 ; rows x15; cols x16
```

Goodness-of-fit tests:

```
: chi x10; exp x12 1
: CHI X14; EXP X10 1; GRO 1 2; GRO 10 13
```

CONfigure — Changed Command

Syntax: : CONfigure

Subcommands: none

Purpose

Allows the alteration of the default settings for the initial configuration of INSTAT.

The command provides a menu of options which should be self-explanatory.

COPy New Command

Syntax: : COPy Xn1 Xn2 ... Xm1 Xm2 ...

Subcommands: ; MATch Xs Xt

Purpose

Without the subcommand, columns Xn1, Xn2, ... are simply copied into columns Xm1, Xm2, ..., respectively.

The number of columns Xm1, Xm2, ... must be equal to the number of columns Xn1, Xn2, ...

The subcommand ;MATch Xs Xt has the effect of copying those rows from Xn1, Xn2, ... for which Xs and Xt are equal, into Xm1, Xm2,... Columns Xs and Xt must be factor (or interaction) columns for this to work. If a value in Xt does not occur in Xs, missing values are inserted in Xm1, Xm2, ...

The ;MATch facility is intended mainly for hierarchical data structures, such as might occur with data on, for example, 'households' and on 'persons' within households. Then Xn1, Xn2, ... would have data at the household level, Xs would index households and Xt (of length equal to the total number of individuals) would indicate the household to which each individual belongs. Columns Xm1, Xm2, ... would then have the same length as Xt and contain the appropriate household data copied for each individual.

Remarks

Note that while the effect of ;MATch is to 'expand' data at one level to longer columns at a lower level in a hierarchical data structure, the reverse effect can be achieved with the STAtistics command. This can be used to 'collapse' data at the lower level into a column of means (or some other statistic), one for each value of the higher level.

Examples

: *COPy X4-X6 into X7-X9*
: *copy x1 x3 x5 x2 x4 x6*

: *FACtor X6 4 X6 4*
: *COPy X1 X2 into X3 X4; MATch X6 X7*

DEFault | Changed Command

Syntax : DEFault

Subcommands: ; XAXis x1 x2
; YAXis y1 y2

Purpose

Sets or erases the default ranges for the x-axis and/or y-axis when using the PLOt or REPlot command.

DEFault without subcommands erases previous axis range settings. The effect of the DEFault command will be overridden by using the ;XAXis or ;YAXis subcommands with PLOt or REPlot.

Remarks

This command is only relevant for systems with enough memory to support high resolution graphics.

The actual limits used for the axes may not be exactly the same as those specified by the subcommands. This happens because INSTAT tries to find 'nice' numbers for the limits. The actual limits will always include the range specified.

Examples

: *DEF; XAX 0 25*
: *default; yaxis -3 to 3*
: *def*

<u>DICtionary</u> <u>New Command</u>

<u>Syntax:</u> : DICtionary

<u>Subcommands:</u> none

<u>Purpose</u>

This command is intended for programmers who wish to change INSTAT by adding their own commands. It is also possible to remove commands. It should NOT be used for any other purpose.

DICtionary should be used with care. It is described in more detail in the Programming Guide. Incorrect use of the command can result in unpredictable behaviour of the system.

EXPort New Command

Syntax: : EXPort @database filename

Subcommands:

; MASterfile
; QUEst

; COLumns Xn1 Xn2 ...
; COMposition
; DRIve n
; FIElds n1 n2
; FROm n
; RECode Ln1 Ln2
; UNTil n2

Purpose

EXPorts data to either the database Masterfile II or the database QUEST from the current worksheet. This command is documented in the Programming Guide and the Schools Guide.

FIT — Changed Command

Syntax:
: FIT
: FIT Xn1 Xn2 ...

Subcommands:
; FVAlues Xm
; RESiduals Xm
; NOConstant

Purpose

Fits a multiple regression model of the current y-variate on the specified columns as explanatory (regressor) variables.

The y-variate must have been previously declared using the YVAriate command, and all the columns specified in the FIT command, together with the y-variate, should first be declared in TERms.

If the ;WEIghts Xm subcommand was used with TERms, the regression calculations will be based on weighted least squares, with weights in Xm.

The FIT command without arguments fits a constant (the mean of the y-variate) and displays the (corrected) total sum of squares of the y-variate with its degrees of freedom.

If explanatory variables are specified with the command, then the output consists of the overall analysis of variance table for the regression, the overall F-ratio and the value of the 'coefficient of determination', R-squared.

The subcommand ;FVAlues Xm saves the fitted values in Xm, and ;RESiduals Xm saves the residuals in Xm.

Normally, the fitted regression model will include a constant term, but a model without the constant can be fitted by using the ;NOConstant subcommand.

After a regression model has been fitted, the following 'system scalars' are saved in memory and may be used in CALculations, etc:

TSS = total (corrected) sum of squares of y-variate
ESS = residual sum of squares
FSS = regression sum of squares
TDF% = total degrees of freedom
EDF% = residual degrees of freedom
FDF% = regression degrees of freedom

Remarks

The parameter estimates (or 'regression coefficients') and their standard errors, etc, can be displayed with the ESTimates command. Terms may be added to or dropped from the regression model by means of the ADD and DROp commands. This does not apply to the constant term, which can be added or removed only by using the REFit command.

The regression model is 'remembered' by the worksheet, so information on the regression can be recalled in a subsequent INSTAT session. This does not apply to the 'system scalars', which are available only during the current session.

Examples

: *terms x1-x5 x8 x9* : *yvar x8*
: *fit x2 x4-x6; fvals x22; resids x23*
: *FIT*
: ADD X6 X2

: *fit x2-x12; noconst*

GROup — New Command

Syntax: : GROup Xn (Xm)

Subcommands:
; BOUndaries Xn1 (or x1 x2 ... xp)
; PERcentages
; VALues Xn2 (or y1 y2 ... yp)

Purpose

Forms a frequency distribution of data in Xn and (optionally) saves the frequencies in Xm.

Without either of the subcommands ;BOUndaries or ;VALues, the command simply counts the occurrences of each value in Xn.

A grouped frequency distribution may be obtained by specifying the interval endpoints with the ;BOUndaries subcommand. The endpoints x1, x2, ..., xp may be given directly or may be read from a column Xn1. The frequencies correspond to intervals x<=x1, x1<x<=x2, ..., x>xp.

As an alternative to grouping into intervals, the ;VALues subcommand can be used to produce frequencies of occurrences of values y1, y2, ..., yp (which may also be read from a column Xn2).

Percentage relative frequencies will be displayed (but not saved) by using the ;PERcentages subcommand.

Remarks

The output column, Xm, of the GROup command is compatible with the output from the FREquencies command, which produces expected frequencies from various distributions. This should facilitate chi-square goodness-of-fit tests with the CHIsquare command.

Examples

: *GROup X3 X9; BOUndaries 50 100 150 200 300*
: *group x4; values x11; percents*

IMPort New Command

Syntax: : IMPort @database filename

Subcommands:

; MASterfile
; QUEst

; COLumns Xn1 Xn2 ...
; COMposition
; DRIve n
; FIElds n1 n2
; FROm n
; RECode Ln1 Ln2
; UNTil n2

Purpose

IMPorts data to the current worksheet from either the database Masterfile II or the database QUEST. This command is documented in the Programming Guide and the Schools Guide.

INDicator Changed Command

Syntax: : INDicator Xn Xm1 Xm2 ...

Subcommands: ; VARiate Xm

Purpose

Generates indicator (or 'dummy') variables Xm1, Xm2, ... from a factor column Xn.

The indicator variables will have the same length as the factor Xn, and there can be one for each factor level. The values in an indicator column Xmi are 1 for cases (rows) where the factor Xn has the value i, and 0 otherwise.

The effect of the ;VARiate Xm subcommand is to copy values of the column Xm into the indicator columns instead of ones. Thus Xmi will contain values of Xm when the factor Xn has the value i, with zeros elsewhere. This subcommand is intended to facilitate the construction of the columns required for regression involving the interaction of a factor with a variate, as in testing for parallelism in the comparison of regressions.

Examples

If X13 is a factor with 4 levels,
: *indic x13 into x1-x4* will generate indicator columns x1-x4

: *IND X13 X5 X6* generates indicator columns for the first two factor levels only

: *IND X13 X5-X8; VARiate X10*

LINe

Changed Command

Syntax:

: LINe Xm n1 n2 n3
: LINe Sm n1 n2 n3
: LINe Xm 0

Subcommands: none

Purpose

Used in high resolution plotting to define the type of line, the line style and the colour for plots of data and functions.

After executing the command LINe Xm n1 n2 n3, subsequent PLOts of Xm (as 'y-variable') will be line plots. LINe Sm n1 n2 n3 sets the line style and colour for a function plot.

OPTIONS FOR n1:

For plotting the data in a column Xm, if n1 is 1, consecutive data points will be joined by straight line segments. If n1 is 2, the points will be plotted with vertical lines drawn from them to the x-axis.

For plotting a function in a string Sm, n1 will normally be 1, which will produce a curve, although n1 = 2 will also work.

The command LINe Xm 0 'switches off' line plotting for a column Xm, so that plots of Xm will be symbol plots.

OPTIONS FOR n2:

The number n2 can be 1, 2 or 3 and defines the line style. This will be a solid line for n2 = 1, or a broken line for n2 = 2.

The option n2 = 3 is only useful for plotting functions and defines a dotted line.

OPTIONS FOR n3:

The colour of the line is set by n3. The colours are:

n3 = 1: yellow
n3 = 2: green
n3 = 3: white

Remarks

The command LINe Xm n1 n2 n3 will suppress symbols in plots of Xm, but it is possible to have both lines and symbols by using the SYMbol command either before or after the LINe command.

Examples

: *LIN X4 1 2 2*
: *LIN X4 2 1 3*
: *line x6 0*
: *LINE S8 1 2 1*

MISsing — New Command

Syntax: : MISsing (ON or OFF)

Subcommands: none

Purpose

Switches ON or OFF INSTAT's recognition of missing values.

With MISsing OFF, data values coded as missing will be given the numeric value corresponding the the missing value code. Otherwise, those commands that can cope with missing values will deal with them in the appropriate way.

Many commands will execute significantly faster with MISsing set to OFF, so if the data are known to contain no missing values MISsing should be OFF.

Remarks

The words 'ON' and 'OFF' with the MISsing command are optional.

The default setting for MISsing is normally ON.

OSCli New Command

Syntax: : OSCli "DFS command"

Subcommands: none

Purpose

Passes the DFS command to the operating system so that it may be executed.

Remarks

The effect of OSC is identical to the direct execution of the DFS command as a *-command. For instance,

: OSC "CAT" has the same effect as : *CAT

OSCli is mainly intended for use in macros, where a *-command cannot otherwise be used.

Example

The following code creates and runs a macro that dumps five successive plots to the printer, using the *DEPSON utility.

```
: ENTer S1
S1: Plot x1 x%1 : OSC "DEPSON" : %1=%1+1
: %1=2
: USE S1; REPeat 5
```

PLOt — Changed Command

Syntax:

: PLOt Xn1 Xn2 ... Xm
: PLOt Sm1 Sm2 ... Xm
: PLOt Sm1 Sm2 ...; XAX x1 x2
: PLOt Xn1 Xn2 ... Sm1 Sm2 ... Xm

Subcommands:

; BY Xn
; TITle "text" or Sn
; WEIghts Xn
; XAXis x1 x2
; YAXis y1 y2
; HREf n1 ... Km1 ... Xp1 ...
; VREf n1 ... Km1 ... Xp1 ...

Purpose

Displays a high resolution plot of data and/or functions.

The command in the form PLOt Xn1 Xn2 ... Xm plots Xn1, Xn2, ... simultaneously against Xm. Plots of data can be symbols, lines or both. See the LINe and SYMbol commands.

To plot a function, a BASIC expression (in upper or lower case) representing the function must first be saved in a string variable, using the ENTer command. Only functions of a single variable can be plotted, and the variable must appear as x or X in the expression. The expression defining the function may contain stored constants Km1, Km2, ...,or the integer variabes %1, %2 ... but no other data types.

Normally, INSTAT automatically calculates a suitable range of values over which to plot the graph, but this can be overridden by using the ;XAXis x1 x2 and/or the ;YAXis y1 y2 subcommands to specify the x- and y-ranges, respectively. The DEFault command (q.v.) can also be used to set the x- and y-ranges. In this case, use of the ;XAXis and ;YAXis subcommands with PLOt temporarily overrides the ranges set by DEFault.

The command PLOt Sm1 Sm2 ...; XAXis x1 x2 plots several functions over the x-range x1 to x2, as specified in the XAXis subcommand. Alternatively, the x-range may previously be specified by using the DEFault command.

PLOt Sm1 Sm2 ... Xm plots the functions Sm1, Sm2, ... over the x-range determined by the range of values of the data in Xm.

Data and functions may be plotted together on the same graph, as in the last form of the command syntax above. The columns and the string variables may be mixed in any order, with the restriction that the last argument must be the column representing the x-values.

If Xn is a factor, the subcommand ;BY Xn plots data corresponding to different factor levels with different symbols. In this case, only one column may be plotted as 'y-variable', although any number of functions can be plotted at the same time.

Similarly, if Xn is a factor, the ;WEIghts Xn subcommand has the effect of plotting the data with the size of the symbols proportional to the elements of Xn.

The graph may be given a title with the ;TITle subcommand. The title may either be supplied with the command, or it may be previously saved in a string variable Sn.

Horizontal and vertical reference lines may be put on the graph by means of the ;HREf and ;VREf subcommands. The arguments of each of these can be numbers, stored constants or columns in any mixture. Reference lines are drawn as broken lines.

Remarks

The PLOt command is only available on systems with enough memory to support INSTAT's high resolution graphics.

The plot will be displayed in screen mode 1, unless the current mode is 0, in which case the plot is also in mode 0. In mode 0, plots have higher resolution, but colours are not available.

When in mode 0 the command *DEPSON can be used to dump the plot to an EPSON compatible printer.

The command REPlot can be used to repeat the last PLOt with modifications in the subcommands.

Examples

```
: PLO X4 X1
: PLOT X3 X5 X2 AGAINST X11; HREF 0 K3 X9; VREF K4 0
: plot x4 s1-s3 x1; by x6

: enter s1
S1: sin(x)/x
: enter s2
S2: (1-cos(x))/x
: plot s1 s2; xax -20 20

: PLO X3 X4 S6 S8 X5; TITLE "Data and Fitted Curves"; XAX 0 20
: plot x3 vs x1; by x8; title s12
```

PREsent Changed Command

Syntax: : PREsent Xn

Subcommands:

; ACCuracy n
; COLumns Xm1, Xm2, ...
; FIX n
; HEAders
; MARgins
; PERcentages
or
; PERcentages Xm
; PRInterwidth n
; ROWs Xm1, Xm2, ...
; WIDth m
; CODe x1 x2 "text"

Purpose

Displays the data in Xn in the form of a multi-way table. Normally, the data in Xn will be a column of counts, or the values of a statistic generated by the TABle command.

If Xn was generated by the TABle command, and if the ;ASSociated subcommand was used with TABle to produce indexing factor columns, then these columns will automatically be used to index the table displayed by PREsent Xn.

The layout of the table can be controlled by using the ;ROWs and/or ;COLumns subcommands. For example, ;COLumns Xm1 Xm2 ... has the effect of indexing the columns of the table by the levels of the *factors* Xm1, Xm2, ..., each factor being 'nested' within its predecessor.

The ;MARgins subcommand adds row and column marginal totals to a table of counts, provided the table is not too large.

A table of counts can be displayed as a table of percentages by using the ;PERcentage subcommand. If used without Xm, each table element is a percentage of the overall total. Used with Xm (which must be a row or column of the table), percentages are given according to the totals per factor level of Xm.

;FIX n displays the data in the table rounded to n decimal places, and ;ACCuracy n to n significant digits. The ;WIDth m subcommand puts each entry in the table in a field width of m characters, and can be used together with ;ACCuracy or ;FIX to obtain a suitable display.

The ;PRInterwidth n subcommand is useful when sending the output of PREsent to a printer. The output can be widened to n characters, which may be more than will fit on the screen.

Values that lie in a specified range, say between x1 and x2, can be displayed as a text code by using the subcommand ;CODe x1 x2 "text".

The subcommand ;HEAders causes each row/column in the table to have its full set of heading labels.

Remark

The columns Xn in PREsent Xn can in fact be *any* column, and not necessarily one generated by the TABle command. The factors used for indexing the table are then specified by the ;COLumns and ;ROWs subcommands. The entries in the table are totals of data in Xn corresponding to each cell.

Examples

```
: PRE X5; COL X11; ROW X8 X9
: pres x6; col x8; row x9; code -50 0 "Low"
:
: table x4-x6; counts x7; means x1 x8; assoc x9-x11
: present x7; col x11; row x10 x9; mar
: pre x8; col x10; row x11 x9; wid 12; fix 4; pri 120
```

REFit — Changed Command

Syntax: : REFit

Subcommands:
; FVAlues Xn1
; RESiduals Xn2
; CONstant
; NOConstant

Purpose

Displays the overall analysis of variance table for the current regression model (if any).

The subcommands enable fitted values and/or residuals to be saved *after* fitting the model using the FIT, ADD and DROp commands. The REFit command can also be used to add or remove the constant term.

Note that the output from the ADD and DROp commands does not show the analysis of variance table for the new regression model, and the REFit command can be used for this purpose.

If the current regression has no constant term, then it can be added by using the ;CONstant subcommand. Similarly, use the ;NOConstant subcommand to drop the constant term from the current model.

Remarks

Without subcommands, the REFit command actually does nothing to the worksheet. It simply displays the analysis of variance table.

The residuals saved by the RESiduals subcommand are in 'unit normal deviate form' - i.e. the differences between observed and fitted values divided by the estimated residual standard deviation.

Example

```
: terms x1-x6: yvar x6
: fit x1 x3
: add x4
: refit
: drop x1
: ref; fvals x11; resids x12
: ref; noconst
```

REPlot — Changed Command

Syntax: : REPlot

Subcommands:
; BY Xn
; TITle "text" or Sn
; WEIght Xn
; XAXis x1 x2
; YAXis y1 y2
; HREf n1 ... Km1 ... Xp1 ...
; VREf n1 ... Km1 ... Xp1 ...

Purpose

Repeats the last graph produced by the PLOt command. The point is to enable modifications to be made to the graph by means of the subcommands, without having to repeat the entire PLOt command line. Changes to the graph can also be made using the SYMbol, LINe, DEFault and MODe commands, followed by REPlot, with or without subcommands.

The ;XAXis and ;YAXis subcommands can be used to 'zoom in' on an interesting portion of a graph, or to extend the range of the axes. Most often, the ;XAXis subcommand alone is enough. INSTAT will then automatically choose the y-axis range.

The subcommands are identical to those of the PLOt command (q.v.).

Remarks

The REPlot command is only available on systems with enough memory to support INSTAT's high resolution graphics.

Examples

```
: PLOT X2 X3 against X9
: SYM X2 1 2 1: LIN X3 1 1 2
: REPLOT

: plot x5 x6 s2 s4 x1; title "Data and Fitted Curves"
: replot; xax 0 20
: replot; href 2.5 7.5 k3; vref 0
```

ROW

New Command

Syntax: : ROW Xn1 Xn2 ...

Subcommands:

; COUnt Xm
; MEAn Xm
; MINimum Xm
; MAXimum Xm
; SDEviation Xm
; SUM Xm
; REStrict x1 (x2)

Purpose

Calculates row statistics for the rows of columns Xn1, Xn2, ...

The statistic is specified by one of the subcommands (except for ;REStrict which is not a statistics subcommand). The result is saved in column Xm, given as the argument of the statistics subcommand.

There must be at least two columns Xn1, Xn2, ... and they must all have the same length.

At least one statistics subcommand is necessary for this command to work.

The ;REStrict subcommand has been included to enable data values outside a given range to be omitted before calculating the statistics. Its effect is as follows: if only one argument, x1, is given, the calculations will be restricted to values >= x1; if two values, x1 and x2, are specified, only values between x1 and x2 (inclusive) will be used. In the event that there are insufficient values remaining in a row to calculate the statistic, then a missing value will be placed in the corresponding row of the 'target' column Xm.

Examples

: *ROW X2-X11; MIN X16; MAX X17*
: *row x3 x6 x9-x13; mean x21; sdev x22; restrict 1 99.9*

TABle — Changed Command

Syntax: : TABle Xn1 Xn2 ...

Subcommands:

; ASSociated Xm1 Xm2 ...
; COUnt Xm
; CSSquares Xn (Xm)
; MAXimum Xn (Xm)
; MEAn Xn (Xm)
; MINimum Xn (Xm)
; SDEviation Xn (Xm)
; SUM Xn (Xm)
; USSquares Xn (Xm)
; MISsing Xn

Purpose

Displays, and optionally saves, a table produced by the classifying factors Xn1, Xn2, ... The result is displayed in columns and not in tabular form. To produce a rectangular display, first save the tabulated data using the TABle command, and then use the PREsent command to display it.

The arguments Xn1, Xn2, ... must be factor columns of equal length. Without subcommands, TABle Xn1 Xn2 ... displays the counts of number of cases in each cell of the table. The counts may be saved in Xm by appending the ;COUnt Xm subcommand. The column Xm becomes a column of type 'table', as shown by the INFo command.

The subcommand ;ASSociated Xm1 Xm2 ... creates new factor columns Xm1, Xm2, ... of length equal to the number of cells in the table. Their main purpose is for indexing the cells for the PREsent command. The number of new factor columns specified in the ASSociated subcommand must equal the number of original factor columns Xn1, Xn2, ...

With the exception of ;COUnt and ;ASSociated, the subcommands are for displaying various statistics of the data in a column Xn, classified according to the table specified by the factors Xn1, Xn2, ... The column Xn must have the same length as Xn1, Xn2, ... For saving the values of the statistic, an optional second column Xm can be given after Xn. The column Xm will become a column of type 'statistic'.

Whenever the values of a statistic are saved, the ;COUnts subcommand must also be used to store the cell counts. The counts column is then locked by the system. (The reason for this is that if the PREsent command is to be subsequently used for displaying the values of a statistic, then it requires the counts.)

The statistics subcommands ;MAXimum, ;MINimum, ;MEAn, ;SDEviation, ;SUM should be self-explanatory. ;CSSquares and ;USSquares are the corrected and uncorrected sums of squares, respectively.

The treatment of missing values in tabulations needs some care. Normally (provided MISsing is ON), any missing value in one of the input factor columns, simply causes the corresponding case to be ignored. However, misleading output could result if there are missing values both in the factor columns and in an input variate column required for the calcuation of a statistic. Then the counts relate to the non-missing frequencies in the factor columns only, and any further missing value in a variate column will result in the corresponding summary statistic being coded as missing. An alternative is to use the ;MISsing Xn subcommand. This causes all missing cases in Xn to be omitted. If only one variate column is used, the column Xn in the ;MISsing subcommand will be the same as that specified in the accompanying statistics subcommand. See the last example below.

Remarks

If statistics columns are saved, then the counts column can only be unlocked by REMoving the statistics columns.

Impossible values, such as a cell mean with a zero cell count, are coded as missing values.

When counts and statistics are saved, some 'pointers' are set up in the worksheet which associate some columns with others. The column of counts 'points to' the generating factor columns, and the statistics columns 'point to' the count column.

Examples

: *tab x5 x6*
: *tab x4; cou x8*
: *TABLE X4-X6; MEAN X1*
: *table x3 x5 x6; counts x7; assoc x8-x10*
: *TAB x4-x6; MEA x1; MAX x2 x8; COU x7*
: *tab x1 x2; mean x4; miss x4*

TERms

Changed Command

Syntax: TERms Xn1 Xn2 ...

Subcommands: ; WEIghts Xm

Purpose

Computes the 'sum of squares and products' (SSP) matrix for the columns Xn1, Xn2, ... and saves it in the worksheet.

The TERms command must be used before fitting regression models involving the variables Xn1, Xn2, ... The list of columns must include the response variable (declared by the YVAriate command).
The columns should all have the same length and no column should be constant.

The TERms command is also necessary before using the CORrelation command with more than two arguments.

If weighted regression analysis is required, the weights must be declared with the ;WEIghts Xm subcommand. The column Xm must have the same length as Xn1, Xn2, ... and must contain only non-negative values. Zero weights have the effect of deleting the corresponding rows of Xn1, Xn2, ... from all regression analysis.

If the CORrelation command is subsequently used with more than two arguments, then the only permissible weights are zero and one, and the effect of zero weights is the same as in regression.

The SSP matrix is referred to as V1 in commands such as REMove or LOCk. A worksheet cannot contain more than one SSP matrix.

Remarks

When creating the worksheet, ensure that if regression or correlation calculations will be required, sufficient space is reserved for the SSP matrix by using the ;SSP subcommand with CREate.

The columns Xn1, Xn2, ... (and the weights column Xm) will be 'system locked' after executing the TERms command. They can subsequently be unlocked by removing the SSP matrix using the command REMove V1, or by overwriting it with a new SSP matrix.

Examples

```
: TERMS X6 X2-X4
: CORR X4 X6 X3

: terms x2-x12; weights x16
: yvar x7
: fit x2 x11
```

INDEX

APPENDIX I

ALPHABETICAL INDEX TO COMMANDS for PART I USER GUIDES

INTRODUCTORY (I), SUPPLEMENTARY (S-I) AND SCHOOLS (SG) GUIDES

Command	Page I	Page S-I	Page SG	Subcommands	Example
ADD	100	13+	-	;FVA;RES	:ADD X1 X4
AGAin	-	-	-	--	:AGAin
ANOva	79	22	-	;ERR;FVA;MEA;RES	:ANO X1X6;FVA X7
BOXplot	54	-	-	;BY ;LIN;NOS;NOT:PRI ;SUM	:BOX X1-X3;NOT
CALc	45	2	13,28	;UNI	:X4=X1+K1*2.32
CHIsq	-	7+	24+	;COL;CON;COU;EXP;FAC ;FVA;GRO;RES;ROW	:CHI X3-X6
CLEar	127	-	34	--	:CLEar
CLOse	126	29	-	--	:CLOse
CONfig	9	-	-	--	:CON 4
CORr	94	15+	32	--	:COR X4 X7 X1 X2
COPy	-	25+	-	;MAT	:COPy X4 X6 X5 X7
CREate	28	20	11	;COL;CON;LAB;MIS;NOT ;SSP;STR;TIT	:CRE@DRY;COL 20 50
DEFault	-	18	-	;XAX;YAX	:DEF;XAX 1 20
DELete	42	-	13	--	:DEL 3 X1-X3
DESc	66	21	23,26	;ALL;COE;CSS;IQU;KUR ;LQU;MED;PER;SKE;STE ;UQU;USS	:DES X1;PER 10 90
DIC	-	-	-	--	:DIC
DISplay	34	5+	12	;ACC;COL;FIX;FRO;LAB ;PRI;TO ;WID	:DIS X1-X3;FRO 5
DROp	101	-	-	;FVA;RES	:DRO X4
ECHo	9	-	-	--	:ECHo OFF
ENTer	33	20	12	;DAT	:ENT X4:ENT K1
ERRor	-	-	-	--	:ERRor ON
EST	95	12,14	33	;COR;COV	:EST X5;COV X6
EXPort	-	30	48	;MAS;QUE;COL;COM;DRI ;FIE;FRO;REC;UNT	:EXP@DBASE1;QUE
FACtor	42	8,26	-	--	:FAC X3 4
FIT	95	12,14	33	;FVA;NOC;RES	:FIT X4 X1;RES X8
FREq	116	10	26	;BIN;CHI;FDI;GAM;NOR ;POI;TDI	:FRE 80;BIN 15 .5
GAMma	120	27	27	;MAX;MOM;RES	:GAM X1;MOM
GEN	122	29	12	;BER;BIN;EXP;GEO;NOR ;PER;POI;SAM;SEE;UNI	:GEN 8 X1;BIN 5 .4
GROup	-	4+	25	;BOU;PER;VAL	:GRO X3X9;BOU X1
HEAding	9	-	38	--	:HEA
HELp	25	30	7	--	:HEL FULL
HISt	50	-	17,25	;FRE;MID;NOF;NUM;TIT ;WID;XAX	:HIS X3;FRE X7
IMPort	-	30	46	;MAS;QUE;COL;COM;DRI ;FIE;FRO;REC;UNT	:IMP@DATA1;MAS
INDic	104	11	-	;VAR	:IND X1 X2-X4
INFo	30	-	12	;ALL;COL;CON;FRE;LAB ;MAC;MIS;SSP;STR	:INF :INF;COL :INF @TEST
INPut	36	-	15	;INT;TRA	:INP@TV X4;INTX1
INSert	39	-	13	;DAT	:INS 3 X1-X3
INTer	82	-	-	--	:INT X2 X4 X5
KEY	10	-	-	--	:KEY2 "OPE@TS:INF"
LINe	60	18+	34,41	--	:LIN X3 1 2 2
LOCk	43	-	-	--	:LOC X1-X3
MACro	-	-	-	--	:MAC USE S1
MENu	127	-	-	--	:MEN
MISs	-	20+	-	--	:MIS

Command	Page I	Page S-I	Page SG	Subcommands	Example
MODe	59	-	34	--	:MOD 0
NAMe	35	19	9	--	:NAM X1 'Yield
NORmal	97	-	-	--	:NOR X2 X4
NOTe	40	23	-	--	:NOTE example 1
ONEway	82	-	-	;COU;FVA;MEA;RES;SE	:ONE X3
OPEn	29	20	13	;MIS;NOM;TIT	:OPE @TEST
OSCli	-	24	-	--	:OSC"DRIVE 1"
OUTput	-	27+	-	;ACC;BAS;CAS;FIL;FIX ;PRI;VAR;WID	:OUT X1-X9;FIL@DD1
PAGe	126	-	39	--	:PAGe ON
PAUse	126	-	34	--	:PAU 8
PERc	113	-	31	;CHI;FDI;GAM;NOR;TDI	:PER X1X6;FDI 3 10
PLOt	58	17+	18	;BY ;TIT;WEI;XAX;YAX ;HRE;VRE	:PLO X2X4;XAX 0 5
PREsent	73	9,24	-	;ACC;COL;FIX;HEA;MAR ;COD;PER;PRI;ROW;WID	:PREX5;COLX6;ROWX7
PROb	110	7,19	31	;BIN;CHI;FDI;GAM;NOR ;POI;TDI;LES	:PRO 5;NOR 3 1
QUIt	12	-	-	--	:QUI
RANk	95	-	32	--	:RAN X2 X12
REAd	34	23	11	;DAT	:REA X1-X4
RECode	46	20	30	;DAT	:REC X1-X3
REFit	100	-	-	;FVA;RES;CON;NOC	:REF
REMove	42	23	33	--	:REM X1-X3 :REM V1
REPlot	59	17	8,41	;BY ;TIT;WEI;XAX;YAX ;HRE;VRE	:REP;XAX 2 4
ROW	-	2+	-	;COU;MEA;MIN;MAX;SDE ;SUM;RES	:ROW X2-X11;MINX12
SCAtter	56	-	18	;HEI;LET;WID	:SCA X4;WID50;LET
SELect	48	22	29	;IF	:SEL X1X2;IF X1>25
SHOw (?)	42	6,22	29	;UNI	:? K3
SORt	-	-	7	--	:SOR X1 X2 X4 X5
STAt	70	27	41	;BY ;COU;DEV;FIT;FOR ;MAX;MEA;MED;MIN;PER ;PRO;SDE;SUM;RES	:STA X2-X4;MEA X7 :STA X2;BY X3
STEm	52	-	18	;HIS;NOT;PRI	:STE X2-X5
SYMbol	60	17	41	--	:SYM X3 3 1 2
TABle	75	22+	-	;ASS;COU;CSS;MAX;MEA ;MIN;SDE;SUM;USS;MIS	:TAB X1X2;MEA X4X5
TERms	93	11,14	32	;WEI	:TER X1-X4
TINt	119	-	23	;CON;TES	:TIN X1;TES 150
TITle	-	-	-	--	:TIT "Rain Data"
UNLock	43	-	-	--	:UNL X2 X6
USE	131	24	39	;IF ;REP;WHI	:USE S4;REP 2
VDU	39	-	13	--	:VDU 2
WARn	9	-	34	--	:WAR
YVAr	94	11	-	--	:YVA X4
%1-%9	130	-	-	--	
*CAT	125	-	-	--	:*CAT :*.
*DRIVE	125	-	14	--	:*DRIVE 1
*DEPSON	143	-	37	--	:PLO X3X4:*DEPSON
*DNEC	143	-	-	--	:PLO X1X2:*DNEC
*DTEXT	143	-	-	--	:DIS X3-X8:*DTEXT
*EXEC	125	-	14,38	--	:*EXEC EGS1
*KEY	125	-	-	--	:*KEY VDU2¦M
*SPOOL	-	28	37	--	:*SPOOL OUT1
*TYPE	-	28	-	--	:*TYPE EGS2

APPENDIX II

INDEX TO COMMANDS FOR EACH TOPIC

REFERENCE (II) and SUPPLEMENTARY REFERENCE (S-II) GUIDES

APPENDIX II Continued

INDEX TO COMMANDS FOR EACH TOPIC

REFERENCE (II) and SUPPLEMENTARY REFERENCE (S-II) GUIDES