Introductory Guide

instat

a statistics package for the BBC micro

Statistical Services Centre

UNIVERSITY of READING

INSTAT

Introductory Guide

by

R.W.Burn R.D.Stern J.Knock

Revised Version March 1986

CONTENTS

Chapter 1: INTRODUCTION

INSTAT is a general purpose statistics package for the BBC microcomputer. It is a system for interactive data analysis and will be equally useful in both statistics teaching and research in any discipline which requires the statistical analysis of data.

INSTAT uses a powerful command language with commands entered at the keyboard (or from a file) to process and analyse data. An on-line help facility and diagnostic error messages make it easy to use and to learn. Many commands have optional sub-commands that add great flexibility to the command language. These features combine to give INSTAT the immediacy and versatility of many mainframe statistics packages, as opposed to the more conventional menu-driven packages for micros.

The difficulties imposed by the limited memory of the BBC micro have been overcome in two ways. First, data are stored in a special disc file called a 'worksheet'. The worksheet is updated after each command is executed, which has the added bonus of providing security for data. Second, the program has a modular structure with part of the program in memory all the time, while program modules are automatically overlaid as they are required.

INSTAT has been written by experienced statisticians engaged in teaching, research and consultancy, in collaboration with professional programmers. It began as a series of programs for teaching statistics written by Bob Burn, presently working as a consultant in Mauritius. Development of the package has continued since late 1983, in collaboration with staff from the University of Reading and the University of Colombo, Sri Lanka, under the guidance of Roger Stern. Preliminary versions of INSTAT have already been extensively used and tested in courses on statistics in agriculture and health given both at Reading and in Colombo.

INSTAT's manual is in two parts. This is Part I, the Introductory Guide to INSTAT while Part II, in a separate book, is the Reference Manual. The Introductory Guide shows you how to get INSTAT going and, by means of a number of examples, how to manipulate and analyse your data. It is not intended to be a handbook of statistical methods. What we have set out to do in the Introductory Guide is simply to help you to learn how to apply statistical methods using INSTAT as a tool. Of course, as practising statisticians ourselves, we have our own ideas as to how the subject should be practised and taught, and you will find instances where we have been unable to resist the temptation of putting our point of view.

Chapter 2 discusses the installation and testing of INSTAT. The installation section can be ignored once you have a satisfactory working version of the program.

Remaining chapters in the Introductory Guide describe most of the facilities available in INSTAT and discuss the different stages in processing data. The data sets described in Chapter 3 are supplied as worksheets on the INSTAT disc, so you should be able to repeat many of the examples used in later chapters.

Two files are supplied on the disc to illustrate possible analyses. Once in INSTAT, type

: **EXEC EGS1* or : **EXEC EGS2*

The commands in EGS1 illustrate a complete 'run' for a simple set of data. The data are entered, summarised and plotted. This takes about 3 minutes. In contrast, EGS2 takes about 11 minutes to run and uses six different data files to illustrate a range of analyses and plots. These files can be run on any system, but the illustrations of high resolution plotting in the examples will not work on the ordinary BBC without either additional screen memory or a 6502 Second Processor.

The Reference Manual, Part II, formally describes the INSTAT commands and gives details of their syntax. Most of this information is also in the HELP files on the disc (see Chapter 5 for a description of INSTAT's on-line HELP facilities).

A convention that has been adopted in both parts of the manual is that all INSTAT commands that you enter from the keyboard are in *italics*, while output produced by INSTAT appears in print as it appears on the screen. Also, since INSTAT can accept commands in either upper or lower case, we have used a mixture of both in the examples. Occasionally, a carriage return appears in the examples, and this is denoted either *<RETURN>* or *<RET>*.

Work on the further development of INSTAT continues. We would also value your comments and suggestions. Additional documentation will include a Programmers' Manual, to enable users who know BBC BASIC to write their own program modules with new commands for INSTAT. Educational guides are also planned to illustrate ways in which INSTAT can be used in teaching.

ACKNOWLEDGEMENTS

Many people have contributed to INSTAT, either by writing modules, testing the package or helping with the documentation.

We would like to particularly mention:

Nihal Kodikara, S.T. Nandasara, Kevin Seneviratne and other staff from the University of Colombo, Sri Lanka.

Philip Swandale, Richard Coe, Alison Ansell and staff of the Department of Applied Statistics and the Statistical Services Centre at the University of Reading.

Don Mather, of Brighton, who wrote the code for the EPROM.

Margaret Lamb of the Department of Typography, University of Reading, who designed the cover for the User Guide and packaging.

Thanks are due to the participants and staff on the 'Statistics in Agriculture' course in Colombo, and the courses in 'Management and Analysis of Statistical Data' and 'Statistical Methods in Agricultural Climatology' at Reading, who helped us by using INSTAT.

Finally, we would like to thank Rosemary Stern and Barry Knock in Reading and Nalini Burn in Mauritius, for their sacrifices and for enduring the inconveniences and sleepless nights caused by the 'birth' of INSTAT.

Bob Burn
Roger Stern
Joan Knock

Mauritius and Reading

March 1986

Chapter 2: INSTALLING AND TESTING INSTAT

2.1 INTRODUCTION

The INSTAT package consists of a 16K EPROM, an 80 track double sided disc and two manuals. The installation and testing of INSTAT is described in three stages:

1) Installing the standard system and producing working discs.
2) Trying out the standard system.
3) Adapting INSTAT.

We assume you are familiar with the BBC microcomputer. If not, Appendix 4 gives an example of a practical sheet to help introduce the novice user to the BBC microcomputer.

2.2 INSTALLING THE STANDARD SYSTEM

Insert the EPROM inside the BBC microcomputer, following the instructions given in Appendix 3.

Appendix 2 gives some information about the EPROM and details of its two screen dumps.

2.3 PRODUCING WORKING DISCS

For safety, we recommend that two additional copies of the INSTAT disc are made on 80 track double sided discs. Remember to copy both sides of the INSTAT disc. Label one disc with the sticker provided and use this disc as the master. Keep the original INSTAT disc as a backup for this master disc. The second disc can then be tailored to your requirements, as described in Section 2.5, and used as a working disc. You can make several working discs from your master.

The INSTAT double sided disc contains the following files:

Side 2	Side 0		
INSTAT	INSTEXT	@.EXPERI	@.REGRESS
M.INSTAT	EGS1	@.SREG1	@.PLUM
INS0 - INS19	EGS2	@.SURVEY	@.CHISQ
ERR	TXT1A	@.RAIN	@.NIATEMP
	TXT1B	@.RAIN10	HPTR1

Side 2 contains the program files and ERR, a file of error messages.

Side 0 has the HELp files (TXT1A, TXT1B and HPTR1), a range of sample data sets (all files beginning @.) and two EXEC files (EGS1 and EGS2). Initially, use your copy of the master disc, with the sample data sets, to introduce INSTAT to yourself or to others. This illustrates many of the facilities in INSTAT.

The files on your working discs will depend on whether you use a single or a dual disc drive. On a single double-sided drive, Side 2 would normally be copied from the master, while on Side 0 you would only need the help files, TXT1A, TXT1B and HPTR1. If you are not using a 6502 Second Processor, the file INSTAT can be deleted from Side 2, making room for the three help files and leaving Side 0 free for the data files.

On a dual drive system, the simplest is to use working discs as copied from the master and use the second drive, i.e. Sides 1 and 3, for your data files.

Appendix 1 gives information on how to organise files if you have less than a single 80 track double-sided disc drive.

2.4 TRYING OUT INSTAT

INSTAT runs on the BBC, BBC+ and the BBC with a 6502 Second Processor. It is possible to use the Aries and other expansion boards to expand the memory of the standard BBC for plotting etc..

Steps applicable to all the systems

1) If you have one, turn off the Second Processor.

2) While holding down the <*CTRL*> key, press the <*BREAK*> key. Among other things, the screen should display

   INSTAT R-A00001

   The number R-A00001 is the serial number of your EPROM (your number will not be this very one), and this message shows the EPROM is installed correctly.

3) Insert your copy of the INSTAT disc.

4) Type **INSTAT* or **IN.*

Your screen should look like this:

```
INSTAT

A STATISTICAL PACKAGE FOR THE BBC

BBC/BBC+ VERSION

Copyright Statistical Services Centre

University of Reading 1985

Serial Number R-A00001
```

5) The next screen, Figure 2.1, displays an 'Introductory Menu'

Figure 2.1 Introductory Menu

```
Introductory Menu      March 1986

 1) Use INSTAT

 ** HELp about INSTAT          **
 2) Overview of INSTAT
 3) Different sections in INSTAT
 4) Help on the options in this menu
 5) Display users instructions stored
    in INSTEXT file

 ** Sample analyses            **
 6) Text of an example
 7) Run the example shown in 6)

 ** Tailoring INSTAT           **
 8) Change starting options in INSTAT
 9) Change standard worksheet size etc
10) Change initial contents of fn keys

11) Stop

  Enter 1,2,...11 and RETURN
```

From this menu, Option 7 runs the demonstration file EGS1, which shows some INSTAT commands and tests the disc and EPROM. There is no need to press any key or the space bar during this run. Note that without a Second Processor or separate screen memory, the PLOt and REPlot commands will not be processed.

At the end of this run, the 'Introductory Menu' is again displayed. Option 11 leaves the package, while Option 1 shows a

screen with something like this at the top of it:

```
INSTAT   ERR :On   War :On   Ech  :On
WS:  Mem  : nnn

: _
```

The first line shows the status of the Error, Warning and Echo flags (see Section 2.5 for their meaning and how to change them). WS (worksheet) is blank now, but the name of the current WS is inserted, whenever a worksheet is being processed. How to load and create a worksheet are explained in Chapter 6. The number nnn indicates how much memory is left, at any stage of the INSTAT run.

The : _ is the INSTAT prompt, and it shows you that the program is ready to receive commands. If you wish to analyse a simple set of data for yourself, try the example given in Figure 3.1. Alternatively, if you wish to run the demonstration example again, type **EXEC EGS1* Type *MENu* to return to the menu, *QUIt* (or just press the <*BREAK*> key) to leave the program. If you have separate screen memory or a 6502 Second Processor, you can type **EXEC EGS2* to give a longer demonstration of INSTAT.

Additional steps for systems with a 6502 Second Processor

6) Switch on the 6502 Second Processor and then press the <*CTRL*> and <*BREAK*> keys simultaneously.

INSTAT requires Hi-BASIC with the 6502 Second Processor. If this is not the default, type **FX 142,n*, where n is a number between 0 and 15 and refers to the position of the EPROM containing Hi-BASIC. If Hi-BASIC is not available on EPROM, it can be put on the INSTAT disc, following the instructions in the 6502 Second Processor USER GUIDE, Chapter 6.

7) Type **INSTAT* and the screen should look like this:

```
INSTAT

A STATISTICAL PACKAGE FOR THE BBC

6502 2ND PROCESSOR VERSION

Copyright  Statistical Services Centre

University of Reading  1985

Serial number R-A00001
```

If instead, the following message appears, load Hi-BASIC as in 6)

"PLEASE LOAD HI-BASIC AND THEN PRESS
THE BREAK KEY BEFORE TRYING AGAIN"

8) When the 'Introductory Menu' is shown, type *7* to rerun the EGS1 file.

9) Use Option 1 from the menu. You should get the system prompt : _ on the screen. Type **EXEC EGS2*. This example file gives a more comprehensive demonstration and test of INSTAT. It takes about 11 minutes to run.

NOTES

1) Before you type *INSTAT, always press the <*BREAK*> key.

2) Within INSTAT the <*ESCAPE*> key normally terminates the command being executed.

3) Pressing the <*BREAK*> key from within INSTAT is not recommended while a command is being executed, because the contents of the current worksheet may be corrupted.

2.5 ADAPTING INSTAT

This section first illustrates how a !BOOT file can be created, so that INSTAT is loaded when <*SHIFT*><*BREAK*> is pressed. Then we show how some of the starting up options can be altered, in particular, how to avoid the 'Introductory Menu' and go instead straight into INSTAT. Finally, we show how you can program the function keys, f0-f9, to your own requirements, so that their contents are automatically loaded every time INSTAT is run.

1) If INSTAT is to be 'adapted', always use a working disc created by you, as instructed in Section 2.3.

2) If you wish to activate INSTAT automatically, you should construct the appropriate !BOOT file for this disc on Side 0. The BBC's *BUILD and *OPT commands can be used.

```
e.g.   *BUILD !BOOT
       0001  *FX 142,2     (assumes 2nd processor with
       0002  *INSTAT        Hi-BASIC in rom position 2)
       0003  <ESCAPE>
       *OPT 4 3
```

Now pressing the <*SHIFT*> and <*BREAK*> keys together should load INSTAT.

3) Menus are provided in only two places in INSTAT. The first is the 'Introductory Menu' given in Figure 2.1. The CONfigure command can be used to reconfigure INSTAT, so that this menu is

bypassed. You will rarely need to reconfigure the package, so menus are also provided here.

The CONfigure command can only be used when there is no Second Processor switched on.

a) Type **INSTAT* and from the 'Introductory Menu'

either i) Press Option 1 to enter INSTAT and type : *CON 5*,
or ii) Use Option 8 from the menu.

The screen should show a menu similar to Figure 2.2. Type *7* and 'Yes' should change to 'No'. Now, after pressing *<RETURN>* and typing **INSTAT* to reload the package, the 'Introductory Menu' will not be displayed and you will go directly to the INSTAT prompt : _

Figure 2.2 Menu to reconfigure the starting options

```
        Starting options

 1) Main drive for program files          2
 2) Secondary drive (if any)              0

    Take care before altering options 3-5
 3) Version number                        2
 4) Overlay size (bytes)               8192
 5) Min. space if ws. in memory        2500

 6) Users instruction shown              No
 7) Initial menu displayed              Yes
 8) Modules not available
 9) Other commands not available

    Which is to be changed?
Enter  1,2,..9, or RETURN if none

    Press <RETURN> to leave this menu.
```

b) Normally, the flags for WARnings, ECHo and HEAdings, etc. are initially 'ON' for safety. These are special INSTAT commands which control certain aspects of INSTAT's output. WARnings are given when, for instance, you enter a command which would overwrite existing data. ECHo controls whether or not sequences of stored commands are displayed on the screen when they are executed. HEAding switches 'OFF' or 'ON' the heading at the top of the screen. This may be useful when you want to send the screen contents to a printer.

Experienced users may wish to change some of these to 'OFF' by typing

: *CONfigure 4*

and changing the new startup settings for the flags.

These new settings are permanent and apply every time INSTAT is used, until changed again using *CONfigure 4*. They can however, be changed temporarily during any INSTAT session, by typing

: *WARn OFF* : *ECHo OFF* : *HEAding OFF*

c) You may wish to change some of the other options in the package. Note that the CONfigure command changes the data stored in the file INSDATA. Hence, to return to the original configuration of INSTAT, the file INSDATA has only to be copied from the master disc (see Section 2.3).

5) Programming the Function Keys:

KEY is another command that alters the file INSDATA. This may be used from Option 10 in the 'Introductory Menu' or by entering the *KEY* command during an INSTAT session.

The initial contents of the function keys are as follows:

f0	---	
f1	AGA|M	Display last commmand typed.
f2	*DEPSON|M	Dump screen plot to Epson printer
f3	INF;ALL|M	Give all infomation on open WS
f4	*CAT|M	List files on current disc
f5	*DRIVE 1|M	Change to disc drive 1
f6	VDU3|M	Turn OFF printer
f7	VDU2|M	Turn ON printer
f8	*DTEXT|M	Dump screen of text to printer
f9	---	

Unless changed with the *KEY* command, the function keys will be loaded with these contents every time INSTAT is run. Within any run, the contents of a function key can always be changed with the BBC's *KEY command, as described in the BBC User Guide, page 141.

Chapter 3: DATA SETS

3.1 INTRODUCTION

Before we look in detail at how data can be manipulated and analysed in INSTAT, it is useful to consider the form of various typical data sets and the types of data they contain.

Many simple sets of data come in the form of 'rectangles'. The rows of data correspond to cases, perhaps the people that have answered a questionnaire. The columns correspond to the variables or the questions that they have answered. As in most other statistical packages, many of INSTAT's commands are instructions to the program to act on 'columns' of data. For example, if X1 and X2 are columns of data of the same length, the command

```
: PLOT X1 X2
```

instructs the program to draw a plot of the data in X1 against the corresponding values in X2. The command

```
: describe x1
```

is an instruction to present simple descriptive statistics for the data in column X1.

In this chapter Figure 3.1 gives an example of the commands typed in a complete session of INSTAT. Details of how to use these commands are given in later chapters and in the Reference Manual.

This figure shows the commands to set up a worksheet, followed by the entry, display and processing of a set of data. It also illustrates how straightforward it is to use the package for simple tasks.

In the remaining sections of the chapter, four examples of data are presented which are used in many later chapters in this Introductory Guide. Ways are suggested in which the data can be organised into a form suitable for the computer. Worksheet files containing the example data sets are on the master disc, so that you may try out the analyses given in this guide.

Figure 3.1 Commands to illustrate the use of INSTAT

Commands	What they do	Description in Reference Guide Page No.
: *CRE @TEST* Title : *Test WS*	Create a blank worksheet file on the disc. INSTAT asks for a title for the worksheet.	14
: *REAd X1 X2 X3* data 1: *10 15 20* data 2: *12 14 19* data 3: *22 11 26* data 4: *41 9 22* data 5: *EOD*	Read 3 columns of data into the worksheet TEST. (Numbers may be separated by one or more spaces, or by commas)	72
: *DISplay X1-X3*	Look at the data on the screen	20
: *INFo*	Look at information about the worksheet.	42
: *SCAtter X1 X2*	A simple scatterplot of X1 v X2	77
: *DEScribe X1 X3*	Display summary statistics for X1 and X3	18
: *TINterval X1;CON 99*	Give the 99% confidence interval for X1.	89
: *QUIt*	Leave INSTAT	70

3.2 EXAMPLE 1: A REGRESSION EXAMPLE

File on Master Disc: @.REGRESS

Figure 3.2 shows a small set of data, consisting of just 10 observations (or 'rows' or 'cases') and two variables (or 'columns'). The data are from a study in Bangladesh on the cotton yield for a variety sown on 10 planting dates spaced at two week intervals from 1st September 1973. The dates of planting are coded with 1st September 1973 as day 1, and the yields are in tens of kilograms per hectare. These data are ready for entry as they stand. The only minor decision the user may wish to make is whether the two variables should be given names in order to clarify the output.

Figure 3.2 Cotton Yields against Planting Date

Day number	Cotton Yield
1	17.39
16	17.74
31	16.02
46	14.94
61	13.88
76	9.78
91	7.38
106	6.09
121	4.26
136	3.92

3.3 EXAMPLE 2: A SURVEY

File on Master Disc: @.SURVEY

Figure 3.3a presents the data from a small survey on the relationship between rice yield and cultivation practice. The data are fictitious but the model used to generate them is based on the results of a regular survey conducted in Sri Lanka. The data matrix consists of the following six variables:

1: Name of village
2: Field number within the village
3: Size of field (acres)
4: Quantity of fertilizer applied (cwt/acre)
5: Variety (New Improved, Old Improved, Traditional)
6: Yield (cwt/acre)

Some statistics packages can accept the data exactly as given in Figure 3.3a. However, the main data matrix in INSTAT consists only of numbers. The first and fifth columns, therefore, have to be rewritten with numeric codes for villages and varieties. Figure 3.3b shows the result. It is convenient to code the four villages as 1, 2, 3 and 4 and the three types of variety as 1, 2 and 3, although other numeric codes could be used.

In fact, although we have to assign numeric codes to villages and varieties to get the data into INSTAT, it is possible to give labels to the codes so that they appear in INSTAT's output. This is done by putting them into a label column and Chapter 7 explains how to do it. In the meantime, we note the label columns corresponding to the village and variety codes in Figure 3.3c.

Figure 3.3 Survey Data on Rice Yield

Fig. 3.3a Data as given

Village	Field	Size	Fert.	Variety	Yield
Sabey	3	2	2.5	Old	53.6
Sabey	13	5	1.5	Old	44.6
Sabey	4	5	3	Old	50.7
Sabey	20	1.5	1	Trad	33.6
Sabey	19	5	2.5	New	62.1
Sabey	10	4	1.5	Trad	30.6
Sabey	8	4.5	2	Old	37.7
Sabey	11	3.5	0	Trad	24.3
Sabey	14	5	2	New	56.8
Sabey	12	4.5	2.5	Old	59.3
Kesen	9	1.5	2	Old	40.4
Kesen	8	7	0	Trad	25.8
Kesen	1	2.5	1.5	Old	40.7
Kesen	4	8	0	Trad	27.6
Kesen	6	4.5	2.5	Old	48.7
Kesen	3	2	0.5	Trad	27
Kesen	5	4	0	Trad	19.1
Niko	2	4.5	0	Trad	26.3
Niko	8	2.5	0	Trad	24.7
Niko	7	6	0.5	Old	40.4
Niko	3	6	0	Old	31.8
Niko	4	8	0	Trad	29.6
Nanda	4	4.5	2	Trad	36.6
Nanda	2	3.5	2.5	Old	57.4
Nanda	13	20	3	Trad	42.7
Nanda	21	4	3	Old	49.3
Nanda	12	3	1	Old	46.2
Nanda	24	6	1.5	Old	42.2
Nanda	26	4	2	Old	41.3
Nanda	8	3	0	Trad	37.6
Nanda	20	2.5	2.5	New	58.1
Nanda	14	2.5	1.5	Old	45.8
Nanda	28	2	2	Trad	38.7
Nanda	25	4	2	Old	42.4
Nanda	5	4.5	1.5	Trad	25.8
Nanda	9	1.5	2	New	61.4

Fig. 3.3b Data with Village and Variety Codes

Village	Field	Size	Fert.	Variety	Yield
1	3	2	2.5	2	53.6
1	13	5	1.5	2	44.6
1	4	5	3	2	50.7
1	20	1.5	1	3	33.6
1	19	5	2.5	1	62.1
1	10	4	1.5	3	30.6
1	8	4.5	2	2	37.7
.	.	.	.	.	.
.	.	.	.	.	.

Fig. 3.3c Village Names and Variety Types

Village code	Village
1	Sabey
2	Kesen
3	Niko
4	Nanda

Variety code	Variety
1	New
2	Old
3	Trad

3.4 EXAMPLE 3: AN EXPERIMENT

File on Master Disc: @.EXPERI

Figure 3.4a shows the data from a simple randomised block experiment, with 4 blocks and 3 treatments, in the form that is suitable if the data are to be analysed 'by hand'. For entry into a computer package, however, the data are normally laid out as given in Figure 3.4b. Here, one column gives the data and two further columns specify the block and treatment codes.

With experimental data, the block and treatment codes are often in a regular sequence. The sequence for blocks in this example is

1 2 3 4 1 2 3 4 1 2 3 4

and the data for treatments are

1 1 1 1 2 2 2 2 3 3 3 3

In Chapter 7, we will see that INSTAT has a quick way of getting such regular sequences as these into the computer.

Figure 3.4 Experimental Data

Fig. 3.4a Standard layout of data for analysis by hand

Block	1	2	3	4
Treat				
1	330	288	295	313
2	372	340	343	341
3	359	337	373	302

Fig. 3.4b Layout for computer entry.

Count	Block	Treat
330	1	1
288	2	1
295	3	1
313	4	1
372	1	2
340	2	2
343	3	2
341	4	2
359	1	3
337	2	3
373	3	3
302	4	3

3.5 EXAMPLE 4: MONTHLY RAINFALL RECORDS

File on Master Disc: @.RAIN

Figure 3.5a consists of 10 years of monthly rainfall totals for Galle in Sri Lanka. This is an example of data that are collected routinely. Here the appropriate form of input may be as in Figure 3.5a or Figure 3.5b, depending on the type of analysis that is required. Figure 3.5b is appropriate if a time series analysis of the whole set of data is to be undertaken, whereas Figure 3.5a is convenient if the data from the separate months are to be considered separately. This would, for example, be the case if a further column gives yields for some crop and one objective is to see which month's rainfall is closely related to these yields. One corollary from this example is that a statistics package should provide facilities to 'reshape' a set of data once it has been entered. One way in which this is done is considered in Chapter 8.

Figure 3.5 Monthly Rainfall Records

Fig. 3.5a Data for 1967 - 1976

Year	Jan.	Feb.	March	April	May	June	July	August	Sept.	Oct.	Nov.	Dec.
1967	89.4	81.0	112.0	154.9	590.3	174.2	292.4	252.7	335.8	457.7	121.2	102.1
1968	78.2	1.3	109.7	73.7	234.4	247.4	293.1	60.2	114.5	265.4	200.4	133.4
1969	58.9	107.4	61.7	191.8	419.9	114.0	49.5	212.8	89.9	495.8	218.4	553.2
1970	100.6	127.5	169.4	160.5	301.0	110.5	173.7	139.2	130.3	231.9	324.4	92.2
1971	115.3	26.2	106.2	184.2	160.8	242.6	156.5	224.0	484.1	297.7	197.1	185.7
1972	80.5	11.9	54.4	87.4	303.0	213.6	53.1	98.8	388.9	519.9	328.2	104.1
1973	36.3	65.8	213.9	109.7	263.6	361.7	239.0	128.5	157.5	422.4	196.6	113.8
1974	0.5	73.2	54.3	415.8	393.5	193.8	365.5	140.7	394.2	45.2	208.0	121.4
1975	90.9	85.4	156.7	357.4	180.6	286.0	144.8	162.1	155.5	352.3	329.2	100.8
1976	95.7	18.3	87.1	403.1	163.6	99.6	146.3	109.5	50.8	448.3	750.3	312.7

Fig. 3.5b Alternative layout for computer entry

YEAR	MONTH	RAIN
1	1	89.4
1	2	81.0
1	3	112.0
1	4	154 9
1	5	590.3
1	6	174.2
1	7	292.4
1	8	252.7
1	9	335.8
1	10	457.7
1	11	121.2
1	12	102.1
2	1	78.2
2	2	1.3
.	.	.
.	.	.

Chapter 4: THE INSTAT LANGUAGE

4.1 INTRODUCTION

INSTAT commands can be considered as making up a simple language that is, in some ways, similar to BASIC.** Figure 4.1a gives the trivial BASIC code to add two numbers and display the results on the screen. Figure 4.1b gives the INSTAT equivalent. The BASIC language has a wide range of commands to manipulate and display numbers and strings. In statistical analysis we often deal with 'columns' of numbers and INSTAT's 'language' is designed to manipulate and display these columns easily. Figure 4.1c illustrates this, by giving the INSTAT instructions to input and display the sum of two columns. The important thing to note is the similarity between the last two commands in Figures 4.1b and 4.1c.

Figure 4.1 Examples of BASIC and INSTAT code

4.1a

```
BASIC instructions
to add two numbers

> K1=4.2 :K2=7.6
> K3=K1+K2
> PRINT K3

>   11.8
```

4.1b

```
INSTAT instructions
to add two numbers

: K1=4.2 : K2=7.6
: K3=K1+K2
: DISplay K3

  K3=11.8
```

4.1c

```
INSTAT instructions
to input and add
two columns

: READ X1 X2
data  1: 4.7 7.6
data  2: 3.7 1
data  3: 2.6 1.9
data  4: 6   8
data  5:
: X3=X1+X2
: DISplay X3

X3
12.3  4.7 4.5 14
```

The full list of current INSTAT commands is given in Figure 4.2. They are given under the nine TOPIC headings, which the HELP TOPIC command (see Chapter 5) will display. Thus the commands in TOPIC 1 refer to all the commands relating to file handling.

.....

** The standard use of INSTAT corresponds to using BASIC in what is called the 'immediate' mode, i.e. commands are executed after a line is typed. An alternative in BASIC is to put line numbers in front of the commands, which are then stored and executed later, when you type RUN. It is possible to store INSTAT commands and this is described in Chapter 16.

Figure 4.2: Commands in INSTAT divided into Topics **

1.File handling	2.Data entry, display & editing	3.Calculations & simple statistics
: CLOse	: DELete	: CALculate
: CREate	: DISplay	: DEScribe
: INFo	: ENTer	: FREquencies
: INPut	: INPut	: GAMma
: MACro	: INSert	: GENerate
: OPEn	: LOCk	: INDicator
: OUTput	: NAMe	: MACro
	: PREsent	: NORmal scores
	: READ	: PERcentiles
	: REMove	: PRObabilities
	: UNLock	: SHOw (or ?)
		: STAtistics
		: TINterval
		: USE

4.Data manipulation, sorting & ranking	5.Analysis of variance	6.Regression & correlation
: RANk	: ANOva	: ADD
: RECode	: FACtor	: CORrelation
: SELect	: INTeraction	: DROp
: SORt	: ONEway	: ESTimates
	: YVAriate	: FIT
		: INDicator
		: REFit
		: TERms
		: YVAriate

7.Tabulations	8.Graphics & data plotting	9.Miscellaneous	
: FACtor	: BOXplots	: AGAin	: MODe
: PREsent	: DEFault	: CLEar	: NOTe
: TABle	: HISTogram	: CONfigure	: PAGe
	: LINe	: ECHo	: PAUse
	: MODe	: ERRor	: QUIt
	: PLOt	: HEAding	: TITle
	: REPlot	: HELp	: VDU
	: SCAtterplot	: INFo	: WARn
	: STEm	: KEY	
	: SYMbol	: MENu	

** Commands that refer to more than one topic are listed under all the appropriate topic headings.

4.2. INSTAT COMMAND LINES

INSTAT is an interactive language. You enter a line of text from the keyboard The program reads it, looks for a command in that text, searches its dictionaries for the command, checks for errors and, if there are none, executes the command. It then waits for the next command. If an error is found in the command, an appropriate error message is displayed and the command is abandoned.

The prompt ': ' indicates that INSTAT is ready to accept a command. Only the first 3 characters of the command name are used by the computer. Apart from any arguments (columns, numbers, etc.) that the command may need, all other text in a command line is optional and purely for the user's benefit. An important rule is that the INSTAT command itself must be the first word in a command line. Commands and text may be in any mixture of capital and small letters. For example, the following three command lines are equivalent:

```
: ENTER THE FOLLOWING DATA INTO X3
: enter X3
: Ent x3
```

Besides the command prompt ': ', other prompts given by the program include 'sub:' when a subcommand is expected and 'data:' for data to be input.

Two or more commands may be entered on the same line. They must then be separated by a colon. For example,

```
: INF : DIS X1-X4
```

Xn denotes a column, and it must be typed with the letter X (or x) directly in front of the column number, e.g. X12. There must be no spaces or other characters between X (or x) and the column number.

Constants are denoted by K1, K2, ...; labels by L1, L2, ..., and strings by S1, S2, ... See Section 4.3 below for a description of the data types available in INSTAT.

When a filename is referred to in an INSTAT command, it must be prefixed by @ (or @.) with no spaces between @ (or @.) and the filename; for example @RAIN, @.SURVEY.

Many of INSTAT's commands have sub-commands which modify the behaviour of the command. The idea is that it is usually possible to use a command in a simple way, with default options chosen by the program, but should you need more flexibility, the sub-commands are available. (There are just a few commands that need a sub-command in order to work.) If a command has sub-commands, they may follow on the same line at the end of the command itself, separated by semi-colons, or may be entered on subsequent lines if just the semi-colon is typed. If you terminate a command line with a semi-colon, then you will get the prompt

'sub: ' on the next line, which means that the program expects a sub-command. Here is an example (the meaning of this command will be explained in Chapter 6):

```
: CRE @TEST; COL 10 20; CON 10
```

is identical to

```
: CRE @TEST;
sub: COL 10 20;
sub: CON 10
```

Note that if a group of columns with consecutive column numbers are referred to in a command, then you can abbreviate the list. For instance,

```
: disp x1-x3
```

is equivalent to

```
: disp x1 x2 x3
```

The same abbreviation of the syntax can be applied to constants, strings and labels, but you cannot mix data types in the same list.

If you do not remember a command, then the HELP facilities described in Chapter 5 are usually quicker to use than looking in the Reference Manual, although the latter is more complete.

If, because of a mistake in typing in a command, it is not executable, INSTAT will give an error message. For example,

```
: RAED X1-X3          (READ misspelt!)

No such command       (INSTAT's response)

: DISplay             (Does not say what to display)

Command too short

: DEScribe X4         (Forgot to enter the data first!)

No data in X4
```

No doubt you will discover other error messages while you are learning to use INSTAT. We hope that their meaning is clear, although admittedly they are rather terse!

4.3 TYPES OF DATA

INSTAT holds information of various types.

Columns - Most numeric data are held in columns labelled X1, X2, ... A column may be of any length up to a maximum that depends on the configuration and on the size of the worksheet that you are using (this is explained in Chapter 6). Columns may also be named and subsequently referred to by name. The command *NAMe* is used to assign a name to a column. An example is

```
: NAME  X3 'Weight
: display 'Weight
```

See the Reference Manual for rules for names.

Constants or Scalars - Single numbers may be held as constants labelled as K1, K2,....

For example, if the rainfall data in Chapter 3, Section 3.5 are to be transformed from inches to millimetres, then it might be useful to store the number of millimetres in an inch as a constant within the worksheet.

Strings - For some applications, it is useful to be able to store lines of text or 'strings' in the worksheet. They are labelled S1, S2,....

INSTAT uses strings in a variety of ways. They may contain a formula which is to be plotted. This use is described in Chapter 9, while Chapter 16 describes how a number of commands that have to be used repeatedly can be stored in strings and then called when they are needed. Finally, a string may just contain a title, which can subsequently be used with the PLOT or HISTOGRAM commands. For example,

"Plot of t-distributions with 1, 4 and 10 d.f."

Labels - A label column of length m is a set of m names or codes. (e.g. YES NO or NEW OLD TRAD)
Label columns are referred to as L1, L2, ...
An example of their use is given in the next section.

SSP matrix - INSTAT calculates and stores a sum of squares and products matrix before doing regression or correlation analysis. Only one SSP matrix can be stored at a time and it is referred to as V1. Unlike the other data types, you cannot enter an SSP matrix directly from the keyboard. It is calculated from columns by the program (this is described in detail in Chapter 13).

4.4 DIFFERENT TYPES OF NUMBER COLUMN - VARIATES AND FACTORS

The survey data given in Chapter 3, Figure 3.3b are used as an example. When the 6 columns of data are entered they are called 'variates', i.e. the data are just numbers. In the analysis of this set of data, it is useful to indicate that the first and fifth columns (the village and variety codes) are special. The first column is always an integer between 1 and 4 indicating the village. Similarly, the 'variety' column is an integer between 1 and 3. In INSTAT, a column of 'codes' may be called a factor column. Factor columns must be coded from 1 upwards. In Chapter 7, Section 7.6, we discuss how factors can be declared. The village and variety columns could be designated as factor columns with 4 and 3 levels, respectively.

Factor columns can, if necessary, be connected to label columns. For the survey data, Figure 3.3c, shows two columns of labels, the first of which contains the village names for the four villages used in the survey and the second gives the varieties. We show in Chapter 11 how one use of this set of facilities is to produce tables etc., which are labelled with the village names rather than merely code numbers. Figure 4.3 shows an example.

Figure 4.3: Table of mean rice yield

Variety	New	Old	Trad
Village			
Sabey:	59.45	49.18	29.5
Kesen:	*	43.27	24.87
Niko:	*	36.1	26.87
Nanda:	59.75	46.37	36.28

* indicates missing values.

It is also possible to RECode variate columns into columns that can qualify as factors. For example, Chapter 8 gives the instructions to recode the fertilizer data from those given in the fourth column of Figure 3.3 into a new column in such a way that:

0 cwt is recoded to 1 in the new column (No fertilizer)
0.5 - 2 cwt is recoded to 2 in the new column (Some fertilizer)
> 2 cwt is recoded to 3 in the new column (A lot of fertilizer)

These fertilizer codes could then be designated as a FACtor column with 3 levels.

Other types of number columns will be introduced in later chapters but the distinction between factors and variates is the most important. It is the distinction between a qualitative variable (e.g. hair colour, eye colour, village) and a quantitative variable (yield, area etc.,)

4.5 SUMMARY

In this chapter and in Chapter 3, where the sets of data were introduced, these data have been used partly to demonstrate the sort of decisions that can usefully be made before data are entered into INSTAT. It is important to give consideration to how large the worksheet will have to be and what components are needed. Chapter 6 explains how to create a suitable worksheet once these decisions have been made. The data are entered as columns. Often, as in the regression example, considering the data as columns is a trivial step, but on other occasions it may require some thought. Although less important, it is also useful to consider at this stage whether any columns will be designated as factors. All of our examples except the regression example could possibly involve factors. If factors are used, the next decision is whether label columns (e.g. Figure 3.3c) will be entered so that allowance may be made for them when the worksheet is created.

Chapter 5: GETTING HELP

5.1 INTRODUCTION

There are a variety of different 'on-line' HELP facilities in INSTAT and users should find that, with experience, they will often be able to use these facilities as an alternative to referring to this User Guide. To get the full benefit of the HELP command, it would be a good idea to spend a little time practising with it, before really starting to use INSTAT.

The amount of information that the HELp command provides can be set to one of two levels, FULL or BRIEF. The idea is that an experienced user will probably be satisfied with a short reminder of how to use a command, whereas a novice will usually want more detailed information. The level of all subsequent HELP information is set by entering the command

: *HELP FULL*

or

: *HELP BRIEF*

The default level of HELP will normally be FULL, but this can be changed to BRIEF when INSTAT is configured (see Chapter 2).

5.2 GENERAL INFORMATION

With the level of HELP set to FULL, entering the command

: *HELP*

on its own gives detailed instructions on further use of the HELp command itself. If HELP has been set to BRIEF, it just gives the basic syntax of the command.

The command

: *HELP OVERVIEW*

gives a short description of INSTAT and, in particular, how to enter commands. All of INSTAT's commands can be displayed by entering

: *HELP TOPIC n*

where n is a number from 1 to 9. These 'TOPICs' are just subject

groupings of commands. For example HELP TOPIC 5 lists the commands associated with analysis of variance. The TOPIC numbers are given as part of the output from HELP on its own (with the HELP level set to FULL). For convenience, they are reproduced in Figure 5.1.

Figure 5.1 TOPIC numbers

NUMBER	TOPIC
1	File Handling
2	Data entry, display and editing
3	Calculations and simple statistics
4	Data manipulation, sorting and ranking
5	Analysis of variance
6	Regression and correlation
7	Tabulation and analysis of tables
8	Graphics and data plotting
9	Miscellaneous

A list of the commands, under each of these topics, is given in Chapter 4, Figure 4.2

5.3 GETTING HELP ON A COMMAND

To get help on the use of a particular command, just enter HELP followed by the name of the command. For example,

: *HELP DISPLAY*

or just

: *HEL DIS*

gives information on the DISPLAY command. The output that you get depends on the level of HELP required. With HELP set to FULL, the output consists of the syntax of the command, a list of its sub-commands (if any), a description of what the command does, and in most cases an example or two of how to use it.

When the level of HELP is set to BRIEF, the information given on a command is just two or three lines indicating the syntax of the command and a short note on what it does. If the command has sub-commands, they are not shown in the BRIEF help unless they are essential for the command to work.

Figure 5.2 gives an example of both BRIEF and FULL help for the command *TINterval*.

Figure 5.2 HELP for the TINterval command

```
: HELP BRIef
: HELp TINterval

: TINterval Xn1 (Xn2)
 calculates one- (or two-) sample t-intervals and tests.

: HELp FULL
: HELp TINterval

: TINterval Xn1 (Xn2)

Sub-commands: ;CONfidence p
              ;TESt mu

 Displays confidence intervals for the mean of a column or for
the difference in means of two columns, and performs t-tests on
means or differences of means.

 : TINt Xn gives a 95% confidence interval for the mean of Xn.

 : TINt Xn1 Xn2 gives a 95% interval for the difference in means
of Xn1 and Xn2.

In either case, the confidence level may be changed by the sub-
command ;CON. p must lie between 0 and 1, or between 0% and 100%.

 Subcommand ;TESt is used for t-tests.

 Examples:

 : TIN X2; CON .99  or  : TIN X1; CON 99 will give a 99% interval.
    ..................
    ..................
 : TIN X1 X2; TES 0  does a t-test of equality of means.
```

5.4 THE HELP SUB-COMMAND

Even experienced users will occasionally forget how to use a command. It may happen that you have started entering a command line and then need to be reminded of the sub-commands available for the command. Adding the sub-command ;HELP to the command line (before pressing ⟨*RETURN*⟩) will list all possible sub-commands. You will then be prompted for the sub-command that you want. For example,

```
: dis x4-x10; help
Sub-commands are: FRO TO COL FIX ACC WID PRI HEL LAB
sub: FIX 2
```

and columns X4-X10 will be displayed with 2 decimal digits.

Chapter 6: WORKSHEETS

6.1 INTRODUCTION

INSTAT requires data to be stored in a special kind of disc file called a worksheet. This chapter explains how to create a new worksheet and discusses the decisions that first need to be made. We also explain how to use an existing worksheet and get summary information about the contents of a worksheet. Some of the example data sets introduced in Chapter 3 are used as illustrations of some of these ideas.

6.2 CREATING A NEW WORKSHEET

The command that sets up a new worksheet file on the disc is CREate. Normally, you would want to specify the amount of space required for your data and choose your own name for the worksheet. The command has sub-commands that allow you to choose the amount of space to allocate for columns, constants, strings, labels and an SSP matrix. However, the command : *CREate* on its own, with no sub-commands or arguments, will create a small worksheet file with the name @.TEMP in the disc directory. By default, the amount of data space reserved in this worksheet is as follows:

10 columns of maximum length 25
10 constants
5 strings
No labels
An SSP matrix for 10 columns

It is possible to alter these default settings by using the command : *CONfigure* 2.

When the CREate command is executed, you will be asked to supply a title for the worksheet (if you don't want a title, just press *<RETURN>*). This title should not be confused with the worksheet filename on the disc. It will appear whenever you subsequently use the worksheet. The worksheet used for the regression example was created by:

```
: CRE @REGRESS
Enter title for worksheet (or RETURN).
Regression Example
```

Note that this is a small data set so the default worksheet size is sufficent, although we have given it a name of our own choice. Recall that when filenames appear in a command line, they must have the prefix '@' (or '@.').

The number and maximum length of columns required can be specified by using the COLumns sub-command. For instance,

```
: CRE @EXDATA; COL 10 100
```

reserves space for 10 columns of length 100. Other sub-commands are similar. To make space for 10 strings, 4 label columns each with 5 labels, and an SSP matrix derived from 8 columns, for example,

```
: CRE @EXDATA; COL 10 100; STR 10; LAB 4 5; SSP 8
```

It should be clear by now that a little foresight is needed when creating a new worksheet. Remember that you will almost always want space for more data than the original data set requires. You may want to derive new columns, or save residuals and fitted values, and so on. If you are going to do regression, then you will need to consider how many variables are likely to be used so that there is enough space for the SSP matrix. Strings are used for many things in INSTAT and it is often a good idea to set aside space for some. The same applies to constants. It is not a disaster if your worksheet turns out not to have enough space, and in Chapter 7 we will see how to move data from one worksheet to another (bigger) one. However, to avoid the trouble, it is worth making an effort to get it right at the start. As an example, here is the command that was used to create the worksheet for the survey example:

```
: cre @survey; col 20 36; lab 4 4
Enter title for worksheet (or RETURN).
Rice Survey data
:
```

Notice that although the original data set has 6 columns of length 36, the worksheet has space for 20 columns. In Chapter 4 we mentioned that we would want to label the villages and varieties, so we have made space for label columns in the worksheet.

See the Reference Manual for a complete description of the CREate command.

6.3 USING AN EXISTING WORKSHEET

Once a worksheet file has been set up on disc, we want to be able to access the data in it, during a later INSTAT session. The command to do this is OPEn. You must specify a filename with this command. For example,

```
: OPEN @REGRESS
Regression Example
```

Note that the title of the worksheet (if it has one) is displayed. The OPEn command has a sub-command TITle, which is used for changing the title of the worksheet, or for giving it one if it did not have one.

If you are using a dual disc drive, you may wish to keep your data files on one drive and the program files on the other. In this case, it may be necessary to use the DFS command **DRIVE* (which INSTAT recognises) before OPEning the file.

The currently open file can be closed before OPEning another by giving the command : *CLOse*, although this is not strictly necessary.

6.4 GETTING INFORMATION ON THE CONTENTS OF A WORKSHEET

Once a worksheet has been OPEned or CREated, we often want to check what data it contains and how much space is left. The command INFormation provides summary information about the contents. The command can also be used to check the contents of another worksheet on the same disc.

Figure 6.1 INFormation command

```
: OPEN @REGRESS
Regression Example

: INFO

     Worksheet filename: REGRESS
  Regression Example

Strings :     (S1 to S5 unused)

Constants :  (K1 to K10 unused)

Columns :Day   Yield X3    X4    X5
Length  :10    10    Free  Free  Free

Columns :  (X6 to X10 unused)

SSP Matrix:  (V1 unused)
```

The command : *INFo* on its own gives a brief summary of the contents of the currently open worksheet, and there are a number of sub-commands which can be used to get more detailed information about data of different types. The sub-commands are described in the Reference Manual, but we mention here the FREe sub-command, which tells us how much space remains on the worksheet for different data types. Note that the INFo command provides information *about* the data as opposed to displaying the actual data (commands for this are described in Chapter 7). Figure 6.1 shows the output from the INFormation command for the regression example, before any analysis is done. Notice that two of the columns have been given names. Figure 6.2 illustrates the INFormation command both on its own, and with a sub-command which gives more details about the columns in the survey example.

Figure 6.2 INFormation with a sub-command

```
: open @survey
 Rice Survey data

: information

  Worksheet filename: SURVEY
 Rice Survey data

 Labels  :L1    L2    L3    L4    L5
 Length  :3     3     4     Free  Free

Columns :Villa Field Size  Fert. Varie Yield
Length  :36    36    36    36    36    36

Columns :  (X7 to X20 unused)

  SSP Matrix:  (V1 unused)


: info; cols

   Worksheet filename: SURVEY
  Rice Survey data

  Cols.  Name Length Type  State Pointers
  X1     Villag 36   Factor  U     L3
  X2     Field  36   Variate U
  X3     Size   36   Variate U
  X4     Fert.  36   Variate U
  X5     Variet 36   Factor  U     L1
  X6     Yield  36   Variate U
( X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 Free)
```

The meaning of 'State' and 'Pointers' in this output is explained in Chapter 7.

To see summary information about any worksheet on the same disc other than the one currently open (if any), simply specify the filename after the INFo command - e.g. : *INFO @EXPERI*. However, the sub-commands are only available with INFo, when used on the current worksheet.

6.5 STORING A WORKSHEET IN MEMORY

The contents of an INSTAT worksheet are, if possible, automatically stored in memory and the status line (heading) indicates this by putting (m) after the name of the active worksheet. Many commands are then executed considerably faster. With a 6502 Second processor, quite large worksheets can be stored in memory.

Chapter 7: ENTERING, DISPLAYING AND EDITING THE DATA

7.1 INTRODUCTION

Having created a worksheet file, the next job is usually to get your data into it. For some data types, there may be more than one way of achieving this. In this chapter, we use the four examples introduced in Chapter 3 to illustrate INSTAT's commands for accepting data from the keyboard. Once in the worksheet, we will want to inspect the data, and edit it in the event that there are errors. Other useful features introduced here are commands for locking your data to protect it, giving names to columns, and reading data from ASCII files.

7.2 PUTTING DATA INTO COLUMNS AND NAMING THEM

The Regression Example

We saw, in Chapter 6, the command used to create the worksheet for this example. The commands that were used for entering the data (see Figure 3.2) are shown in Figure 7.1. Notice that we have decided to enter the two columns one variable (or column) at a time, rather than by row. The command for doing this is ENTer. Another point to note is that we have referred to the columns by name. When entering data into named columns, INSTAT looks for the first available free columns, in this case X1 and X2, since the worksheet was initially empty.

Figure 7.1 The ENTer command

```
 : CREate @REGRESS
 Enter title for worksheet (or RETURN).
Regression Example

 : ENTer 'Day
data  1: 1 16 31 46 61 76 91 106 121 136
data 11: EOD

 : ENTer 'Yield
data  1: 17.39 17.74 16.02 14.94 13.88
data  6: 9.78 7.38 6.09 4.26 3.92
data 11: EOD
```

We can use the DISplay command to look at the columns entered, as shown in Figure 7.2

Figure 7.2 The DISplay command

```
: DISplay 'Day 'Yield

Row          Day        Yield

  1            1        17.39
  2           16        17.74
  3           31        16.02
  4           46        14.94
  5           61        13.88
  6           76         9.78
  7           91         7.38
  8          106         6.09
  9          121         4.26
 10          136         3.92
```

The Survey Data

In this case, suppose that we decide to enter the data by case, that is by row. The command to do this is REAd. In Figure 7.3, we again choose to name the columns, and do so by referring to them by name directly.

Figure 7.3 The REAd command

```
: CREate @SURVEY ;COLumn 20 36 ;LABels 4 4
Enter title for worksheet (or RETURN).
Rice Survey data

: REAd  'Village  'Field  'Size  'Fert.  'Variety  'Yield

 data  1: 1   3   2  2.5  2  53.6
 data  2: 1  13   5  1.5  2  44.6
 data  3: 1   4   5  3.0  2  50.7
 data  4: 1  20 1.5    1  3  33.6
 data  5: 1  19   5  2.5  1  62.1
 data  6: .   .   .    .  .     .
   .      .   .   .    .  .     .
 data 36: 4   9 1.5    2  1  61.4
```

The Experimental Data

This set of data consists of 3 columns each of length 12. One feature of the data is that the block and treatment codes are regular sequences. The treatment codes can be entered as

```
: ENTer X3
data 1: 1 1 1 1 2 2 2 2 3 3 3 3
data 13: EOD
:
```

but the shorthand way

```
: ENTer X2
data 1: 4(1]3)
data 13: EOD
```

may be slightly quicker. See ENTer in the Reference Manual for a full description of facilities for entering regular sequences. The commands used for creating the worksheet @EXPERI and entering the data are in Figure 7.4. This time the columns are named after the data have been entered, using the NAMe command.

Figure 7.4 Another example of the ENTer command

```
: cre @experi
Enter title for worksheet (or RETURN).
 Randomised Block Design (M&C pp 52)
: enter x1
data  1: 330 288 295 313
data  5: 372 340 343 341
data  9: 359 337 373 302
data 13: eod
: enter x2
data  1: (1]4)3
data 13: eod
: enter x3
data  1: 4(1]3)
data 13: eod
: name x1 'Count  x2 'Block  x3 'Treat
```

Fig. 7.4 cont'd

```
: display x1-x3

Row        Count        Block        Treat

  1          330            1            1
  2          288            2            1
  3          295            3            1
  4          313            4            1
  5          372            1            2
  6          340            2            2
  7          343            3            2
  8          341            4            2
  9          359            1            3
 10          337            2            3
 11          373            3            3
 12          302            4            3
```

7.3 TRANSFERRING DATA FROM ANOTHER WORKSHEET

It is sometimes convenient to copy all or part of the data resident in another worksheet on the disc to the currently open worksheet. For example, you may find that your worksheet does not have sufficient space for new columns that you want to use. You should then CREate a new worksheet of sufficient size and copy the data across from the old one. The command that does this is INPut. The Reference Manual gives full details of its use. It can be used simply to transfer data in exactly the same form as it is in the other worksheet, but here we illustrate its use with a sub-command that copies columns so that they are transposed at the same time.

The Rainfall Data

Suppose that the rainfall data that we want in worksheet file @RAIN have already been entered into another worksheet called @GALLE. To illustrate a common problem, it is assumed that the instructions to the person entering the data were not precise and, as shown in Figure 7.5, they are the wrong way round. The first column, X1 contains all the data for 1967, X2 contains the data for 1968 etc., but what is required is for X1 to contain the data for January, X2 for February, etc. That is, the columns need to be transposed.

Figure 7.5 The Rainfall Data

```
: MODE 0
: OPEn @GALLE
Rainfall Data for Galle
: DISplay X1-X10; WIDth 7

Row      X1     X2     X3     X4     X5     X6     X7     X8     X9    X10

  1    1967   1968   1969   1970   1971   1972   1973   1974   1975   1976
  2    89.4   78.2   58.9  100.6  115.3   80.5   36.3    0.5   90.9   95.7
  3    81.0    1.3      .      .      .      .      .      .      .      .
  4   112.0  109.7      .      .      .      .      .      .      .      .
  5   154.9   73.7      .      .      .      .      .      .      .      .
  6   590.3  234.4      .      .      .                                  .
  7   174.2      .      .      .                                         .
  8   292.4      .      .      .                                         .
  9   252.7      .      .                                                .
 10   335.8      .      .                                                .
 11   457.7      .      .                                                .
 12   121.2      .      .                                                .
 13   102.1  133.4  553.2   92.2  185.7  104.1  113.8  121.4  100.8  312.7
```

To remedy this, after CREating the worksheet @RAIN10 (Figure 7.6), the data are transferred from the worksheet @GALLE. This uses the INPut command together with the sub-command TRAnspose, which instructs the program to make the rows of the initial worksheet into the columns of the new one.

Figure 7.6 The INPut command

```
: MODe 0
: CREate @RAIN10; COL 20 10; CONstants 10; STRing 5
Enter title for worksheet (or RETURN).
Rain Data for 1967-1976
: INPut @GALLE X1-X10 ;INTo X1-X13 ;TRAnspose
: NOTE The INPut command is used to transfer & transpose the
: NOTE data from  worksheet @.GALLE into @.RAIN10
: DISplay X1-X13

(The data display is the same as Figure 3.5)
```

7.4 ENTERING CONSTANTS, STRINGS AND LABELS

So far we have seen how to get data into columns in three ways: using the ENTer command, using the REAd command and by copying from another worksheet. This short section deals with entering data of other types. The command that does almost all of the work is again ENTer. In fact, this is quite a powerful command and the Reference Manual shows a number of its uses. You can, for instance use ENTer to concatenate columns either with other columns, or with new data.

Storing a number in a constant Kn is particularly simple, and can be done in two ways. For example, both of the following commands achieve the same effect:

```
: ENT K3
data 1: 2.345

: K3 = 2.345
```

The second command implicitly uses the CALculate command.

Putting data into a string Sn is just as simple:

```
: enter s2
S2: This is too easy!
```

Note that you get 'S2:' as a prompt, and that no quotation marks are needed around your text.

Again, use ENTer for entering data into a label column Ln. For instance, the labels for variety names in the survey example could be entered as follows:

```
: ENTER L1
data 1: New Old Trad
data 4: EOD
```

Note that 'EOD' (for End Of Data) is optional - you can just do ⟨*RETURN*⟩ when finished. There are more examples in Figure 7.7.

7.5 DISPLAYING AND EDITING DATA

Once the data have been entered, the next job should normally be to look at what you have entered, check for errors and make any necessary corrections. If a printer is connected, it may be preferable to print the data for the purposes of checking, especially if it is a large data set. Facilities for displaying, printing and editing data are illustrated with the rice survey data from Figure 7.3. Part of an INSTAT session which illustrates these techniques is reproduced in Figure 7.7.

The work was done during another INSTAT session, after the data entry had been completed, so we begin by inserting the disc containing @.SURVEY and then typing

: *OPEn @SURVEY*

to make this the current worksheet.

The DISplay command is used to show the data on the screen. We have already seen examples of its use in Section 7.2 above. If the printer is activated by pressing ⟨*CTRL*⟩ *B* or using the command : *VDU* 2, the data are printed at the same time. ⟨*CTRL*⟩ C or : *VDU* 3 turns the printer off again. The form of the output depends on whether INSTAT is in a 40 or 80 column mode. All six columns are displayed in the 80 column mode. In 40 column modes, only the first three columns are displayed, followed by the last three columns, after the SPACE BAR has been pressed. This command has many sub-commands which control the format of the output. They are explained in the Reference Manual. Here is an example:

```
: DISplay X1-X6; WIDth 5; FROm 18; TO 20

    Row Vill Fiel Size Fert Vari Yiel

     18    1    3    2  4.5    0 26.3
     19    1    3    8  2.5    0 24.7
     20    1    3    7    6  0.5 40.4

      Press SPACE BAR to continue
```

Entering data from the keyboard is a notoriously error-prone activity, and there are various mistakes that commonly occur. We deal here with some of them, but undoubtedly there are other kinds of error that will need correcting. Rows may be overlooked and need to be inserted afterwards. Rows may be duplicated and, of course, simple transcription errors are quite frequent.

Figure 7.7 includes some examples of techniques for editing data in columns. Some explanatory notes have been added during the INSTAT session using the NOTE command. A few other remarks may also be helpful:

(1) The INSert command has here been used to insert a single row, but you can, if required, carry on inserting further rows just as if you were using the REAd command.

(2) The command : *X4(9) = 2* uses the CALculate command implicitly.

Figure 7.7 Displaying and Editing Data

```
   : MODE 3
   : OPEN @SURVEY
   Rice Survey data

   : DISplay X1-X6

Row   Village     Field      Size      Fert.    Variety      Yield

  1         1         3         2       2.5          2       53.6
  2         1        13         5       1.5          1       44.6
  3         1         4         5         3          2       50.7
  4         1        20       1.5         1          3       33.6
  5         1        19         5       2.5          1       62.1
  6         1        19         5       2.5          1       62.1
  7         1         8       4.5         2          2       37.7
  8         1        11       3.5         0          3       24.3
  9         1        14         5        22          1       56.8
 10         1        12       4.5       2.5          2       59.3
  .         .         .         .         .          .          .
  .         .         .         .         .          .          .

   : NOTE row 5 has been duplicated and row 6 omitted
   : DELete row 6 for columns X1-X6
   : INSert from row 6 for X1-X6
   data  6: 1 10 4 1.5 3 30.6
   data  7:

   : NOTE Fert. value for row 9 & variety for row 2 are wrong

   : X4(9)=2

   : X5(2)=2

   : NOTE Editing complete so display data again

   : DISplay X1-X6; FROm 1; TO 10

Row   Village     Field      Size      Fert.    Variety      Yield

  1         1         3         2       2.5          2       53.6
  2         1        13         5       1.5          2       44.6
  3         1         4         5         3          2       50.7
  4         1        20       1.5         1          3       33.6
  5         1        19         5       2.5          1       62.1
  6         1        10         4       1.5          3       30.6
  7         1         8       4.5         2          2       37.7
  8         1        11       3.5         0          3       24.3
  9         1        14         5         2          1       56.8
 10         1        12       4.5       2.5          2       59.3
```

Fig. 7.7 cont'd

```
    : NOTE Enter labels into L3 with Village names
    : ENTer L3
    data  1: Sabey Kesen Niko Nanda

    : NOTE Enter labels in L1 for rice varieties
    : ENTer L1
    data  1: New Old Trad
    data  4:

    : ENTer L2
    data  1: 0cwt .1-1.9cwt  >1.9cwt
    data  4:

    : NOTE declares X1 (Village) as Factor with 4 levels
    : NOTE (i.e length of L3)

    : FACtor X1 L3

    : NOTE declares X5 (Variety) as Factor with 3 levels
    : NOTE (i.e. length of L1)

    : FACtor X5 L1

    : INFO

       Worksheet filename: SURVEY
     Rice Survey data

     Labels  :L1    L2    L3    L4    L5
     Length  :3     3     4     Free  Free

Columns :Villa Field Size  Fert. Varie Yield
Length  :36    36    36    36    36    36

Columns :  (X7 to X20 unused)

     SSP Matrix:  (V1 unused)


   :  INFormation; COLumns

       Worksheet filename: SURVEY
     Rice Survey data

     Cols.  Name Length Type  State Pointers
     X1     Villag 36   Factor  U    L3
     X2     Field  36   Variate U
     X3     Size   36   Variate U
     X4     Fert.  36   Variate U
     X5     Variet 36   Factor  U    L1
     X6     Yield  36   Variate U
  ( X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 Free)
```

(3) The command DELete is used to delete a row from several columns. Deletion of one or more entire columns requires the command REMove. In fact, REMove can be used to get rid of unwanted data of *any* type from the worksheet. Some examples:

```
: REMove X9-X11
: rem k2 k4-k6
: REM S2
```

and so on. After REMoving data, the space it occupied is effectively 'empty' and ready for new data.

(4) The usual way of displaying data is with the DISplay command. However, another way of looking at a constant or a single column is to use the '?' command (which is the short form of the command SHOw). This command is really intended for displaying the result of a calculation without saving it. These examples show how '?' can also be used just to display data:

```
: ? X2
: ? K6
```

7.6 DECLARING FACTOR COLUMNS

In Chapter 4, a distinction was made between two types of column: variates and factors. Factor columns can only contain positive integer values 1, 2, ..., which represent codes for a categorical or qualitative variable. Some INSTAT commands (notably ANOVA and TABLE, but there are others) require that factor columns have previously been declared as such. The declaration is accomplished by the FACtor command. For example, suppose that X12 is to be declared a factor with 4 levels and X13 a factor with 2 levels. The command is:

```
: FACtor X12 4 X13 2
```

An alternative is when you want to attach labels to the factor levels. There are two examples of the use of FACtor with labels in Figure 7.7; note that they could have been combined into a single command:

```
: FACtor X1 L3  X5 L1
```

The effect of this is to declare X1 a factor with number of levels equal to the length of the label column L3 (namely 4), and similarly X5 becomes a factor with 3 levels. The labels themselves become 'attached' to the factor levels and will automatically appear in the output of certain commands (ANOVA, TABLE, etc).

7.7 PROTECTING DATA

A certain measure of security is automatically provided for the data in your worksheet, unless you have used the WARnings command to turn warnings off. Any attempt to overwrite data will prompt INSTAT to ask you whether you really want to lose the original data. However, it is still possible to REMove data without getting any warning. To give extra protection for columns, there is a command LOCk, which can be used to give complete protection to columns of important data. For example:

```
: LOC X6
: REM X6
```

will produce the message 'Locked: X6'. Once locked, a column cannot be overwritten or edited in any way. The protection can be lifted by the UNLock command, i.e.

```
: UNL X6
: REM X6
```

and it's gone.

In the output from the INFO command, the column headed 'State' tells you whether the columns are unlocked (U) or locked (L).

Sometimes, INSTAT locks columns automatically, without being told to do so with the LOCk command. The columns are then said to be 'system-locked'. This happens when one data structure is derived from others, or attached to them in some way. If the original columns were changed, then it might be possible to get contradictory results. For example, in Figure 7.7, the label columns L1 and L3 have been used to declare some factors. If the labels were altered, then the definition of the factors may no longer be compatible with the labels. In fact, L1 and L3 have been locked by the system. Other instances of system-locking will arise in later chapters. If, in the example in Figure 7.7, you were to get INFO on label columns, the output should look something like Figure 7.8 (we will use L2 in the next chapter).

Figure 7.8 Label Information

```
: info; lab

     Worksheet filename: SURVEY
 Rice Survey Data

Labels     Length  State
L1         3         U 1L
L2         3         U 1L
L3         4         U 1L
( L4 Free)
```

The meaning of the 'State' column is as follows: 'U' means that you have not locked the columns yourself with the LOCk command; '1L' means that the column has been locked by the system once. It is possible that several different data structures could be associated with L3, say, in which case instead of '1L', you would get '2L', or '3L', and so on, depending on how many times the system has locked the column.

The only way to lift the protection imposed by system-locking is to remove all of the structures which caused the locking in the first place. For example, in Figure 7.7, if we were to REMove the factor X1, then L3 would no longer be system-locked. To discover which structures are causing the system-locking, look at the column headed 'Pointers' in the output from the INFO command (referring back to Figure 7.7). We see that X1 'points to' L3 and X5 'points to' L1. This implies that it is X1 which is responsible for the locking imposed on L3, and X5 which causes L1 to be locked.

7.8 READING DATA FROM ASCII FILES

It is possible to read data from a standard ASCII file on disc provided that you have access to word-processing or text-editing software. The editor should first be used to insert the INSTAT commands in the file which would be required to read the data if it were entered from the keyboard. If, for example, the file contains 5 columns of data and you want to read them into columns X1 - X5 in an INSTAT worksheet, then the file should be edited so that the line before the data is 'READ X1-X5'. To read the data into an INSTAT worksheet, go into INSTAT, CREate or OPEn the worksheet, and use the DFS command

: **EXEC filename*

During the development of INSTAT, we have sometimes transferred data files from a mainframe computer to the BBC Micro using suitable communications software (for example to use INSTAT for things which certain well-known mainframe packages could not cope with!). In such cases, it is possible to use an editor on the mainframe to put in the necessary INSTAT commands, before doing the file transfer.

It is not entirely satisfactory to have this as the only way of reading data from ASCII files. We intend to have better facilities in the next release of INSTAT.

Chapter 8: GETTING THE DATA INTO SHAPE

8.1 INTRODUCTION

The effort devoted to the analysis of data often does not do justice either to the effort and expense of their collection, or to the value of the data. There are a variety of reasons, but two common failings are as follows:

(1) There is sometimes a hope of a single 'right' analysis. More often a variety of analyses reveal different aspects of the data.

(2) The raw data are often summarised before entry to a computer. Although the resulting data can usually be analysed simply, this initial step of summarising the data has sometimes averaged over much that is of value.

This chapter illustrates the commands for transforming and recoding data and for splitting data into different subsets. Later chapters, particularly Chapter 10, demonstrate further commands for data summary which allow the summaries to be stored for further analyses. Together, these facilities are designed to encourage users to try alternative analyses on their data and to enter all the (raw) data where possible, in the knowledge that subsets or summaries of the data can be derived when needed.

8.2 MAKING NEW COLUMNS WITH CALculate, RECode AND ENTer

Figure 8.1 shows a simple use of the CALculate command on the data in the file @.REGRESS to derive quadratic and cubic terms from the x variable stored in column X1. These are used in Chapter 13 to fit alternative polynomial models relating the yield to the date of planting. The CALculate facility is so important in data analysis that it has been honoured by not requiring the three letters of the command to be given. For instance, these two command lines are equivalent:

```
: CALc X3 = X1*X1
: X3 = X1*X1
```

(This is reminiscent of many dialects of BASIC, including BBC BASIC, where the LET command is optional.)

Figure 8.1

```
: OPEn @REGRESS
Regression Example
: X3=X1*X1
: X4=X3*X1
: DISplay X1-X4
```

Row	Day	Yield	X3	X4
1	1	17.39	1	1
2	16	17.74	256	4096
3	31	16.02	961	29791
4	46	14.94	2116	97336
5	61	13.88	3721	2.2698E5
6	76	9.78	5776	4.3898E5
7	91	7.38	8281	7.5357E5
8	106	6.09	11236	1.191E6
9	121	4.26	14641	1.7716E6
10	136	3.92	18496	2.5155E6

The CALculate command is discussed further in Section 8.4 below.

Another way of deriving new columns from variates is to use the RECode command. Suppose that we want to group the values in a column into mutually exclusive classes. The RECode command is illustrated in Figure 8.2, which consists of some further analysis of the survey example, introduced in Section 3.3. The meaning of the entry following the first 'data :' prompt is that each occurrence of 0 in X4 is recoded into 1 in X7. The next line means that all values in X4 which lie between 0.1 and 1.9, inclusive, will become 2 in X7, and so on. Since the recoded values in X7 are now all positive integers, X7 can be declared a factor column, and labels for the 3 levels are provided in L2. It will be used in Chapter 11 to produce a two-way table of the data categorised by the amount of fertilizer used and the variety of rice.

A third problem is how to concatenate or join columns together. For example, if the experimental data had been entered into 4 columns, X2-X5, each of length 3, the following examples use the ENTer and CALculate commands to show alternative ways of joining them together into one column, X1, of length 12.

```
either  : ENTer X1
          data  1: X2 X3 X4 X5
          data 13: EOD

or        : X1=X2 : X1(+)=X3 : X1(+)=X4 : X1(+)=X5
```

Figure 8.2 The RECode and SELect commands

```
: OPE @SURVEY
Rice Survey Data
: INF

     Worksheet filename: SURVEY
Rice Survey Data

Strings :   S1  S2   (8 unused)

Labels  :L1    L2    L3    L4    L5
Length  :3     3     4     Free  Free

Constants : K1  K2   (8 unused)

Columns :Villa Field Size  Fert. Varie
Length  :36    36    36    36    36

Columns :Yield X7    X8    X9    X10
Length  :36    Free  Free  Free  Free

Columns :  (X11 to X20 unused)

SSP Matrix:  (V1 unused)

: RECode X4 into X7
data : 0 1
data : 0.1 1.9 2
data : 1.91 3 3
data : eod
: SELect X1-X6 into X8-X13;IF (X5=3)

Number of cases = 15

: DISplay X8-X13
```

Row	X8	X9	X10	X11	X12	X13
1	1	20	1.5	1	3	33.6
2	1	10	4	1.5	3	30.6
3	1	11	3.5	0	3	24.3
4	2	8	7	0	3	25.8
5	2	4	8	0	3	27.6
6	2	3	2	0.5	3	27
7	2	5	4	0	3	19.1
8	3	2	4.5	0	3	26.3
9	3	8	2.5	0	3	24.7
10	3	4	8	0	3	29.6
11	4	4	4.5	2	3	36.6
12	4	13	20	3	3	42.7
13	4	8	3	0	3	37.6
14	4	28	2	2	3	38.7
15	4	5	4.5	1.5	3	25.8

8.3 CHOOSING SUBSETS OF THE DATA

The SELect command is used to choose subsets of the data. In Figure 8.2, all fields with the traditional varieties of rice have been selected for separate analyse, if required.
The command

```
: SELect X1-X6 into X8-X13; IF (X5=3)
```

has 3 components. The first part states that the data from which we are selecting are in colmuns X1 - X6. The second section gives the columns where the selected data are to be stored and the IF sub-command gives the condition for selection. This is a powerful command because the conditions can, if necessary, be quite complicated. For example,

```
: SELect X1 X2 X6 into X14-X16; IF (X3>4) AND (X5=1) AND (X6>60)
```

selects all large farmers growing new improved varieties with a yield greater than 60 bushels/acre. Here there is only one such farmer so further summary is simple!

8.4 CALCULATING WITH COLUMNS

Simple instances of the CALculate command have already been encountered in this and earlier chapters. Figure 8.3 gives further illustrations with the rainfall data, which was introduced in Section 3.5.

The command normally produces a new column from one or more existing ones. For example, a commonly used transformation,

```
: X15 = SQR(X2)
```

takes the rainfall totals for January and stores their square roots in column X15.

```
: X14 = X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13
```

stores the annual totals in X14.

Sometimes a calculation involving one or more columns produces a single number as result. The result can be stored in a constant, as in

```
: K1 = MEA(X14)
```

which saves the mean of column X14 in the constant K1. Then,

```
: X16 = ((X14-K1)/K1)*100
```

gives the percentage departure of the annual totals about the

mean. These two steps could, if necessary, have been combined to

```
: X16 = ((X14 - MEA(X14))/MEA(X14))*100
```

Experienced BBC users who wish to exploit this command to the full may choose to use it instead of the RECode command. Thus

```
: X7 = 3 - (X5 < 1.9) - (X5 < 0.4)
```

gives the same result as the RECode command used in Figure 8.2. Note that BBC BASIC evaluates expressions such as (X5 < 0.4) as either

TRUE, which is given the value -1,
or
FALSE, which is given the value 0.

This last example indicates the flexibility of the CALculate command. It is very powerful and its description in the Reference Manual is the longest of all the commands. Further examples of its use are also given in Chapter 10 on summarising data, in Chapter 14 on probability distributions and in Chapter 15 on random numbers.

Figure 8.3 Calculating with Columns

```
: MODe 3
: OPE @RAIN10
Rainfall for 1967-1976

: X14=X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13
: X15=SQR(X2)
: K1=MEAN(X14)
: X16=((X14-K1)/K1)*100
: DISplay X2 X15 X14 X16
```

Row	Jan.	X15	Total	X16
1	89.41	9.4557	2763.8	16.93
2	78.24	8.8453	1811.7	-23.35
3	58.93	7.6766	2573.5	8.88
4	100.6	10.03	2061.2	-12.80
5	115.32	10.739	2380.3	0.70
6	80.53	8.9739	2243.8	5.07
7	36.33	6.0274	2308.9	2.31
8	0.51	0.71414	2406.1	1.80
9	90.91	9.5347	2401.6	1.61
10	95.75	9.7852	2685.3	13.61

Chapter 9: PLOTTING DATA

9.1 INTRODUCTION

A recent trend in data analysis is an increasing emphasis on the use of informal techniques for looking at data in various ways, before attempting any formal statistical analysis or modelling. Besides the use of summary statistics, the most important techniques used in this exploratory stage of data analysis are graphical methods. INSTAT has a number of commands for looking at data in various ways. This chapter describes these facilities in detail. Some require high resolution graphics and cannot be used on systems that do not have sufficient memory.

The first set of commands deals with ways of looking at variables (columns), one at a time. The other commands do two-way plots, and there are also facilities for plotting mathematical functions.

9.2 HISTOGRAMS

Note: the HIStogram command requires high resolution graphics.

The command HIStogram plots a histogram for the data contained in a column. If you enter the command

```
: HIS X3
```

then INSTAT makes its own choices of interval width and number of classes, and displays the histogram. The display appears in screen mode 1, unless you were in mode 0 when the HISt command was entered, in which case the display will also be in mode 0.

You have the option of choosing your own interval width by using the sub-command WIDth. For example,

```
: HIS X3; WID 20
```

displays a histogram with a class width of 20 units.
Alternatively, you can choose the number of classes. For this, use the NUMber sub-command:

```
: HIS X3; NUM 12
```

Note that the sub-commands WIDth and NUMber cannot be used together.

It sometimes happens that the actual number of classes displayed is not exactly what you specified with NUM. This occurs because

INSTAT tries to find round numbers for the class end-points, and one or more of the end classes may turn out to be empty.

You can inspect a part of a histogram by specifying the range of values covered by the x-axis. This is done with the XAXis sub-command. For instance,

```
: HIS X3; XAX 60 180
```

reproduces the portion of the histogram for which values of X3 lie between 60 and 180, 'blown up' to fill the whole screen. In some cases, the actual values that you specify for the x-axis may be rounded out to 'nice' numbers.

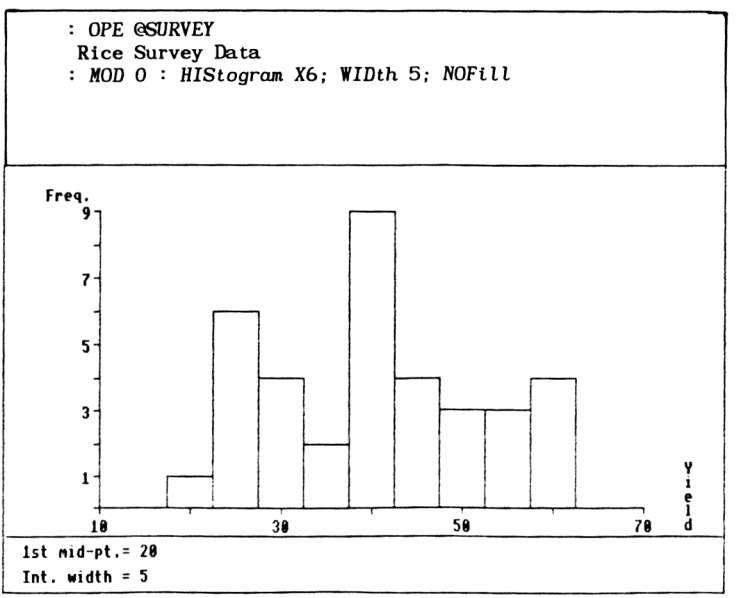
To save the frequency distribution calculated by the HISt command, there are two sub-commands - MIDpoints and FREquencies. The command

```
: HIS X3; MID X8; FRE X9
```

displays the histogram, and saves the class mid-points in the worksheet in X8 and the frequencies in X9.

Some dot-matrix printers object to printing large areas of black, so a sub-command NOFill has been provided to display the histogram as an outline only.

Figure 9.1 An Example of an INSTAT Histogram



9.3 STEM AND LEAF PLOTS

'Exploratory data analysis' (EDA) consists of a set of techniques that derive from a particular approach to the exploratory phase of data analysis due to J.W. Tukey [1977]. These techniques are especially good at highlighting unexpected features of data, such as locating outliers. They are therefore useful tools for looking at the residuals from regression or analysis of variance. INSTAT has commands for two of the most popular of these techniques and these are described in this and the next section. Since this User Guide is not a handbook of statistical methods, we make no attempt here to explain the meaning and interpretation of these plots, but instead refer the reader to Tukey's book or the more accessible account of EDA by Velleman and Hoaglin [1981].

The STEm command produces a stem and leaf plot for the data contained in a column. To get a stem and leaf plot for a single column, say X4, the command is

```
: STEM X4
```

You can also produce stem and leaf plots for several columns at once:

```
: STE X2-X4 X1
```

will display plots for X2, X3, X4 and X1.

The default option for this command 'trims' extreme values, if any, and notes them on the display as LO and HI. You have the choice of not trimming the data, however, by means of the NOTrim sub-command. Thus

```
: STE X4; NOT
```

produces an untrimmed stem and leaf plot.

The STEm command has another sub-command which may be useful for teaching purposes. This is HIStogram, which displays the plot in a form similar to a histogram, just to point out the connection between the two forms of display. Examples based on the survey data are shown in Figure 9.2.

Figure 9.2 Examples of Stem and Leaf Plots

```
: MODE 3
: OPEn @SURVEY
 Rice Survey Data
: STEmandleaf X6

  STEM & LEAF DISPLAY FOR Yield
  Min=19  Max=62  No. obs.=36
 62 is represented by 6:2  2

    1    1:5-9  9
    3    2:0-4  44
    9    2:5-9  556779
   12    3:0-4  013
   16    3:5-9  6778
   (8)   4:0-4  00012224
   12    4:5-9  5689
    8    5:0-4  03
    6    5:5-9  6789
    2    6:0-4  12

: STE X6; HIStogram ;PRI

  STEM & LEAF DISPLAY FOR Yield
  Min=19  Max=62  No. obs.=36
 62 is represented by 6:2  2

    1    1:5-9  X
    3    2:0-4  XX
    9    2:5-9  XXXXXX
   12    3:0-4  XXX
   16    3:5-9  XXXX
   (8)   4:0-4  XXXXXXXX
   12    4:5-9  XXXX
    8    5:0-4  XX
    6    5:5-9  XXXX
    2    6:0-4  XX
```

9.4 BOXPLOTS

The other EDA technique included in INSTAT is the BOXplot command. This provides a useful graphical summary of the data in a column (see Velleman and Hoaglin [1981] for a description of the display). The command for a boxplot of data in a single column, say X2, is simply

```
: BOX X2
```

Note that the amount of detail in the plot depends on the screen mode. An 80 column mode gives more detail than 40 columns. If you are sending the output to a printer, this point is worth remembering, because it is the boxplot as it appears on the screen that is reproduced on the printer.

Just as with the STEm command, you can have boxplots of several columns at once. For example,

```
: BOX X2-X4 X1
```

The default summary information displayed with the plot includes the extremes (labelled L and H) and the median (M). You can omit these, if you wish, by using the NOSummary sub-command:

```
: BOX X3; NOS
```

The display can be reduced to a one line plot by using the LINe sub-command:

```
: BOX X3; LIN
```

There is also an option to display a boxplot with 'notches'. This is EDA jargon for a confidence interval for the median, and the sub-command is NOT. So to produce a 'notched' boxplot of X3, the command is

```
: BOX X3; NOT
```

Sometimes the data in a single column are grouped or classified according to the levels of a factor, and it is then of interest to examine the data separately for each group, or factor level. There is a sub-command BY which accomplishes this. For example if X7 is a factor column (it must have the same length as X3), the command

```
: BOX X3; BY X7
```

displays a separate boxplot for the data corresponding to each level of the factor. This sub-command can be used very effectively together with the NOT and LIN sub-commands.

Examples of boxplots for the survey data are shown in Figure 9.3.

Figure 9.3 Examples of INSTAT's Boxplots.

```
       : OPE @SURVEY
       Rice Survey Data
       : BOXplot X6


BOXPLOT FOR Yield
                 ----------------------------
 ----------------(              +             )--------------------
                 ----------------------------
 L:::::::::::::::::::::::::::::M:::::::::::::::::::::::::::::::H
 19.1                          40.5                            62.1

      :
      :
      :
      :
      :
      :
      : BOX X6; BY X5


BOXPLOT FOR Yield BY Variety
                                                          -------
1                                                        -(   + )-
                                                          -------
                                 --------------
2                    ------------(      +       )--------------
                                 --------------
            ---------------
3  ---------(  +           )-----------
            ---------------

  Range of obs. :19.1 - 62.1
```

9.5 SCATTER PLOTS

There are two commands in INSTAT for doing scatter plots of one variable against another. One of these is PLOt (see Section 9.6), which is only available on systems with enough memory for high resolution graphics, and the other is SCAtter, available on all systems, which is described in this section. The SCAtter command uses ordinary characters and the plot can be sent to a printer in just the same way as text. An example of the basic use of the SCAtter command is

```
: SCA X3 X5
```

which produces a scatter plot of X3 against X5.

The output of the SCAtter command includes some information, along the axes, about the marginal distributions of the two variables. Below the x-axis, and to the left of the y-axis, are displayed the minimum, maximum and median values of the x- and y-variables, respectively. Along the axes themselves are numbers representing the frequencies of occurrence of the corresponding values of the variables (see the example in Figure 9.4). The scatter plot is therefore very informative - it contains information not only about the relationship between the two variables, but also about their marginal distributions.

As an alternative to having the minimum, maximum and median values displayed along the axes, you can use the LETter sub-command to show a 'letter value' display. This is another idea from EDA. The letters displayed are M for median, H for 'hinges' and E for 'eighths'. See Velleman and Hoaglin [1981] for an explanation.

Two other sub-commands, WIDth and HEIght, allow the control of the overall size of the plot. These are intended primarily for use with a printer, and with them you can produce a scatter plot with considerably more detail than is possible on the screen. This is especially true if you want to take advantage of the greater number of characters per line available in 'condensed' mode, such as with Epson printers. Note that you can send the appropriate control codes to the printer using VDU commands during an INSTAT session - for example,

```
: VDU 2,1,15
: SCA X3 X5; WID 120; HEI 50
```

sets an Epson printer in condensed mode and then prints a large scatter plot on the printer. Note that using the WIDth and HEIght sub-commands will probably produce a mess on the screen. Figure 9.4 presents some scatter plots for some of the survey data.

Figure 9.4 Use of SCAtterplot Command

```
        : OPE @SURVEY
        Rice Survey Data
        : SCAtterplot X6 X4

        PLOT OF Yield v Fert.
Yield
 62.1 2                                          +         +
      1                                          +         +
      3                                          +         2
      1                                                    +
      :
      3                                                    +         2
      2                    +         +
      1                              +
      5                              2         2                     +
40.55 3          +                              2
      3 +                                       2
      1                    +
      2 +                              +
      1 +
      5 3         +                    +
      2 2
      :
 19.1 1 +

        9---------2---------2---------6--------8---------6---------3
        0                              M=1.5                       3
                                      Fert.
```

```
        : NOTE X7 is a factor, so y-axis is set to 3 levels.
        :
        : SCA X7 X6; HEIght 6

        PLOT OF X7 v Yield
X7
 3 *                          + ++ ++ 2        2 +   +    ++++  ++
   :
 2 *          + +    +   +         ++ +  + 2
   :
 1 9 +      ++++ + +  +       +

     1------112111-1-11--1---1111-22-3--1-2---2-1---1----1111--11
   19.1                            M=40.55                   62.1
                                  Yield
```

9.6 THE PLOT COMMAND 1: SIMPLE USAGE

Note: this command requires high resolution graphics.

The PLOt command can produce scatter or line plots of data, plots of mathematical functions, or simultaneous plots of both data and functions. Its simplest usage is exemplified by

```
: PLO X3 X5
```

which produces a scatter plot of X3 against X5. The graph will appear in screen mode 1, unless you were in mode 0 when the PLOt command was executed, in which case the plot will also be in mode 0.

Used in this way, the PLOt command chooses its own scaling of the x- and y-ranges. There are options, however, for you to choose your own axis ranges. The sub-commands XAXis and YAXis enable you to set the ranges of the axes. For example,

```
: PLO X3 X5; XAX 10 25
```

restricts the x-range to values between 10 and 25. Note that the actual range plotted may not be quite what you specified because the program rounds the values to 'nice' numbers suitable for plotting. But the actual range displayed will always include the range that you specified. The YAXis sub-command is used in a similar way. These sub-commands can be used to effectively 'zoom in' on an interesting section of a plot. In most cases only the x-axis need be specified; the program then determines the necessary y-range automatically.

Any graph produced by PLOt can be given a title by using the TITle sub-command. There are two ways of doing this: either the title can first be stored in a string variable in the worksheet, or it can be specified directly with the sub-command. For instance,

```
: PLO X3 X5; TIT "A SCATTER PLOT"
```

produces the same result as

```
: ENT S1
S1: A SCATTER PLOT
: PLO X3 X5; TIT S1
```

Note: if you want lower case letters in the title, then you have to use the latter method, saving the title in a string.

INSTAT 'remembers' the last plot executed by the PLOt command for the duration of the session, and this may be re-executed using the REPlot command. The advantage of this is that new sub-commands may be added to modify the plot. This facility is particularly useful for examining a section of a plot using the XAXis or YAXis sub-commands. For example, after

```
: PLO X3 X5
```

you can do

```
: REP; XAX 10 25
```

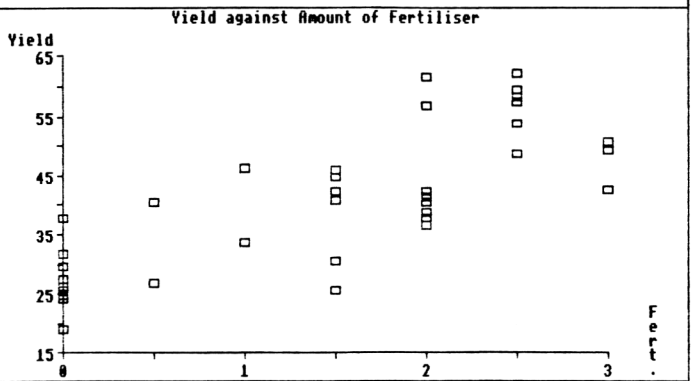
Another example is to look at the same plot in a different screen mode:

```
: MOD 0 : REP
```

In Figure 9.5, the scatter plot of yield against amount of fertiliser in the survey data, obtained in Figure 9.4 by the SCAtter command, is reproduced using the PLOT command.

Figure 9.5 Plot of Rice Survey data

```
: MOD 0
: OPE @SURVEY
Rice Survey Data

: ENT S1
S1: Yield against Amount of Fertiliser

: PLOt X6 v X4 ;TITle S1
```



9.7 THE PLOT COMMAND 2: SYMBOLS AND LINES

Used as described in the previous section, the PLOt command will choose its own symbol for the points in a scatter plot. You can, however, choose a different symbol from a list of 10 possible shapes. These are shown in the Reference Manual. You can also choose the size and colour of the symbol. All of this is achieved with the SYMbol command. This command must be entered before executing the PLOt (or REPlot) command. An example:

```
: SYM X3 4 2 1
: PLO X3 X5
```

The column referred to on the SYMbol command is the 'y-variable' to be plotted. This must be followed by three numbers: the first is the symbol code (1 to 10), the second is the size (an integer from 1 to 6) and the last is the colour code (1, 2 or 3). See the Reference Manual for details.

A line plot is more suitable for certain types of data. Line plots can be obtained with the PLOt command by first entering the LINe command. This has a similar syntax to the SYMbol command:

```
: LIN X3 1 2 2
: PLO X3 X5
```

The first of the three numbers must be 1 if you want a line plot, and should otherwise be 0. The second number (1, 2 or 3) is the line style and the last number is the colour code. Note that line style 3 is not really suitable for data plots and is intended for plotting functions (see Section 8.9). Again, see the Reference Manual for details.

After executing the LINe command, the data will be plotted without symbols, but they can be added to the line plot by also entering the SYMbol command. For example,

```
: LIN X3 1 2 2 : SYM X3 4 2 1
: PLO X3 X5
```

will produce a line plot with symbols as well.

To revert to a plot with symbols only, it is sufficient to 'switch off' the lines by entering

```
: LIN X3 0
: REP
```

So far we have been plotting one variable against another, but the PLOt command can just as easily plot several variables simultaneously against a common 'x-variable'. For example,

```
: PLO X2 X3 X4 X5
```

produces a scatter plot of X2, X3 and X4 against X5. The three

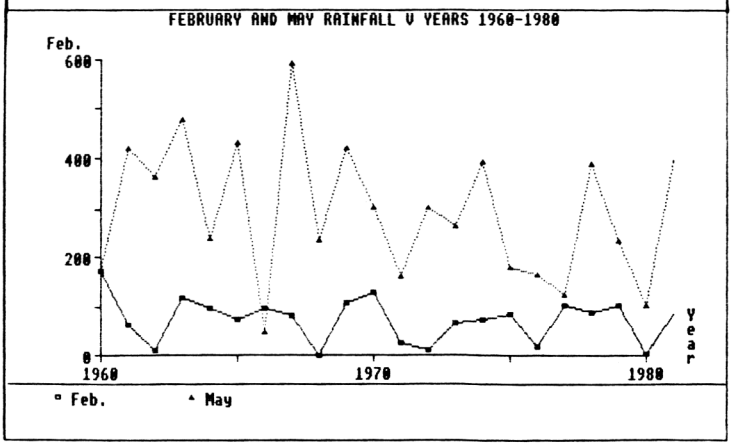
variables will automatically be plotted with different symbols. You can, of course, choose your own symbols separately for the variables by using a separate SYMbol command for each. Similarly, by using the LINe command, you can make some or all of the variables line plots, while others are symbols. For instance,

```
: LIN X2 1 1 1 : SYM X2 5 1 1 : SYM X3 2 2 2
: PLO X2 X3 X4 X5
```

will produce a line and symbol plot of X2, simultaneously with a scatter plot of X3 (with your own choice of symbol), and a scatter plot of X4, with the program's choice of symbol. An example based on the rainfall data is shown in Figure 9.6.

Figure 9.6 Plot of Monthly Rainfall Data

```
: MODe 0
: OPE @RAIN
 Monthly rainfall totals for 1950-83
: LINe X3 1 1 1 : LIN X6 1 2 2
: SYMbol X3 1 1 3 : SYM X6 2 1 3
: PLOt X3 X6 v X1 ;TITle "FEBRUARY AND MAY RAINFALL V
  YEARS 1960-1980"; XAX 1960 1980
```



9.8 THE PLOT COMMAND 3: PLOTTING GROUPED DATA

There are facilities for plotting data which are grouped or classified according to levels of a factor so that the different groups are distinguishable from each other. This is accomplished with the BY sub-command for the PLOt or REPlot commands. If, for example, X7 contains a factor with 4 levels, then

```
: PLO X3 X5; BY X7
```

produces a scatter plot of X3 against X5, in which the four groups are represented by four different symbols. If you want line plots, then use the LINe command first:

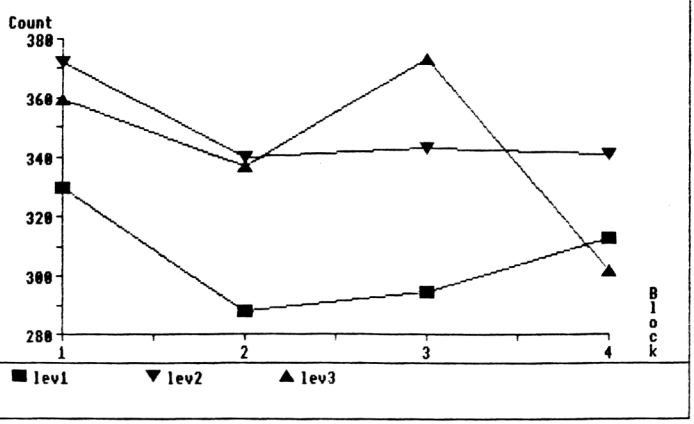
```
: LIN X3 1 1 2 : PLO X3 X5; BY X7
```

and the graph will consist of separate lines for the four groups. Figure 9.7 shows an example taken from the experimental data.

Another way of distinguishing between groups is to plot the data with symbols of different sizes for different levels of the factor. This may be appropriate when the factor levels in some sense represent the relative importance of the groups, and you may want to 'weight' the plot accordingly. This can be done with the WEIght sub-command (for PLOt or REPlot). The syntax is similar to the BY sub-command.

Figure 9.7

```
: OPEn @EXPERI
 Randomised block design from M&C pp52
: LINe X1 1 1 2 : SYMbol X1 8 3 1
: PLOt X1 X2 ;BY X3
```



9.9 THE PLOT COMMAND 4: PLOTTING FUNCTIONS

A mathematical function (of a single variable) can be stored as the contents of a string variable in an INSTAT worksheet. The rules for constructing such a formula are:

(1) It must be a valid BBC BASIC expression (upper or lower case);
(2) The variable must be represented by X (or x);
(3) Constants K1, K2, ..., stored in the worksheet may appear in the expression.

Note that you cannot use INSTAT functions (MEAN, SUM, etc.) and you must not use column names X1, X2, etc. in the expression. Some examples of valid expressions are:

```
sin(x)/x
(X -K3)*EXP(-0.5*X)
1.34 + 2.2*x + 0.17*x*x
```

The ENTer command (see Chapter 7) is used to put the formula into a string variable, and its graph can then be produced by PLOt. This is one instance where the XAXis sub-command is actually necessary in order to specify the range of x-values over which to plot the function. For example,

```
: ENT S4
S4: sin(x)/x
: PLO S4; XAX -5 5
```

It is possible to plot a function over an x-range specified by the data in a column:

```
: PLO S2 X5
```

plots the graph of the function in S2 between the minimum and maximum of the data in X5.

The line style and colour for a function plot can be chosen by using the LINe command, just as with data plots, except that instead of specifying a column, you specify a string variable - `: LIN S2 1 2 2` , for instance.

Several functions can be plotted simultaneously:

```
: PLO S1-S4; XAX 0 15
```

The line styles will cycle through the three possibilities for the four functions in this example. So S4 will have the same line style as S1.

A mixture of functions and data can also be plotted together. This is particularly useful for plotting data together with the graphs of fitted models. Some examples are

```
: PLO X3 S2 X5
```

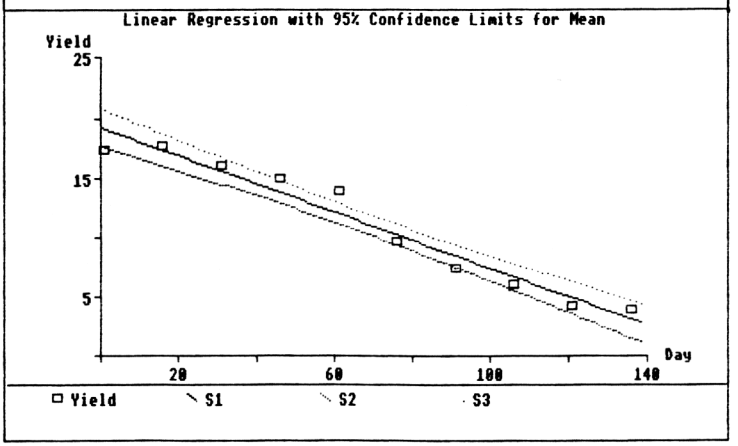
```
: PLO X2 X3 S4 S5 X5; XAX 0 15
```

You can experiment with various combinations of symbols, lines, etc. to produce a suitable graph. The REPlot command is very useful in this kind of experimention with graphs.

The final example, Figure 9.8, is based on a simple linear regression for our regression data (not a good model, in fact, and used here just for illustration). The plot shows the data, the fitted line and 95% confidence bands for the mean. We defer the explanation of how we arrive at the formulae in the strings S1-S3 until Chapter 13.

Figure 9.8 Plot of Regression Data

```
: open @regress
Regression Example
: ent s5
S5: Linear Regression with 95% Confidence Limits for Mean
:
: dis s1-s3
  S1 : 19.222-0.11799*x
  S2 : 19.222-0.11799*x-k4*sqr(k3*(0.1+((x-k1)^2)/k2))
  S3 : 19.222-0.11799*x+k4*sqr(k3*(0.1+((x-k1)^2)/k2))
: plot x2 s1-s3 x1; title s5
```



Chapter 10: DATA SUMMARY

10.1 INTRODUCTION

Descriptive statistics, or summary statistics, are invariably required during the initial exploratory phase of data analysis, preferably in conjunction with graphical methods. In some studies no further analysis may be warranted. Data summary is also frequently needed in the final stages when results are to be presented in a report.

The DEScribe and STATistics commands have been devised to provide summary statistics for one or more columns of data. The use of these commands is illustrated in subsequant sections of this chapter. There are other commands, besides DEScribe and STAtistics, which also provide some summary information as part of their output. For example, the SCAtterplot command always gives the minimum, median and maximum of the columns used in the plot. The STEm-and-leaf and BOXplots commands also give summary statistics. Various summary statistics can be calculated for data which have been cross-tabulated by several factors using the TABle command. Descriptions of these other commands and their facilities for data summary are to be found in other chapters of this User Guide, and also in the Reference Manual.

In addition to these facilities, there are a number of functions available with the CALculate command, which give certain summary statistics for a column. We begin with a brief description of these.

10.2 SUMMARY STATISTICS WITH THE CALCULATE COMMAND

There are a number of functions that can be used with CALculate to produce scalars (i.e. single numbers) from columns. The full list of available functions is given in the Reference Manual under CALculate. Figure 10.1 gives simple examples of the use of some of these functions. They can equally well be used with the '?' (or SHOw) command, which displays the results without saving them in the worksheet. For instance,

```
: ? mean(x5)
: ? 100*sde(x5)/mean(x5)  (the coefficient of variation of X5)
```

and so on. It is possible to use the flexibility of the CALculate command to calculate other statistics for which no function has been provided. In Chapter 14, for instance, there are examples which produce chi-squared and Kolmogorov-Smirnov statistics for testing goodness of fit.

Figure 10.1 Summary Statistics using the Calculate Command

```
: ENTer X1
data  1: 10 15 17 12 14 3 11
data  8: EOD
: ENT X2
data  1: 1 2 4 7
data  5: EOD
: K1=MEAn(X1)
: K2=MEAn(X1(2]5))       (Just uses elements 2,3,4 and 5)
: K3=MEAn(X1(X2))        (Just uses elements 1,2,4 and 7)
: K4=MAX(X1)-MIN(X1)
: K5=SDE(X1)
: K6=SQR(CSS(X1)/(COU(X1)-1)) (Should be the same as K5)

: DISplay K1-K6
  K1 =       11.714
  K2 =         14.5
  K3 =           12
  K4 =           14
  K5 =       4.5356
  K6 =       4.5356              (Phew - it is!)

: X3=(X1-MEA(X1))/SDE(X1)
: X4=CUS(X1-MEA(X1))
: DIS X1 X3 X4

Row          X1          X3          X4

  1          10    -0.37796     -1.7143
  2          15     0.72443      1.5714
  3          17      1.1654      6.8571
  4          12   6.2994E-2      7.1429
  5          14     0.50395      9.4286
  6           3     -1.9213     0.71429
  7          11    -0.15749   3.7253E-8
```

10.3 USING THE DESCRIBE COMMAND

The DEScribe command displays a range of summary statistics but does not provide facilities for them to be stored. Figure 10.2 illustrates its use on one month of the rainfall data introduced in Chapter 3.

If it is used without sub-commands, only a few statistics are output. The ;PERcentile sub-command gives specified percentage points, while the remaining sub-commands give a range of other statistics. As explained in Chapter 5, the ;HELp facility is always a valid sub-command and may be useful as a reminder of

Figure 10.2

```
: OPE @RAIN
Monthly rainfall totals for 1950-83

: DEScribe X5

Column                  April

 No. of observations    32
 Minimum                18.701
 Maximum                421.3
 Range                  402.6
 Mean                   217.29
 Std. deviation         106.44

: DES X5; MEDian; PERcents 20 50 80

Column                  April

 No. of observations    32
 Minimum                18.701
 Maximum                421.3
 Range                  402.6
 Mean                   217.29
 Std. deviation         106.44
 Median                 204.33
  20th percentile   =    115.11
  50th percentile   =    204.33
  80th percentile   =    328.98

: DES X5; HELP
Sub-commands are: PER MED LQU UQU IQU STE SKE KUR CSS USS COE
   ALL HEL
sub: ALL

Column                  April

 No. of observations    32
 Minimum                18.701
 Maximum                421.3
 Range                  402.6
 Mean                   217.29
 Std. deviation         106.44
 Median                 204.33
 Lower Quartile         156.27
 Upper Quartile         283.89
 Inter Quartile Range   127.62
 Std. Error of Mean     18.817
 Skewness               4.2479E-2
 Kurtosis               -0.59477
 Corrected SSQ          3.5124E5
 Uncorrected SSQ        1.8622E6
 Coeff. of Variation    49.0%
```

the various alternatives. The sub-command ;ALL is identical to giving all the subcommands except the PERcentage points.

The formulae used for each statistic in the DEScribe command are presented in Figure 10.3, together with a small data set that has been used as an example.

Figure 10.3 Formulae used for summary statistics.

Figure 10.3a Example Data and Notation

Data	Example	Ordered Data	Example
x_1	10	$x_{(1)}$	3
x_2	15	$x_{(2)}$	10
.	17	.	11
.	12	.	12
.	14	.	14
.	3	.	15
x_n	11	$x_{(n)}$	17

Figure 10.3b Formulae

Command/Statistic	Formulae (or equivalent statistic)	Example
Sample size	n	7
Minimum	$x_{(1)}$	3
Maximum	$x_{(n)}$	17
Range	$x_{(n)}-x_{(1)}$	14
Mean	$\bar{x} = \Sigma \frac{x}{n}$	11.71
Standard Deviation	$s = \sqrt{\frac{\Sigma(x-\bar{x})^2}{(n-1)}}$	4.54
Pth percentile	P*(n+1)/100 th observation $x_{(1)}$ if P*(n+1)/100 < 1, $x_{(n)}$ if P*(n+1)/100 > n	
MEDian	P_{50}	12
LQUartile	P_{25}	10
UQUartile	P_{75}	15
IQUartile	$P_{75}-P_{25}$	5
STError	$s/\sqrt{n}$	1.71
SKEwness	$\frac{\sqrt{n}\ \Sigma(x-\bar{x})^3}{(\Sigma(x-\bar{x})^2)^{3/2}}$	-0.911
KURtosis	$\frac{n\Sigma(x-\bar{x})^4}{(\Sigma(x-x)^2)^2} - 3$	0.08
CSS (corrected ssq)	$\Sigma(x-\bar{x})^2$	123.4
USS (uncorrected ssq)	Σx^2	1084.0
COEfficient of variation (only given if x > 0)	$(s / \bar{x}) * 100\%$	38.7%

10.4 THE STATISTICS COMMAND

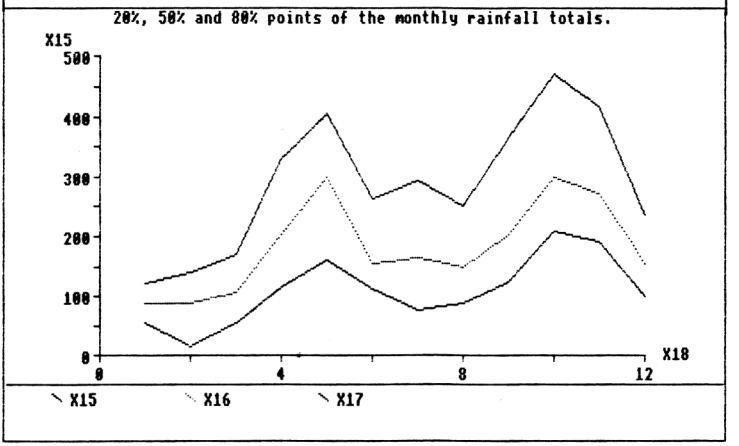
Summary statistics are often required, not merely as an end product, but to be used in further analyses or presentations. The STATistics command is a powerful facility for this purpose. Figure 10.4 gives an example using the full 32 year record of the monthly rainfall data, introduced in Chapter 3. Because there are 12 columns, X2 - X13, the derived columns of percentage points are of length 12.

Figure 10.4 The STAtistics Command

```
: OPEN @RAIN
Monthly rainfall totals for 1950-83
: STAT X2-X13; PERcent 20 into X15; MEDian X16; PER 80 X17

 Column       20%        Median      80%
              X15        X16         X17

 Jan.         56.92      90.16       122.7
 Feb.         15.75      87.73         141
 March        54.35      107.9       169.3
 April        115.1      204.3         329
 May          162.5        300       404.4
 June         112.8      155.8       262.8
 July         76.44      163.3       292.7
 August       90.17      149.7       250.6
 Sept.        126.3      203.2       364.8
 Oct.         208.3      298.3       472.8
 Nov.         191.4      273.9       416.5
 Dec.         101.6        154       237.3

: ENT X18
data  1: (1]12)
data 13:
: MODe 3
: DIS X15-X18

 Row          X15          X16          X17          X18

   1        56.92        90.16       122.71            1
   2        15.75       87.728       141.01            2
   3       54.346       107.94       169.28            3
   4       115.11       204.33       328.98            4
   5       162.46       299.97        404.4            5
   6       112.82       155.82       262.81            6
   7       76.444       163.32       292.67            7
   8       90.168       149.74       250.58            8
   9       126.34       203.24        364.8            9
  10       208.28       298.34       472.83           10
  11        191.4       273.94       416.48           11
  12        101.6       154.03       237.29           12
```

Fig. 10.4 cont'd

```
: LINe X15 1 1 1 : LIN X16 1 2 3 : LIN X17 1 1 1
: PLOt X15-X17 against X18;TITle "20%, 50% and 80% points
    of the monthly rainfall totals."
```



Data are often grouped or classified and interest frequently focusses on comparisons among the different groups. We have already encountered some of INSTAT's facilities for making graphical comparisons among groups in the previous chapter. The PLOt and BOXplot commands each have a sub-command 'BY' for this purpose. The same sub-command is available with the STAtistics command. If a factor column is specified with BY, then the result consists of the requested summary statistics separately for each level of the classifying factor. As an example, Figure 10.5 shows the count and mean yields of rice in the survey data, subdivided by each variety. These are 'one-way' tables. Survey data are often summarised by both one-way and multiway tables. The facilities for multiway tables use the TABle and PREsent commands and are introduced in the next chapter.

Figure 10.5 STAtistics - A One-way Table

```
: OPEn @SURVEY
Rice Survey Data
: STAtistics X6; BY X5; COUnt to X8; MEAn to X9

   Statistics for Yield

Variety      Count       Mean
Levels          X8         X9
  1              4       59.6
  2             17      45.44
  3             15         30
```

Chapter 11: ANALYSING SURVEY DATA

11.1 INTRODUCTION

The production of multiway tables is an important component of the analysis of many surveys. The small survey of rice yields introduced in Chapter 3 is used to illustrate the tabulation facilities within INSTAT. There are two commands concerned with tabulation. One command (TABle) does the work of computing the table and stores the result in columns so that further analysis may be done, if required, on the tabulated data. The other command (PREsent) does the job of displaying (and printing) tables in a way that is clear and informative (we hope!).

Each margin of a table corresponds to the different levels of a factor column. Some columns may first have to be RECoded to qualify as factors. The contents of a table may be simply the counts of the number of observations at each combination of factor levels. Alternatively, as mentioned in the previous chapter, it is possible to display summary statistics in tables. In the rice survey it may be useful to know the mean yield of farmers subdivided by each village and variety as well as the count or percentage in each category.

11.2 THE PRESENT COMMAND

Although the analysis of survey data is likely to be the most important area of application of tables, it is by no means the only one. The example in Figure 11.1 shows a year of daily water balance displayed using the PREsent command. In this example, each column was of length 366 and the two factors were 'month' with 12 levels and 'day of the month' with 31 levels. The command to produce the output was

```
: PRE X22; ROW X24; COL X23; FIX 1; WIDth 8; PRInterwidth 120
```

The ROW and COL sub-commands specify the factor columns to be used in defining the table. The remaining sub-commands control the format of the output: FIXed 1 prints all data to 1 decimal place; WIDth 8 defines the field width for the entries in the table; PRInterwidth 120 specifies the width of the display on the printer.

There are more examples of the use of the PREsent command in this Chapter, and the Reference Manual should be consulted for a complete description.

Figure 11.1 Use of the PREsent command.

Table for WBAL40: Month BY Day

Month / Day	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1	0.0	0.0	0.0	0.0	1.7	2.5	2.9	81.0	100.0	69.5	6.6	0.0
2	0.0	0.0	0.0	0.0	0.0	13.5	0.0	76.0	95.0	64.5	1.6	0.0
3	0.0	0.0	0.0	0.0	0.0	8.5	0.0	78.9	90.0	59.5	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	3.5	0.0	100.0	93.6	54.5	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	13.7	16.1	96.8	88.6	55.1	0.0	0.0
6	0.0	0.0	0.0	0.0	12.8	8.7	25.3	100.0	100.0	50.1	0.0	0.0
7	0.0	0.0	0.0	0.0	7.8	4.5	20.3	100.0	95.5	45.1	0.0	0.0
8	0.0	0.0	0.0	0.0	6.8	0.0	23.4	100.0	98.6	40.1	0.0	0.0
9	0.0	0.0	0.0	0.0	1.8	0.0	41.8	95.0	93.6	35.1	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	36.8	90.0	90.7	30.3	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	15.6	31.8	100.0	85.7	25.3	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	10.6	27.6	95.0	90.8	20.3	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0	24.6	26.9	95.8	85.8	15.3	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	32.1	25.9	90.8	82.9	10.3	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0	27.1	20.9	85.8	100.0	5.3	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	22.1	24.8	80.8	100.0	0.3	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	23.2	19.8	81.9	95.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	18.2	40.0	77.4	90.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	32.5	41.3	84.9	100.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	27.5	36.3	79.9	95.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	22.5	31.3	75.9	90.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	41.8	26.3	70.9	85.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	6.7	36.8	34.8	80.1	80.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0	1.7	31.8	31.1	97.5	75.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0	0.0	26.8	42.8	97.8	79.7	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0	0.0	27.9	69.6	92.8	83.8	0.0	0.0	0.0
27	0.0	0.0	0.0	21.7	6.7	22.9	65.6	87.8	78.8	0.0	0.0	0.0
28	0.0	0.0	0.0	16.7	1.7	17.9	67.5	82.8	73.8	4.1	0.0	0.0
29	0.0	0.0	0.0	11.7	17.5	12.9	64.5	77.8	79.5	0.0	0.0	0.0
30	0.0		0.0	6.7	12.5	7.9	60.0	72.8	74.5	16.6	0.0	0.0
31	0.0		0.0		7.5		78.4	84.8		11.6		0.0

11.3 THE TABLE COMMAND

This discussion is based on the rice survey data introduced in Chapter 3. The input of the data for the survey was considered in Chapter 7. There were six columns of data as follows:

X1 Village

X2 Field

X3 Size of field

X4 Quantity of fertilizer

X5 Variety

X6 Yield

X1 and X5 were designated as factor columns with 4 and 3 levels respectively. Labels for the levels of the factor were previously entered into L3 and L1. In Chapter 7 the method of recoding the fertilizer data was described. The derived column, X7, was designated as a factor column with 3 levels and the corresponding labels were in L2.

The command

```
: TABle X1 X5
```

results in the output shown in Figure 11.2, which gives the number of observations within each category.

Figure 11.2 Two-way table of counts - Village by Variety

```
: TABle X1 X5

 Villa  Varie   COUNT
 LEVEL  LEVEL

Sabey   New        2
Sabey   Old        5
Sabey   Trad       3
Kesen   New        0
Kesen   Old        3
Kesen   Trad       4
Niko    New        0
Niko    Old        2
Niko    Trad       3
Nanda   New        2
Nanda   Old        7
Nanda   Trad       5
```

Tables for variety (X5) by fertilizer (X7), or even the three-way table of village by variety by fertilizer (Figure 11.3) are produced in a similar way. With such a small survey many of the counts in the three-way table are, of course, zero.

Figure 11.3 A three-way table

```
: TAB X1 X5 X7

 Villa  Varie     X7    COUNT
 LEVEL  LEVEL  LEVEL

Sabey   New    Ocwt       0
Sabey   New    .5-2cw     0
Sabey   New    >2cwt      2
Sabey   Old    Ocwt       0
Sabey   Old    .5-2cw     1
Sabey   Old    >2cwt      4
Sabey   Trad   Ocwt       1
Sabey   Trad   .5-2cw     2
Sabey   Trad   >2cwt      0
Kesen   New    Ocwt       0
Kesen   New    .5-2cw     0
   .     .       .        .
   .     .       .        .
Nanda   Trad   .5-2cw     1
Nanda   Trad   >2cwt      3
```

The TABle command has a number of optional sub-commands. Their use, together with the subsequent use of the PREsent command, is illustrated in Figure 11.4. The sub-commands are explained in the Reference Manual, but the ASSociated sub-command deserves a mention here. Its effect in the example is to produce factor columns, X10 and X11, whose lengths are equal to the number of cells in the table. They are needed in the PREsent command for indexing the margins of the table, and they could also be used for further analysis of the table entries.

If your system is limited to a 40 column screen, you may wish to use the WIDth subcommand to set the number of characters for each field of the table. For example,

```
: PRE X8; COL X10; ROW X11; WID 6
```

Alternatively, if the results are to be printed the subcommand

```
...; PRINT 80
```

may be used. For example,

```
: PRE X8; COL X10; ROW X11; WID 12; PERcents; PRInt 80
```

produces a nice output on the printer, although it looks fairly incomprehensible on a 40 column screen.

Figure 11.4 Use of the TABle and PREsent commands.

```
: MODe 3
: OPEn @SURVEY
 Rice Survey Data
: TABle X5 X7; ASS X10 X11; COUnts X8; MEAns X6 X9

   Varie      X7    COUNT     MEAN
   LEVEL   LEVEL              Yield
   (X10)   (X11)     (X8)     (X9)

   New     0cwt         0     -9999
   New     .5-2cw       0     -9999
   New     >2cwt        4      59.6
   Old     0cwt         1      31.8
   Old     .5-2cw       6     43.32
   Old     >2cwt       10     48.08
   Trad    0cwt         8     26.88
   Trad    .5-2cw       4     29.25
   Trad    >2cwt        3     39.33

   : NOTE Data in X8-X11 are as follows..
   : DISplay X8-X11

   Row          X8          X9          X10          X11

     1           0       -9999            1            1
     2           0       -9999            1            2
     3           4        59.6            1            3
     4           1        31.8            2            1
     5           6      43.317            2            2
     6          10       48.08            2            3
     7           8      26.875            3            1
     8           4       29.25            3            2
     9           3      39.333            3            3
```

Fig. 11.4 cont'd

```
: PREsent X9; MISsing "xxx"; FIX 1

 Table for X9: Variety BY X7

                    X9
------------------------
Varie      X7 -----------
  New    0cwt :     xxx
       .5-2c :     xxx
       >2cwt :    59.6
  Old    0cwt :    31.8
       .5-2c :    43.3
       >2cwt :    48.1
 Trad    0cwt :    26.9
       .5-2c :    29.3
       >2cwt :    39.3

: NOTE  Now use ASSociated counts to specify layout
: PRE X9; COL X10; ROW X11; MIS "xxx"

 Table for X9: X10 BY X11

  X10       New     Old    Trad
-----------------------------
  X11 ----------------------
 0cwt :     xxx    31.8  26.875
.5-2c :     xxx  43.317   29.25
>2cwt :    59.6   48.08  39.333

: PRE X8; COL X10; ROW X11; PER of X10

 Percentage Table for X8: X10 BY X11

  X10       New     Old    Trad     MAR
-----------------------------------
  X11 ----------------------------
 0cwt :    0.00    5.88   53.33   25.00
.5-2c :    0.00   35.29   26.67   27.78
>2cwt :  100.00   58.82   20.00   47.22
  MAR :  100.00  100.00  100.00  100.00

 Total count = 36
```

Chapter 12: ANALYSING DATA FROM DESIGNED EXPERIMENTS

12.1 INTRODUCTION

Data arising from designed experiments are usually analysed by analysis of variance techniques. Most statistics packages for micros (and even some popular mainframe packages) can do analysis of variance for only the simplest designs, while there are, of course, mainframe programs, notably GENSTAT [1983], which have very flexible facilities for designed experiments.

The ANOva command in INSTAT can be used to analyse data from most of the common designs. These include randomized block, Latin square and split plot. The treatments may be unstructured or may be factorial combinations. There are also facilities for saving columns of mean values, if required, for more detailed study. The residuals can also be saved and plotted, or put into tabular form, to help assess whether the implied model is adequate.

INSTAT's current facilities for the analysis of variance have a number of limitations. There are no automatic facilities to handle missing values (a 'macro' for missing values is described in Chapter 16). Furthermore, the algorithm cannot cope with unbalanced designs (except for simple one-way designs). However, many unbalanced designs can be analysed by means of INSTAT's regression facilities together with the INDicator command. This is described in Chapter 13.

The next section considers the analysis of the simple experimental data introduced in Chapter 5. Further examples are then used to illustrate the additional steps necessary to analyse a factorial experiment, a split plot design and a simple confounded design.

12.2 SIMPLE USE OF THE ANOVA COMMAND

Figure 12.1 presents the analysis for the data in @.EXPERI, supplied on the master disc. The data entry is described in Chapter 7. The first steps in the analysis are to declare the 'blocks' and 'treatments' columns to be factors, and to declare the y-variate (or response variable) to be analysed. The ANOva command specifies which factor columns are to be used in the analysis. The procedure for all designs is broadly the same. The structure of the design is specified by the way in which the factor columns are set up (as later examples will show), but the three steps of declaring the factors, declaring the y-variate and then using the ANOva command are common to the analysis of all designs.

Figure 12.1 ANOva of Randomised Block design

```
: OPEn @EXPERI
Randomised Block Design (M&C. pp52)

: FACtor X2 4 X3 3
: YVAriate  X1
: ANOva X2 X3

          ANOVA TABLE

Source   DF      SS        MS         F
-----------------------------------------
Block     3    2330.2    776.75      2.0
Treat     2    4212.5    2106.2      5.4
Error     6    2321.5    386.92
-----------------------------------------
Total    11    8864.25
-----------------------------------------

      MAIN EFFECTS
      ------------


        Block                Treat
    Level    Mean        Level    Mean

        1  353.667           1  306.500
        2  321.667           2  349.000
        3  337.000           3  342.750
        4  318.667

 SE.diff.   16.061   SE.diff.   13.909
```

Although trivial in this case, it is important to note that the ANOva command is one of very few commands where the order of the arguments is important. They correspond to the successive lines in the Analysis of Variance Table. Before doing the analysis, it is therefore useful for you to visualise the structure of the ANOva table that is to be produced.

It is always a good idea to look at the residuals to check the adequacy of the model that you have fitted. Many textbooks tend to overlook this step, although the same textbooks may make quite an issue of examining residuals in regression. In both ANOVA and regression, however, we are fitting a linear model to data by the method of least squares with the same underlying assumptions, and these assumptions need to be checked. The sub-command

...;RESids to X4

saves the residuals in the designated column X4. The residuals that are then saved in X4 are standardised residuals - i.e. the differences between the observed and fitted values divided by the square root of the error mean square. In Figure 12.2, the residuals are saved and then displayed in tabular form by means of the PREsent command.

Figure 12.2 Table of Residuals

```
: ANOva X2 X3; RESiduals X4

          ANOVA TABLE

     (as in Figure 12.1)

: NAMe X4 'Resid

: PREsent X4; ROW 'Treat; COLumn 'Block; FIX 2

 Table for Resid: Block BY Treat

 Block        1      2      3      4
 ---------------------------------------
 Treat ----------------------------------
     1 :   0.13  -0.38  -0.80   1.05
     2 :   0.11   0.11  -0.52   0.31
     3 :  -0.24   0.27   1.32  -1.36
```

A number of scalar quantities are automatically saved in memory by the ANOva command. They remain in memory for the duration of the current INSTAT session, or until overwritten by another ANOva or by a regression command, which saves the same quantities (see Chapter 13). The quantities saved are:

	Sum of squares	Degrees of Freedom
Fitted	FSS	FDF%
Error	ESS	EDF%
Total	TSS	TDF%

Note: the 'fitted' sum of squares is often called the 'treatment' or 'regression' sum of squares.

These constants may be used with the CALculate command. For

example, the residual standard deviation may be calculated and saved by

```
: k1 = sqr(ess/edf%)
```

This could be used, for example, to calculate confidence limits for means.

12.3 ONE-WAY ANALYSIS OF VARIANCE

The ANOva command can, in principle, be used for the analysis of variance of a simple one-way completely randomised design, but such designs would then be restricted to balanced ones - i.e. with the same number of observations in each treatment group. The command ONEway does not suffer from this restriction and should be used instead. An example of its use in an analysis of part of the survey data is presented in Figure 12.3.

Figure 12.3 ONEway ANOVA for Rice Survey Data

```
: OPEN @.SURVEY
 Rice Survey Data

: yvar  X6
: ONEway X5

          ANOVA TABLE
Source   DF      SS         MS          F
-----------------------------------------
Variety   2    3527.8     1763.9      40.8
Error    33    1426.9      43.24
-----------------------------------------
Total    35   4954.74
-----------------------------------------

       MAIN EFFECTS
       ------------
        Variety
   Level    Mean       Count      S.E.
     New   59.600         4       3.29
     Old   45.441        17       1.59
    Trad   30.000        15        1.7
```

12.4 INTERACTION COLUMNS

Many of the designs that the ANOva command is capable of analysing need interaction terms in the underlying model. In INSTAT, interactions are derived from factor columns by the INTeraction

command. An example of its use is:

```
: INTeraction X3 X6 X10
```

Here, X3 and X6 would have been declared as factors, and the resulting interaction column is X10. An interaction column is really a special kind of factor column, but they are given a designation 'Inter.' in the output of the INFo; COLs command. Second and higher order interaction terms are derived by repeated applications of the INTeraction command. For instance, to continue with the same example, if we want to save the interaction between factors X3, X6 and X7 in X11, the above command would be followed by

```
: INTeraction X10 X7 X11
```

As another example of the use of the ONEway command, Figure 12.4 shows a way of analysing the interaction between villages and varieties in the rice survey data.

Figure 12.4 ONEway ANOVA

```
: OPEN @SURVEY
 Rice Survey Data
: YVAR X6
: INTeraction  X1 X5 X7
: ONEway  X7

            ANOVA TABLE
Source   DF      SS          MS         F
-----------------------------------------
X7        9      4125     458.34      14.4
Error    26    829.72     31.912
-----------------------------------------
Total    35   4954.74
-----------------------------------------

      MAIN EFFECT
      -----------
 Villa Varie   Mean     Count      S.E.
 Level Level Values
 Sabey New     59.5         2      3.99
 Sabey Old     49.2         5      2.53
 Sabey Trad    29.5         3      3.26
 Kesen New        0         0         0
 Kesen Old     43.3         3      3.26
 Kesen Trad    24.9         4      2.82
 Niko  New        0         0         0
 Niko  Old     36.1         2      3.99
 Niko  Trad    26.9         3      3.26
 Nanda New     59.7         2      3.99
 Nanda Old     46.4         7      2.14
 Nanda Trad    36.3         5      2.53
```

12.5 TEACHING THE ANALYSIS OF VARIANCE

Many students find the analysis of variance a difficult technique to understand. If necessary, once the simpler STAtistics command has been understood, it may be used to indicate where some of the numbers in the ANOVA table come from. Figure 12.5 shows how this may be done for an example using the rice survey data. The results may be compared with the output from the ONEway command given in Figure 12.3.

Figure 12.5 STAtistics command

```
: OPEn @SURVEY
 Rice Survey Data

: STAT X6; BY X5; COUnts to X13; MEAns to X14; FITted X15

    Statistics for Yield

  Variety      Count       Mean
  Levels         X13        X14
    1              4       59.6
    2             17      45.44
    3             15         30

: X16=X6-X15
: NOTE    DISplay X6 X5 X15 with width 7 to see
: NOTE    what has to be done.

: K1=SSQ(X16)                      (error sum of squares)
: K2=K1/(COU(X16) - COU(X14))      (error mean square)

: DISplay  K1 K2
    K1 =       1426.9
    K2 =        43.24
```

12.6 A FACTORIAL EXPERIMENT

Figure 12.6 shows the use of the ANOva command to analyse a 2*2*2 factorial experiment on water uptake in frogs and toads (Mead and Curnow [1983], pp 92-95). After the factor levels have been declared, the next step in setting up the data, is the INTeraction command, to specify columns which correspond to the two and three way interactions.

Once the data are set up, the ONEway command, : ONEway X8 , could alternatively be used to give an analysis, where the 8 different combinations of factor levels are just considered as 'treatments'.

Figure 12.6 ANOva of a 2*2*2 Factorial Experiment

```
: CREate @rept; COL 10 16; LAB 4 2
Enter title for worksheet (or RETURN).
ANOVA 2*2*2 Factorial Expt. (M&C pp92-95)
: mode 3
: ENTer 'Weight
data  1: 2.31 -1.59 17.68 25.23 28.37 14.16 28.39 27.94
data  9: .85 2.90 2.47 17.72 3.82 2.86 13.71 7.38
: ENT 'Spec.
data  1: 8(1]2)
: ENT 'Moist.
data  1: 2(1]2)4
: ENT 'Horm.
data  1: 4(1]2)2
: ENT L1
data  1: Toad Frog
: ENT L2
data  1: Wet Dry
: ENT L3
data  1: Control Hormone
: FAC 'S  L1
: FAC 'M  L2
: FAC 'H  L3
: DIS X1-X4; LAB

Row       Weight        Spec.       Moist.        Horm.

  1         2.31         Toad         Wet      Control
  2        -1.59         Toad         Wet      Control
  3        17.68         Toad         Dry      Control
  4        25.23         Toad         Dry      Control
  5        28.37         Toad         Wet      Hormone
  6        14.16         Toad         Wet      Hormone
  7        28.39         Toad         Dry      Hormone
  8        27.94         Toad         Dry      Hormone
  9         0.85         Frog         Wet      Control
 10          2.9         Frog         Wet      Control
 11         2.47         Frog         Dry      Control
 12        17.72         Frog         Dry      Control
 13         3.82         Frog         Wet      Hormone
 14         2.86         Frog         Wet      Hormone
 15        13.71         Frog         Dry      Hormone
 16         7.38         Frog         Dry      Hormone

: INT 'S 'M  X5
: INT 'S 'H  X6
: INT 'M 'H  X7
: INT 'S  X7 X8
: YVA 'Weight
: ANOva X2-X8
```

Fig. 12.6 cont'd

ANOVA TABLE

Source	DF	SS	MS	F
Spec.	1	515.06	515.06	14.9
Moist.	1	471.32	471.32	13.7
Horm.	1	218.01	218.01	6.3
Spec.Moi	1	39.501	39.501	1.1
Spec.Hor	1	165.12	165.12	4.8
Moist.Ho	1	57.836	57.836	1.7
S.M.H.	1	43.428	43.428	1.3
Error	8	276.05	34.506	
Total	15	1786.33		

MAIN EFFECTS

Spec. Level	Spec. Mean	Moist. Level	Moist. Mean	Horm. Level	Horm. Mean
Toad	17.811	Wet	6.710	Control	8.446
Frog	6.464	Dry	17.565	Hormone	15.829
SE.diff.	2.937	SE.diff.	2.937	SE.diff.	2.937

INTERACTIONS

Spec. Level	Moist Level	Mean Values
Toad	Wet	10.8
Toad	Dry	24.8
Frog	Wet	2.61
Frog	Dry	10.3
SE.diff		4.15

Spec. Level	Horm. Level	Mean Values
Toad	Contr	10.9
Toad	Hormo	24.7
Frog	Contr	5.98
Frog	Hormo	6.94
SE.diff		4.15

Fig. 12.6 cont'd

```
Moist Horm.   Mean
Level Level Values

Wet   Contr    1.12
Wet   Hormo    12.3
Dry   Contr    15.8
Dry   Hormo    19.4

     SE.diff   4.15


Spec. Moist Horm.   Mean
Level Level Level Values

Toad  Wet   Contr    0.36
Toad  Wet   Hormo    21.3
Toad  Dry   Contr    21.5
Toad  Dry   Hormo    28.2
Frog  Wet   Contr    1.87
Frog  Wet   Hormo    3.34
Frog  Dry   Contr    10.1
Frog  Dry   Hormo    10.5

           SE.diff   5.87
```

12.7 A SPLIT PLOT EXPERIMENT

Figure 12.7 gives the analysis of a split plot experiment on the yield of lettuce (Mead and Curnow [1983], pp 100-103). There were 4 blocks. The main plot treatments were three uncovering dates. They were split into six subplots for each of six varieties. In this analysis, the main plot error term is the interaction between blocks and the uncovering dates. The subcommand

```
....; ERRor X5
```

specifies it as an error term. For another example of the use of the ANOva command for a split plot design, see the Reference Manual under ANOva.

Figure 12.7 ANOva of a Split Plot Design

```
: OPEn @LETTUCE
Lettuce Yield (M&C, pp 100-103)
: INF; COL

   Worksheet filename: LETTUCE
Lettuce Yield (M&C, pp 100-103)

Cols.  Name Length Type  State
X1     ---- 72   Var(Y)  U
( X2 X3 X4 X5 X6 X7 X8 X9 X10 Free)

: ENT X2
data  1: (1]4)18
data 73:
: ENT 'MPLT
data  1: 24(1]3)
data 73:
: ENT 'SPLT
data  1:  4(1]6)3
data 73:
: FAC X2 4 X3 3 X4 6  :YVA X1
: INT X2 X3 X5  : INT X3 X4 X6
: ANO X2 X3 X5 X4 X6 ;ERRor X5

          ANOVA TABLE
Source    DF      SS        MS         F
-----------------------------------------
X2         3    29.343    9.7809      1.3
MPLT       2    38.003    19.002      2.6
ERROR 1    6    43.566    7.2609
-----------------------------------------
SPLT       5    260.51    52.102     10.3
MPLTSPLT  10     163.7     16.37      3.2
Error     45    227.28    5.0506
-----------------------------------------
Total     71   762.395
-----------------------------------------

      MAIN EFFECTS
      ------------
      X2                  MPLT                SPLT
  Level   Mean        Level   Mean        Level   Mean

      1  10.656           1  10.433           1   8.775
      2  10.950           2  11.075           2   8.608
      3  10.233           3   9.317           3   8.083
      4   9.261                               4  12.592
                                              5  12.767
                                              6  10.825

SE.diff.   0.898   SE.diff.    0.778   SE.diff.    0.917
```

Fig. 12.7 cont'd

INTERACTIONS

MPLT Level	SPLT Level	Mean Values
1	1	8.85
1	2	9.25
1	3	9.6
1	4	12.9
1	5	13.2
1	6	8.82
2	1	10.1
2	2	9.32
2	3	11.2
2	4	11.5
2	5	11.7
2	6	12.6
3	1	7.37
3	2	7.25
3	3	3.45
3	4	13.4
3	5	13.3
3	6	11.1
	SE.diff	1.59

NB: The S.E. for split plot interaction terms is for comparing split plot means within main plots only.

12.8 A 2^4 FACTORIAL EXPERIMENT WITH SIMPLE CONFOUNDING

Cochran and Cox [1957], pp 188-192, describe a fertilizer experiment on the yield of beans, which is confounded in blocks of size 8. The analysis of this set of data is given in Figure 12.8. No new concepts are involved, except to note the treatment effect which is confounded, namely the third order interaction, is not included as such, because its effect is already in the blocks term.

Figure 12.8 Analysis of a Simple Confounded Design

```
: OPEn @BEANS
 Example of confounded expt.
: INF; COL

      Worksheet filename: BEANS
  Example of confounded expt.

 Cols.  Name Length Type  State Pointers
 X1     Yield  32   Var(Y)  U
 X2     Block  32   Factor  U
 X3     D      32   Factor  U 3L L1
 X4     N      32   Factor  U 3L L2
 X5     P      32   Factor  U 4L L3
 X6     K      32   Factor  U 6L L4
 X7     D.N    32   Inter.  U 2L X3X4
 X8     D.P    32   Inter.  U 1L X3X5
 X9     N.P    32   Inter.  U 1L X4X5
 X10    D.K    32   Inter.  U    X3X6
 X11    N.K    32   Inter.  U    X4X6
 X12    P.K    32   Inter.  U    X5X6
 X13    D.N.P  32   Inter.  U    X7X5
 X14    D.N.K  32   Inter.  U    X7X6
 X15    D.P.K  32   Inter.  U    X8X6
 X16    N.P.K  32   Inter.  U    X9X6
 ( X17 X18 X19 X20 X21 X22 X23 X24 X25 Free)
```

Fig. 12.8 cont'd

```
: ANOva X2-X16

            ANOVA TABLE

Source   DF      SS        MS         F
-----------------------------------------
Block     3   126.37    42.125      1.7
D         1        2         2      0.1
N         1   325.12    325.12     13.4
P         1    6.125     6.125      0.3
K         1      4.5       4.5      0.2
D.N       1       32        32      1.3
D.P       1      242       242     10.0
N.P       1   78.125    78.125      3.2
D.K       1    6.125     6.125      0.3
N.K       1       32        32      1.3
P.K       1     24.5      24.5      1.0
D.N.P     1        2         2      0.1
D.N.K     1   10.125    10.125      0.4
D.P.K     1   15.125    15.125      0.6
N.P.K     1       32        32      1.3
Error    14   339.75    24.268
-----------------------------------------
Total    31  1277.87
-----------------------------------------

          MAIN EFFECTS
          ------------

   Block              D              N              P
Level   Mean     Level  Mean    Level  Mean    Level  Mean

    1  46.875     None 47.187     None 50.125     None 46.500
    2  47.625   10tons 46.687   0.4cwt 43.750   0.6cwt 47.375
    3  43.875
    4  49.375

SE.diff. 2.463  SE.diff. 1.742  SE.diff. 1.742  SE.diff. 1.742


       K
   Level   Mean

    None  47.312
  1.0cwt  46.562

 SE.diff.   1.742
```

Fig. 12.8 cont'd

```
     INTERACTIONS
     ------------

    D     N   Mean
Level Level Values

None  None     51.4
None  0.4cw      43
10ton None     48.9
10ton 0.4cw    44.5

     SE.diff   2.46


    D     P   Mean
Level Level Values

None  None     49.5
None  0.6cw    44.9
  .     .       .
  .     .       .

  .       .     .       .
  .       .     .       .


    N     P     K   Mean
Level Level Level Values

None  None  None      52.5
None  None  1.0cw       50
None  0.6cw None      46.5
None  0.6cw 1.0cw     51.5
0.4cw None  None        43
0.4cw None  1.0cw     40.5
0.4cw 0.6cw None      47.2
0.4cw 0.6cw 1.0cw     44.2

          SE.diff   3.48
```

Chapter 13: REGRESSION AND CORRELATION

13.1 INTRODUCTION

Regression facilities in INSTAT include the fitting of simple, multiple and polynomial regression models, together with commands for generating 'dummy variables' so that qualitative variables can be included in a model. The regression commands allow, and indeed encourage, interactive modelling of relationships between variables. This interactive approach to modelling is enhanced by the combined use of regression commands and INSTAT's data plotting facilities.

Before regression models can be fitted, there is a simple initialisation procedure consisting of two commands. One of these commands, TERMS, specfies which variables (columns) may be needed in the regression model. The other command, YVAR, tells the program which of these variables is to be the response variable or 'y-variate'. These two commands must be executed before attempting to do any regression analysis, but it does not matter which of the two commands comes first.

Usually, the first stage in exploring relationships between variables uses graphical techniques to get a preliminary idea of the kind of relationships that might exist. The plotting commands available in INSTAT are explained in Chapter 9. One of the reasons for this exploratory stage is to form a list of variables that may subsequently be included in a regression model. This list may include new variables formed by combining or transforming some of the original ones. It could include variables that may well not end up in the final regression model, but perhaps are thought sufficiently interesting to try anyway. Once this list of possible variables has been decided, it should be declared using the TERMS command. For example, suppose that the variables are X2, X4, X5, X8 and X9. The command is

```
: TERMS X2 X4 X5 X8 X9
```

Note that
(i) The order of the variables is immaterial;
(ii) The response variable must be in the TERMS list;
(iii) The columns must all have the same length.

Internally, what the TERMS command does is to compute the 'sums of squares and products' matrix (SSP matrix) used in subsequent regression calculations. The SSP matrix is stored in the worksheet and is referred to as V1. Space for an SSP matrix is automatically reserved by the CREate command. Using the sub-command ;SSP n will override this default size (see Chapter 6). Note that in this release of INSTAT, a worksheet cannot have more than one SSP matrix, but this restriction may be removed in a

future release.

Clearly, spurious results could be produced if it were possible to remove or alter any of the columns that have been used in computing the SSP matrix. To avoid this difficulty, INSTAT automatically 'locks' the columns that have been declared in TERMS so that they cannot be changed in any way. This locking remains in force until the SSP matrix is REMoved or a new one is formed. To remove the SSP matrix, you just enter the command

```
: REM V1
```

For an explanation of locking and unlocking, see Chapter 7.

The YVAR command is used to specify which of your variables is the response variable. Note that the y-variate must be included in the TERMS list. If it is X8, for example, the command is

```
: YVA X8
```

You can, if you wish, specify a different y-variate at any stage, provided the new one was also included in TERMS. This may be useful if you want to examine regression models separately for several response variables but with the same set of regressor or 'explanatory' variables. You just have to remember to include all of the response and explanatory variables in the TERMS list. In this way, the TERMS command need be executed only once.

13.2 CORRELATION

Before describing INSTAT's regression commands, let us first see how to obtain correlation coefficients. The correlation between any two columns of the same length can be calculated using the CORR command. For example, the correlation between X7 and X11 is given by the command

```
: CORR X7 X11
```

It is also possible to obtain a matrix of correlations for several variables at once, provided they have previously been declared in TERMS. The correlations between X4, X5, X8 and X9, for example, are given by

```
: CORR X4 X5 X8 X9
```

the matrix being displayed in lower triangular form. Note that if you just want correlations between variables two at a time, the TERMS command is not necessary.

Some users may find it useful to look at correlations as part of the preliminary exploratory phase of regression modelling. However, correlations or, for that matter, any other summary

statistics, are usually no substitute for graphical techniques at this stage. Graphs and scatter plots generally give you much more information about the nature of the relationship between variables, although it may sometimes be useful to supplement this information with correlation coefficients.

The CORR command calculates the usual product-moment coefficient of correlation. There is no single command for the coefficient of rank correlation, although it is easy to get INSTAT to calculate it by first using the RANK command to obtain the ranks of the two variables. The coefficient of rank correlation is then calculated as the ordinary correlation between the two columns of ranks. For example, to calculate the coefficient of rank correlation between X3 and X4, first save the ranks in, say, X11 and X12:

```
: rank x3 x11: rank x4 x12
```

and then

```
: cor x11 x12
```

produces the result.

13.3 SIMPLE LINEAR REGRESSION

The simplest regression model is a straight line regression of one variable on another. As described in Section 13.1, the two variables must first be declared in TERMS and the response variable declared with the YVAR command. The regression calculations are then done by the FIT command. For example, to fit a straight line regression model of X3 (the response variable) on X2 (the regressor variable), the command sequence is

```
: TER X2 X3
: YVA X3
: FIT X2
```

The output produced by the FIT command consists of the analysis of variance table for the regression, the F-ratio to test its significance and the value of the 'coefficient of determination', R-squared. An example is given in Figure 13.1.

You can obtain further information about the fitted model by using the ESTimate command. This gives ESTimates of the regression coefficients (sometimes called the 'parameter estimates'). They can optionally be saved in the worksheet by specifying a column with the command. Thus, EST on its own just displays the estimates, while

```
: EST X12
```

displays them and also saves them in X12. The output produced by

EST also gives the standard errors of the estimates and the corresponding t-values (the estimate divided by its standard error).

After a regression model has been fitted, a number of 'system constants' are stored in memory which can be used in further calculations during the same INSTAT session. The quantities stored, and their names, are:

Total corrected sum of squares of y-variate	= TSS
Residual ('error') sum of squares	= ESS
Regression ('fitted') sum of squares	= FSS
Total degrees of freedom	= TDF%
Residual ('error') degrees of freedom	= EDF%
Regression ('fitted') degrees of freedom	= FDF%

These constants can be used in calculations. For example,

```
: ? sqr(ess/edf%)
```

will calculate and display the residual standard deviation.

These quantities are not automatically stored in the worksheet, although their values remain in memory for the duration of the INSTAT session, or until the next FIT (or ADD or DROP) or ANOva command. However, their values can be saved in the worksheet as constants, if required. Thus, the command line

```
: k3=sqr(ess/edf%)
```

will save the residual standard deviation in K3.

The 'fitted values', or 'predicted values', are defined as the y-values derived from the regression equation corresponding to the x-values for the data points. They can be saved by specifying a column with the FVA sub-command to the FIT command.

An important feature of interactive regression modelling is the examination of residuals. Basically, residuals are the differences between observed y-values and the corresponding fitted values, usually standardised in some way. INSTAT calculates regression residuals in 'unit normal deviate' form (see Draper and Smith [1981], section 3.1, for a discussion of different types of residuals). This means that the differences between observed and fitted values are divided by the estimated residual standard deviation. These residuals can be saved by using the RES sub-command with FIT. Thus in the above example, if we wanted to save the fitted values and residuals in columns X8 and X9, the FIT command would be

```
: FIT X2; FVA X8; RES X9
```

We could then use graphical methods to examine the residuals to investigate the adequacy of the model. For example,

```
: SCA X9 X8
```

or, if your system supports INSTAT's high resolution graphics,

```
: PLOT X9 X8
```

If you want the usual significance tests and confidence intervals derived from regression calculations to be valid, then your examination of residuals should include some check on the normality of their distribution. A popular graphical technique for this is to plot the residuals against their 'normal scores'. Roughly, these are the expected values of the residuals if they really had come from a normal distribution. They can be calculated in INSTAT by the NORmalscores command. Thus, in our example, the following commands save the normal scores of the residuals in X10, and then plot the residuals against them:

```
: NOR X9 X10
: PLO X9 X10
```

This scatter plot should be close to a straight line if the residuals are approximately normally distributed.

If you are using INSTAT's high resolution graphics, then it is easy to obtain other interesting graphs associated with regression. We have already seen an example in Chapter 9, Figure 9.8. This graph is a plot of the data together with the fitted line and 95% confidence bands for the mean, based on a simple linear regression for the regression example introduced in Chapter 3.

Figure 13.1 presents the analysis for this example, including the calculations for the functions representing the confidence bands. The formula for the standard error used in calculating the confidence limits can be found, for example, in Mead and Curnow [1983], p. 132. The confidence limits are the appropriate t-value times the standard error on either side of the fitted line. The t-value is obtained in Figure 13.1 by the PERcentile command, which is explained in Chapter 14. Note that in entering the formulae into the strings, effective use can be made of the BBC's cursor control ('arrow') keys and the <COPY> key to avoid having to type the numbers manually.

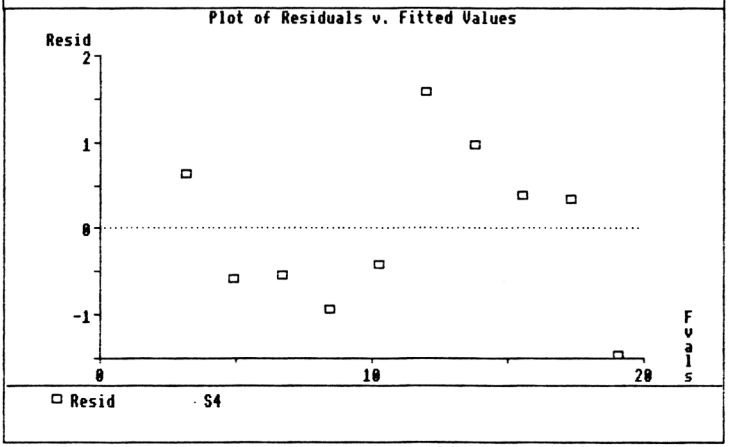
Figure 13.1 REGression Example

```
: open @regress
 Regression Example

: terms x1 x2
: yvar x2
: fit x1; fvals x9; resids x10

ANOVA for regression of Yield
on Day
---------------------------------------
Source      df      SS        MS
---------------------------------------
Regression   1  258.411     258.411
Residual     8  10.9258     1.36573
---------------------------------------
Total        9  269.337
---------------------------------------

Overall F = 189.211   R-squared = 0.9594

: k1 = mea(x1)
: k2 = css(x1)
: k3 = ess/edf%
: perc 97.5 k4; tdi 8
 t distribution with 8 d.f.
 97.5% point is        2.306
: est

REGRESSION COEFFICIENTS

Y-variate: Yield
---------------------------------------
Param.  Estimate   SE           t
---------------------------------------
Const   19.222     0.69412     27.69
Day     -0.11799   8.5776E-3   -13.76
---------------------------------------

: ent s1
S1: 19.222-0.11799*x
: ent s2
S2: 19.222-0.11799*x-k4*sqr(k3*(0.1+((x-k1) 2)/k2))
: ent s3
S3: 19.222-0.11799*x+k4*sqr(k3*(0.1+((x-k1) 2)/k2))
: plot x2 s1-s3 x1; title s5

  (The plot is as shown in Figure 9.8)
```

A plot of residuals against fitted values is shown in Figure 13.2, from which it appears that the straight line model is suspect, since there is a clear pattern in the residuals. The analysis of this example is continued in Section 13.6.

Figure 13.2 PLOT of Residuals v Fitted Values

```
: name x9 'Fvals x10 'Resids
: ent s4
S4: 0
: line s4 1 3 1
: plot x10 s4 x9; tit "Plot of Residuals v. Fitted Values"
```



13.4 MULTIPLE REGRESSION

The FIT command can be used to fit a multiple regression model just as easily as the simple linear regression described in Section 13.3. Remember that the response and explanatory variables should have been previously declared using the TERMS and YVAR commands, as explained in section 13.1. Then to fit a multiple regression of, say, X8 (the response variable) on the explanatory variables X2, X4 and X9, you could use the command

```
: FIT X2 X4 X9
```

If you wanted to save fitted values and residuals, then the FVA and RES sub-commands could be appended to the FIT command, as in Section 13.3. There is also a way of saving fitted values and residuals as an afterthought, after doing the regression. The command REFIT fits the current regression model, and displays the analysis of variance table for the model, and the FVA and RES sub-commands can be added to it. Thus, the command

```
: REF; RES X15; FVA X16
```

displays the same output as the above FIT command, but also saves the fitted values and residuals. Note that the worksheet 'remembers' the last regression model fitted, so the REFIT command will reproduce the last regression ANOVA even in a later INSTAT session.

By not specifying any columns, the FIT command on its own can be used to fit the simplest model of all, just a constant (the mean of the y-variate).

The 'system constants', described in the previous section, are available with multiple regression just as they are with simple regression.

To display and (optionally) save the parameter estimates after fitting a regression model, use the EST command exactly as described for simple regression in Section 13.3. The EST command has two sub-commands which were not mentioned in connection with simple regression. These are COV and COR, for displaying the estimated covariances and correlations, respectively, between parameter estimates. These are displayed in lower triangular form. For example,

```
: EST X21; COV
```

displays the estimates, their standard errors and t-values, saves the estimates in X21, and also displays their covariance matrix. Note that the diagonal elements of this matrix are the variances of the estimates. The COR sub-command is used in the same way.

A particular feature of INSTAT that encourages the interactive approach to regression modelling, mentioned in the introduction to this chapter, is the ability to add new terms to, and drop existing terms from a model. The command ADD includes one or more new terms in the current regression model. These new variables, of course, should have been declared in the TERMS list at the start. Similarly, the command DROP removes one or more terms from the current model. For example, if the current regression is of X8 on X2, X4 and X9, then

```
: ADD X5
```

produces the regression of X8 on X2, X4, X9 and X5. The output

from the ADD command consists of the new residual sum of squares and degrees of freedom, the change in regression sum of squares achieved by adding the new term(s) and the F-ratio for testing the effect of adding the new term(s), given the variables already in the model. This F-ratio thus provides a test of the amount of variation in the response that can be explained by the added variable(s), after accounting for variation due to the terms already in the model. This test is sometimes called a 'partial F-test' (Draper and Smith [1981], section 2.9).

Terms can be dropped from the current model one or more at a time using the DROP command. For instance, continuing the same example,

```
: DROP X4 X9
```

fits the regression of X8 on X2 and X5. The output from DROP is similar to that for ADD.

Neither ADD nor DROP displays the overall analysis of variance table for the new regression. This can easily be produced, however, by the REFIT command.

Example

An example of a multiple regression analysis with two explanatory variables is presented in Figure 13.3. The example is taken from Mead and Curnow [1983], pp 179-182. The response variable 'O2' is a measure of the production of oxygen in samples of water from the River Thames, 'Chlor' is the amount of chlorophyll and 'Light' the amount of light. The first step would be to plot the response against each of the other variables, and each explanatory variable against the other, but these have been omitted here.

Note the use of the PRObability command to obtain the significance level for the partial F-test. This command is explained in Chapter 14. It looks as if a reasonable model is simple linear regression on 'Light' only, with the regression equation of

$$y = -1.18 + 0.0106*x2.$$

The contribution from 'Chlor' is, however, substantial although not significant at the conventional 5% level. We might therefore prefer the model with both variables:

$$y = -1.34 + 0.0118*x1 + 0.0091*x2$$

One would normally proceed with graphical checks of residuals, etc., but this has been omitted here.

Figure 13.3 Multiple Regression Example

```
: open @thames
 Multiple Regr'n example - M&C p179
: dis x1-x3

Row        Chlor        Light          02

  1         33.8        329.5         2.16
  2         47.8        306.8         4.13
  3        100.7        374.7         2.84
  4        105.5        432.8         4.65
  5         33.4        222.9        -0.42
  6           27        352.1         1.32
  7           46        390.8         4.04
  8        139.5        232.6         1.97
  9           27        277.7         1.63
 10         22.5        358.5         1.16
 11         16.5          210         0.61
 12         71.3        361.8         1.94
 13         49.4        300.4          1.7
 14         19.3         96.9         0.21
 15         71.6        151.8         0.98
 16         13.4          126         6E-2
 17         11.8         67.8        -0.19

: ter x1-x3: yvar x3: fit 'Chlor

ANOVA for regression of 02
on Chlor
-----------------------------------------
Source       df       SS         MS
-----------------------------------------
Regression    1   10.9147      10.9147
Residual     15   25.9329      1.72886
-----------------------------------------
Total        16   36.8476
-----------------------------------------

Overall F = 6.313     R-squared = 0.2962

: add 'Light
Residual S.S.          = 12.4206  Residual d.f.          = 14
Increase in Reg.S.S. = 13.5124  Increase in Reg.d.f. = 1
F-ratio for change     = 15.231 on (1,14) d.f.
```

Fig. 13.3 cont'd

```
: fit 'Light

ANOVA for regression of O2
on Light
---------------------------------------
Source      df      SS        MS
---------------------------------------
Regression   1  21.8668     21.8668
Residual    15  14.9808     0.998718
---------------------------------------
Total       16  36.8476
---------------------------------------

Overall F = 21.895    R-squared = 0.5934

: add 'Chlor
Residual S.S.         = 12.4206  Residual d.f.         = 14
Increase in Reg.S.S. = 2.56022  Increase in Reg.d.f. = 1
F-ratio for change    = 2.886 on (1,14) d.f.

: prob 2.886; fdist 1 14
 F dist. with 1 and 14 d.f.
 Probability > 2.886 = 0.1115
: est
REGRESSION COEFFICIENTS

Y-variate: O2
---------------------------------------
Param.  Estimate   SE           t
---------------------------------------
Const   -1.3384    0.62892     -2.13
Chlor   1.1776E-2  6.9321E-3   1.70
Light   9.0772E-3  2.3259E-3   3.90
---------------------------------------

: drop 'Chlor
Residual S.S.         = 14.9808  Residual d.f.         = 15
Decrease in Reg.S.S. = 2.56022  Decrease in Reg.d.f. = 1
F-ratio for change    = 2.886 on (1,14) d.f.

: est
REGRESSION COEFFICIENTS

Y-variate: O2
---------------------------------------
Param.  Estimate   SE           t
---------------------------------------
Const   -1.1771    0.65963     -1.78
Light   1.0625E-2  2.2706E-3   4.68
---------------------------------------

: refit; fvals x7; resids x8     ..... etc.
```

13.5 FACTORS IN REGRESSION MODELS

Sometimes data are grouped or classified according to different levels of one or more factors. Factors have been discussed in connection with tables and analysis of variance in previous chapters. There are facilities in INSTAT for including factors as explanatory variables in multiple regression models. In this context it is useful to think of a factor as representing a 'qualitative variable', as opposed to variates which are quantitative variables.

The usual technique for fitting regression models with factors is to represent the different levels by 'dummy variables', also called indicator variables (see Draper and Smith [1981], section 5.4). The command INDicator generates a set of dummy variables from a factor. If, for example, X6 is a factor with 4 levels, the command

```
: IND X6 into X11-X14
```

saves the appropriate dummy variables in X11-X14. The first dummy variable, X11, takes the value 1 in rows for which X6 is 1 and is 0 otherwise; X12 is 1 or 0 according to whether or not X6 is 2, and so on.

If a factor has n levels, then (n-1) dummy variables are required to represent it in a regression model. So you would normally use all but one of the dummy variables generated by the INDicator command. In principle, the variable excluded could be any one, but a reasonable convention is to use all but the first. This set of dummy variables should then be included in your TERMS list, along with any quantitative variables that you may wish to fit. For example, suppose that we want to investigate regression models for X8 (the response) on quantitative variables X2, X4, X5 and X9 and a factor X6 with 4 levels. The sequence of commands to initialise the regression could be

```
: IND X6 X11-X14
: TER X2 X4 X5 X9 X8 X12-X14
: YVA X8
```

The effect of the factor X6 on the response can then be examined by fitting the group of dummy variables X12-X14. Note that you would normally fit the dummy variables corresponding to a factor together as a group. For example,

```
: ADD X12-X14
```

adds the factor X6 to the current model, but you would not usually want to add X12, for example, on its own. (It is not meaningless to do this, however. It has the same effect as recoding the original factor to a new factor with just 2 levels, corresponding to whether or not the original factor, X6, is at level 2.)

Factors in regression models have a number of useful applications.

Linear models which contain only factors are particularly useful for analysing data from 'unbalanced' experimental designs. Suitably balanced designs can be analysed more efficiently by the ANOva command (see Chapter 12), but by using the regression commands a much wider class of linear models can be fitted. There is a good discussion of these techniques in Chapter 9 of Draper and Smith [1981]. Another application is to what older statistical texts refer to as 'analysis of covariance' (see, for example, Snedecor and Cochran [1980], Chapter 18). A final important special case is the comparison of several regressions, testing for parallel lines etc.

An example of a comparison of regressions analysis, based on the rice survey data, is shown in Figure 13.4. A reasonable model turns out to be three parallel regression lines, one for each variety. The lines have slope 5.2643 (S.E. = 0.9555), and the intercepts of the three lines are 47.755, 35.686 (i.e. 47.755-12.069) and 25.764 (i.e. 47.755-21.791). The standard error of the first intercept is 3.216, but the standard errors of the other two need to be calculated from the covariance matrix, because they are sums of parameter estimates. For example, the standard error of the second one is

SQR(10.343 + 7.211 - 2*6.5366) = 2.1168

A plot of the fitted lines together with the data classified by variety is shown at the end of Figure 13.4. The analysis should continue with checks of residuals, but this has been omitted.

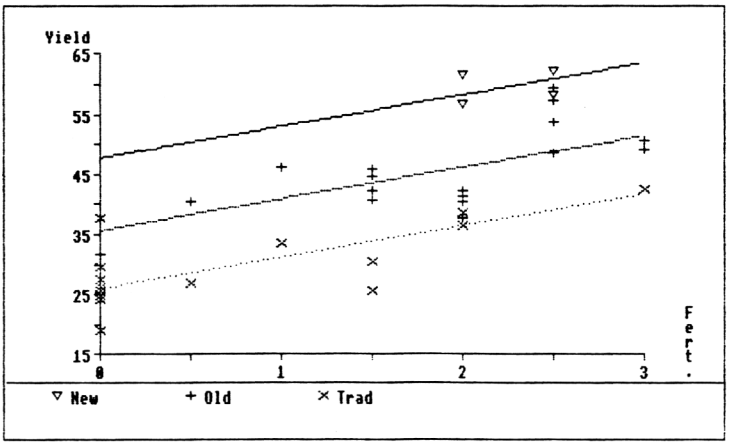
Figure 13.4 Comparison of Regressions

```
: OPEn @SURVEY
 Rice Survey Data
: NOTE  From Fig. 9.4, looks as if both variety and
: NOTE  fertilizer affect rice yield.
: INDicator X5 into X11-X13
: X14 = X4*X11 : X15 = X4*X12 : X16 = X4*X13
: NOTE  X14-X16 will be used for
: NOTE  fitting non-parallel lines
: TERms X4 X6 X12 X13 X14-X16 : YVAR X6
: NOTE  First try a single line
: FIT X4

 ANOVA for regression of Yield on Fert.
 ---------------------------------------
 Source       df       SS         MS
 ---------------------------------------
 Regression    1   2993.7       2993.7
 Residual     34   1961.04      57.6775
 ---------------------------------------
 Total        35   4954.74
 ---------------------------------------

 Overall F = 51.904   R-squared = 0.6042
```

Fig. 13.4 cont'd

```
: NOTE  Now try 3 parallel lines
: ADD X12 X13
Residual S.S.         = 732.283   Residual d.f.         = 32
Increase in Reg.S.S. = 1228.75   Increase in Reg.d.f. = 2
F-ratio for change    = 26.848  on (2,32) d.f.

: NOTE Seems useful - what about 3 separate lines?
: ADD X15 X16

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.

: NOTE  No evidence that they are needed,
: NOTE  so back to parallel lines
: DROP X15 X16

Residual S.S.         = 732.283   Residual d.f.         = 32
Decrease in Reg.S.S. = 14.2587   Decrease in Reg.d.f. = 2
F-ratio for change    = 0.298   on (2,30) d.f.

: ESTimates; COVars
REGRESSION COEFFICIENTS

Y-variate: Yield
---------------------------------------
Param.  Estimate   SE           t
---------------------------------------
Const   47.755     3.216        14.85
Fert.   5.2643     0.95549      5.51
X12     -12.069    2.6853       -4.49
X13     -21.791    3.0423       -7.16
---------------------------------------

VAR-COVAR MATRIX OF PAR. ESTIMATES

Const  10.343
X4     -2.0541    0.91296
X12    -6.5366    0.3625     7.211
X13    -8.768     1.3542     6.2587     9.2553
       Const      X4         X12        X13

: ENT S2
S2: 47.755+5.2643*X
: ENT S3
S3: 47.755-12.069+5.2643*X
: ENT S4
S4: 47.755-21.791+5.2643*X
: SYM X6 3 2 1 : PLOT X6 S2-S4 X4; BY X5
```

Fig. 13.4 cont'd



13.6 POLYNOMIAL REGRESSION

Polynomial regression models can be fitted by regarding them as a special case of multiple regression. If a plot of the response variable, y, against an explanatory variable, x, suggests that a polynomial regression might fit the data, you just have to create new variables representing the powers of x required, and then declare them in the TERMS list. The FIT, ADD and DROP commands can then be used to choose a suitable model.

A word of warning about this procedure: if the original x-values are large and a high order polynomial equation is to be fitted, it is likely that there will be some numerical rounding error in the regression calculations. It is generally safer to work with a new x-variable derived from the original one by means of a simple change of scale.

Once a polynomial equation has been fitted to the data, the fitted values can be saved and plotted, together with the observed data, if you wish. Alternatively, the equation can be saved in a string variable and plotted, together with the data (provided your system supports INSTAT's high resolution graphics).

It is possible to fit other types of equation to data by means of similar techniques. For instance, trigonometric functions of an

x-variable may be useful in fitting equations to periodic data.

Figure 13.5 shows the continuation of the analysis of the regression data, begun in Figures 13.1 and 13.2. We noticed that after fitting a straight line, there appeared to be some curvature in the residuals, suggesting that a polynomial regression might be appropriate. Again, residual plots have been omitted but they should be done.

Figure 13.5 (Continuation of analysis in Figures 13.1 and 13.2)

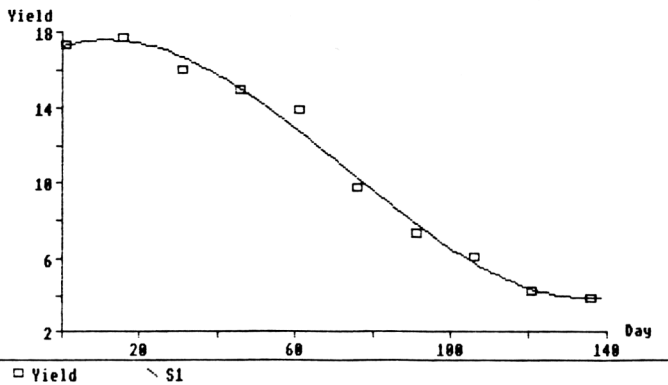
```
: NOTE  Calculate quadratic and cubic terms
: X6=X1*X1 : X7=X6*X1
: TERms X1 X2 X6 X7
: FIT X1

ANOVA for regression of Yield  on Day
-------------------------------------
Source      df      SS        MS
-------------------------------------
Regression   1  258.411     258.411
Residual     8  10.9258     1.36573
-------------------------------------
Total        9  269.337
-------------------------------------

Overall F = 189.211    R-squared = 0.9594

: ADD X6
Residual S.S.        = 9.86204   Residual d.f.         = 7
Increase in Reg.S.S. = 1.06381   Increase in Reg.d.f. = 1
F-ratio for change   = 0.755 on (1,7) d.f.

: ADD X7
Residual S.S.        = 2.33736    Residual d.f.         = 6
Increase in Reg.S.S. = 7.52468    Increase in Reg.d.f. = 1
F-ratio for change   = 19.316 on (1,6) d.f.

: NOTE  Cubic looks interesting, examine it further
: REFit ;FVAlues X8; RESids X9

ANOVA for regression of Yield  on Day X6 X7
-------------------------------------
Source      df      SS        MS
-------------------------------------
Regression   3  267         88.9999
Residual     6  2.33736     0.38956
-------------------------------------
Total        9  269.337
-------------------------------------

Overall F = 228.462      R-squared = 0.9913
```

Fig. 13.5 cont'd

```
: EST

REGRESSION COEFFICIENTS

Y-variate: Yield
```

Param.	Estimate	SE	t
Const	17.258	0.59657	28.93
Day	6.7E-2	3.9766E-2	1.68
X6	-3.2048E-3	6.9438E-4	-4.62
X7	1.4624E-5	3.3275E-6	4.39

```
: NOTE put fitted equation into string S1 for plotting
: ENT S1
S1: 17.258+x*(6.7E-2+x*(-3.2048E-3+x*1.4624E-5))
: PLOT X2 S1 X1
```



For teaching purposes, a nice demonstration of the folly of extrapolation from a polynomial model is to do the same plot over a wider x-range, e.g.

```
: replot; xaxis 0 200
```

Chapter 14: PROBABILITY DISTRIBUTIONS

14.1 INTRODUCTION

Statistical data analysis usually involves, although not always explicitly, an underlying probability model for the data. At some stage in the course of analysing a set of data it will almost always be desirable to compare the distributional properties of the data with properties of the corresponding theoretical distribution. INSTAT has a number of commands which are intended to facilitate such comparisons. These commands will also give significance levels for most of the common significance tests used in data analysis.

Another major use of these facilities is in teaching. All too often in teaching (or learning) statistics, the important statistical ideas associated with probability distributions are obscured by efforts to manipulate formulae and 'get the right answer'. These may be important skills to learn, but, at least in courses on applied statistics, should be given no more than secondary importance.

This chapter describes INSTAT's commands for obtaining probabilities, percentage points, etc for the common 'standard' distributions encountered in statistics, and demonstrates some techniques for checking how well a probability model fits a sample of data.

14.2 CALCULATING PROBABILITIES

The PRObability command can be used to find probabilities for the following distributions: normal, chi-square, t-distribution, F-distribution, gamma, binomial and Poisson. The simplest form of the command is

: *PROB x; sub-command*

where x is some number and the sub-command specifies the probability distribution. This command is one of the few in INSTAT which *needs* a sub-command in order to work - there is no 'default' distribution. For full details of the syntax of the sub-commands, see the Reference Manual.

The meaning of the result depends on whether the distribution is discrete or continuous. The only discrete distributions available are the binomial and Poisson. For a discrete distribution, the number x must be an integer (within the appropriate range of values), and the result is the probability of being equal to that

value. Here are some examples:

To find the probability that a Poisson variable with mean 4.5 is equal to 6:

```
: PROB 6; POI 4.5

 Poisson dist. with mean 4.5
 Probability of 6 = 0.1281
:
```

The probability of 5 'successes' in a binomial distribution with 12 trials and probability of success = 0.36 is given by

```
: PROB 6; BIN 12 0.36

 Bin. 12 trials. Success prob 0.36
 Probability of 5 = 0.2106
:
```

If the distribution is continuous, then the result is the probability of being greater than x. For example, to find the probability that a random variable with a normal distribution with mean = 10 and S.D. = 2.63 exceeds 12.5:

```
: PROB 12.5; NOR 10 2.63

 Normal dist. Mean 10 and s.d. 2.63
 Probability > 12.5 = 0.1709
:
```

Suppose that an analysis of variance gave an F-value of 2.9 on 3 and 12 degrees of freedom. The significance level is given by

```
: PROB 2.9; FDI 3 12

 F dist. with 3 and 12 d.f.
 Probability > 2.9 = 0.0788
:
```

The chi-square and t-distributions are handled in a similar way, but the gamma distribution deserves special attention. There are various equivalent ways of defining this two-parameter distribution. INSTAT specifies the distribution in terms of its mean (m) and the 'shape parameter' (k). With this choice of parameters, the density function is

$$f(x\ m,k) = \frac{1}{\Gamma(k)} \cdot \left[\frac{k}{m}\right]^k \cdot x^{(k-1)} \cdot \exp(-kx/m)$$

An important special case is k = 1, which corresponds to the exponential distribution with mean m. The chi-square distribution is another special case of the gamma disribution. The chi-square

distribution with n degrees of freedom is the same as the gamma distribution with m = n and k = n/2. For example, the two commands

: *PROB* 6.5; *CHI* 5 and : *PROB* 6.5; *GAM* 5 2.5

produce the same result (0.2606).

An alternative way of using the PRObability command is to replace the number x by a stored constant Kn. For instance, in the F-distribution example above, if K2 has value 2.9, we could get the same result by : *PROB K2*; *FDI* 3 12.

It is also possible to store the result of the probability calculation by specifying a constant Km as the second argument of the command. For example,

: *PROB* 2.9 *K3*; *FDI* 3 12

produces the same output on the screen as in the F-distribution example above, and saves the result (0.0788) in K3.

A useful feature of the PRObability command is that it is possible to obtain the probabilities for a whole set of values in one command. Instead of a number (x) or a constant Kn, the values can first be saved in a column which is then specified as the argument. This is illustrated in Figures 14.1 and 14.2.

Figure 14.1

```
: ent x3
data  1: (0]8)
data 10:
: prob x3; bin 8 0.3

 Bin. 8 trials. Success prob 0.3
 Value - x       Probability of x
         0              0.0576
         1              0.1977
         2              0.2965
         3              0.2541
         4              0.1361
         5              0.0467
         6              0.0100
         7              0.0012
         8              0.0001
```

Figure 14.2

```
: ent x5
data  1: (-2]2!0.5)
data 10:
: prob x5; nor 0 1

 Normal dist.   Mean 0 and s.d. 1
 Value - x        Probability > x
        -2               0.9772
      -1.5               0.9332
        -1               0.8413
      -0.5               0.6915
         0               0.5000
       0.5               0.3085
         1               0.1587
       1.5               0.0668
         2               0.0228
```

It may sometimes be helpful to save the probabilities in a column in the worksheet. This could be used for plotting graphs of distributions, for instance. The probabilities can be stored in Xm by simply specifying Xm as a second argument. In the binomial example in Figure 14.1, for instance, we could save the probabilities in X4 by executing

```
: pro x3 x4; bin 8 0.3
```

14.3 PERCENTAGE POINTS OF PROBABILITY DISTRIBUTIONS

The converse problem to that of finding a probability for a given distribution, is the problem of determining the value of a variate (with a known distribution), given the probability. The value is called a percentile, or percentage point of the distribution. One application of this is in hypothesis testing, where we may need the critical value of a test statistic for a specified significance level. Traditionally, tables are used for this purpose, but getting the computer to calculate percentage points has the advantage that intermediate values not given in tables are available, thus avoiding the need for interpolation. Another application of percentiles is the comparison of an empirical distribution with a theoretical one. A concise way of summarising such a comparison is to present the empirical percentiles with the corresponding theoretical ones, preferably as a graph (see the note on Probability Plots at the end of this section).

The INSTAT command for percentage points of continuous distributions is PERcentile. There are no such facilities for discrete distributions. The syntax of the command is very similar to the PRObability command. In particular, the distribution is specified by a sub-command, and a sub-command is necessary. The distributions available are the normal, chi-square, gamma, t- and F-distributions. Refer to the Reference Manual for details of the syntax.

Like the PRObability command, PERcentile can take a number p, a constant Kn or a column Xn. The value must be in the range 0% to 100%, and there will usually be further restrictions on the value, depending on the distribution. For example, the percentage point of a normal distribution corresponding to p = 100% is infinite, so 100% (and similarly, 0%) is not allowed. The result displayed is the value, x, of the variate such that the probability of not exceeding x is p/100, i.e. the pth percentile of the distribution. For example,

```
: PER 97.5; NOR 0 1
 Normal dist.  Mean 0 and s.d. 1
 97.5% point is          1.960
:
```

The percentage points can also be saved in a constant or, if the first argument is a column Xn, in another column Xm. Here are a few more examples:

```
: DIS K1
  K1 =          99.5

: PER K1 K2; NOR 0 1
 Normal dist.  Mean 0 and s.d. 1
 99.5% point is          2.576

: DIS K2
 K2 =        2.5758

: ENT X5
data  1: 1 5 10 90 95 99
data  7:
: PER X5; CHI 8
 Chi squared  dist. with 8 d.f.
 Percentage              Value
         1%              1.646
         5%              2.733
        10%              3.490
        90%             13.362
        95%             15.507
        99%             20.091
```

Application to Probability Plots

A quick and simple way of checking whether a sample of data can be assumed to come from a particular probability distribution is to plot the data, sorted in ascending order, against suitable percentiles of the theoretical distribution. In principle the percentiles should be the percentage points corresponding to percentages 100*i/n (i = 1,2,...,n), where n is the sample size. However, in order to avoid difficulties that may arise at the 100% point, it is customary to use the 100*i/(n+1) points. This adjustment makes little difference when n is not too small. A plot of the sorted data against the percentage points should not deviate too much from a straight line if the theoretical distribution is a reasonable model for the data. You might like to try the following experiment. The first command generates a pseudo-random sample (explained in the next Chapter) of size 25 from the exponential distribution with mean 2. The remaining commands produce a probability plot of the data against the same theoretical distribution.

```
: GEN 25 X2; EXP 2
: SORT X2 X2
: ENT X3
data  1: (1]25) <RET>
data 26: <RET>
: X3=100*X3/26
: PER X3 X4; GAM 2 1
: SCA X2 X4
```

14.4 SPECIAL FACILITIES FOR THE NORMAL DISTRIBUTION

The normal distribution is, of course, by far the most important one, not least because so many of the most widely used techniques of data analysis require normally distributed data or residuals. Although the PRObability and PERcentile commands can be used with the NORmal sub-command, the distribution has so many different applications that it has been thought useful to add some special facilities.

First, the normal distribution function is available as one of the functions that can be used in calculations with columns or constants. It is denoted NPR(x) and is defined as the probability that a normally distributed variate, with mean zero and variance one, is less than or equal to x. Some examples of its use are:

```
: ? NPR(1.96)
: X8 = NPR((X5-MEA(X5))/SDE(X5))
```

The inverse distribution function of the standard normal distribution can also be calculated. It is denoted NDE(x) (for Normal DEviate), where x must be between 0 and 1. For example:

```
: ? NDE(K3)
: X7 = NDE(X8)
```

Finally, normal probability plots are particularly easy to produce with the NORmalscores command. The 'normal scores' of a sample of values are defined as the expected values of the corresponding order statistics of a standard normal sample of the same size. The command would typically be used as follows:

```
: NOR X4 X5  (saves the normal scores of X4 in X5)
: SCA X4 X5  (could use PLOt, if available)
```

The resulting plot should roughly be a straight line if the data in X4 are normally distributed.

14.5 EXPECTED FREQUENCIES AND GOODNESS-OF-FIT TESTS

There are no commands in INSTAT for directly calculating goodness-of-fit statistics, but it is easy to use existing commands to perform the most commonly used tests. Although the PRObability command could be used to calculate expected frequencies, it is slightly more convenient to use the FREquencies command. Its syntax is very similar to PRObability, and in particular has the same sub-commands, so will be described here only briefly. The command

```
: FRE 60 X6 X8; GAM 2 1
```

calculates the expected frequencies, corresponding to the values in X6 for a sample of size 60, from the gamma distribution with mean 2 and shape parameter 1 (i.e. the exponential distribution with mean 2), and saves them in X8. If the second column, X8, is omitted, the results are displayed but not saved.

The command is likely to be most useful when X6 contains the upper class limits for a frequency distribution. Note that X8 will have length one greater than X6. X8(1) will then be the expected frequency of values less than or equal to X6(1), and the last value in X8 will be the expected frequency for values *greater* than the last class boundary in X6. The intermediate values X8(i), for i = 2,3, ..., are the expected frequencies corresponding to values less than or equal to X6(i) and greater than X6(i-1).

Example: A Chi-Squared Goodness-of-Fit Test

Suppose we have a sample of 60 values in X5 and we wish to test the hypothesis that they come from an exponential distribution with mean 2. The first step is to form a frequency distribution. The most direct way of achieving this is to use the HIStogram command with the FREquencies and MIDpoints sub-commands (see

Chapter 9). However, this method requires a system that supports INSTAT's high resolution graphics, so we also show how to get the same result using the TABle and RECode commands.

(a) Using HIStogram: first experiment with different histograms to find suitable class intervals. Then save the midpoints and frequencies. For instance,

```
: HIST X5; WID 0.5; MID X6; FRE X7
```

X6 then contains the class mid-points and X7 the (observed) frequencies. We shall need the upper class limits, so

```
: X6 = X6+0.25
```

(b) Using TABle and RECode: first choose a suitable class interval and starting value by examining the minimum and maximum values in X5 (use : ? *min*(x5) and : ? *max*(x5)). Suppose we choose classes 0-0.5, 0.5-1.0, ..., 5.0-5.5. Use the RECode command:

```
: RECODE X5 into X10
data : 0 0.5  1
data : 0.5 1  2
          .
          .
data : 5 5.5  11
```

To use the TABle command (see Chapter 11) to calculate the frequencies, the recoded column X10 must first be declared a factor:

```
: FAC X10 11
: TABLE X10; COUNTS X7
```

We also need the upper class boundaries in X6, which here have to be ENTered:

```
: ENT X6
data  1: (0.5]5.5!0.5)
data 12:
:
```

Whichever method you use, X7 should end up containing the frequencies and X6 the upper class limits. Remembering that the FREquencies command produces an expected frequency for values greater than the last upper class limit, we need to append a zero to the column of observed frequencies. There are several ways of doing this. Here is one:

```
: ENT X7
data  1: X7,0
data 13:
:
```

Now to calculate the expected frequencies, save them in X8 and calculate the chi-squared value:

```
: FRE 60 X6 X8; GAM 2 1
: X9 = (X7-X8) 2/X8
: DIS X6 X7 X8 X9 : ? SUM(X9)
```

If appropriate, you can now use the PROB command with the CHI sub-command to check the significance level.

Note that if you want to test the observed frequency distribution against other probability models, only the last commands above need to be changed.

Example: A Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov test statistic is defined as the maximum absolute difference between an observed cumulative distribution function and a theoretical one. If the statistic exceeds a critical value at a specified significance level, then we reject the hypothesis that the theoretical distribution is an adequate model for the data. Care is needed in using this test as there are circumstances in which the significance level may be incorrect, and the power of the test reduced. (See, for example, Pollard [1977].) However, some people find the test a useful alternative to the chi-squared test, especially when the sample size is small. Tables of percentage points of this statistic are to be found in Pollard [1977], or Siegel [1956]. In the example shown in Figure 14.3, we calculate the Kolmogorov-Smirnov statistic to test the hypothesis that the sample of 15 observations in X1 comes from a standard normal distribution.

Figure 14.3 An example of the Kolmogorov-Smirnov Test

```
: dis x1

X1
     -0.4        -0.77        -1.39         0.69        -0.19
    -0.47         0.66         1.84        -1.11         0.24
    -0.84         1.29         3.31        -0.72        -0.73
: sort x1 x2
: ent x3
data  1: (1]15)
data 16:
: warn: x3 = x3/15 : warn
: x4 = npr(x2)
: x5 = abs(x3-x4)
: ? max(x5)
  0.18876
```

The calculated value of the Kolmogorov-Smirnov test statistic is 0.18876, which is not significant even at the 20% level. Note that X3 contains the empirical distribution function and X4 the theoretical one for the standard normal distribution. Other distributions can be tested by using the PRObability command instead of using NPR. To derive the distribution function of the gamma distribution with m = 2 and k = 1.5, for instance, the commands would be:

```
: prob x2 x4; gam 2 1.5
: x4 = 1-x4
```

14.6 OTHER FACILITIES

To conclude this chapter, we make a brief mention of two of INSTAT's other features: the TINterval command, which produces t-tests and confidence intervals, and the GAMma command for fitting a gamma distribution. These two commands are straightforward to use and their output should be clear. For details of their syntax see the Reference Section of the User Guide.

Confidence Intervals and Tests Based on the t-Distribution

The TINterval command can be used to obtain confidence intervals for the mean of a sample and for the difference in means of two samples (which may be of different sizes). For example,

```
: TIN X3
```

gives a 95% confidence interval for the mean of X3, while

```
: TIN X3 X4
```

produces a 95% confidence interval for the difference in the means of X3 and X4. To change the confidence level, use the CON sub-command.

A t-test of the hypothesis that the mean of X3 is 4, say, is obtained by using the TEST sub-command:

```
: TIN X3; TEST 4
```

Similarly, we can test the hypothesis that the difference in means between X3 and X4 is some specified value. For instance,

```
: TIN X3 X4; TEST 0
```

provides a test of equality of the means of X3 and X4.

Of course, t-tests and confidence intervals are only really valid if the data are normally distributed. The methods described earlier in this Chapter can be used to check that this is so.

Fitting a Gamma Distribution

Estimates of the mean, m, and the shape parameter, k, of the gamma distribution can be obtained from the data in a column, X5 say, by

```
: GAM X5
```

The output calls the estimate of the mean 'muhat' and that of the shape parameter 'khat'. These estimates are maximum likelihood estimates of the parameters, but as an alternative, moment estimates can be calculated by using the MOM sub-command.

Chapter 15: RANDOM NUMBERS, SAMPLES AND PERMUTATIONS

15.1 INTRODUCTION

Pseudo-random numbers have many interesting applications. Much can be learned about the behaviour of a probabilistic model by getting the computer to simulate data according to the 'rules' of the model. Sometimes a theoretical analysis is too difficult to contemplate and simulation may then be the only way to study the model. Another application arises in data analysis. The previous chapter describes probability plotting as a means of checking whether data are consistent with some theoretical distribution, and usually we produce a plot which should 'roughly' be a straight line if the distribution fits the data. But how 'rough' can the line get before we start to have doubts? A good procedure is to generate several samples of pseudo-random data, with the same sample size as your data, and subject these samples to the same probability plot. In this way you can get a feel for just how much deviation from the 'ideal' straight line you can reasonably expect if the model were a good one.

Computer-generated random samples really come into their own in teaching. Although by no means a substitute for learning by analysing 'real' data, there is no better way of understanding the nature of sampling variation than by getting the computer to repeatedly generate samples for you.

Besides generating pseudo-random data, we also describe how INSTAT can select pseudo-random samples and produce pseudo-random permutations which can be used, for instance, in randomising the allocation of treatments in a designed experiment.

15.2 USING BBC BASIC'S RANDOM NUMBERS IN INSTAT

Columns of pseudo-random numbers can be obtained directly using the CALculate command (implicitly). We can simply exploit the property of CALculate that it can use functions from BBC BASIC. Thus for instance, the command

```
: X1 = RND(1)
```

generates pseudo-random numbers which are uniformly distributed on (0,1). The length of X1 is usually equal to the maximum column length of the worksheet, but it is possible to generate a shorter column by using the UNITS sub-command. For instance,

```
: X1 = RND(1); UNITS 20
```

generates a column of length 20 (provided the maximum column length is 20 or more).

This is a convenient way of generating pseudo-random integers (again uniformly distributed):

```
: X2 = RND(10)-1
```

generates random integers from 0 to 9, inclusive. See your BBC Micro's User Guide for more information about the RND function.

15.3 INSTAT'S PSEUDO-RANDOM NUMBER GENERATOR

The technique described in the preceding section can be used only for uniformly distributed random numbers (although some distributions require only a simple transformation of the uniform distribution). The command GENerate produces pseudo-random samples from a variety of distributions. The distribution is specified by a sub-command, and a full list of those currently available is given in the Reference Section. Here are some examples:

```
: GEN 25 X3          produces a sample of size 25 from the
                     uniform (0,1) distribution, and saves it in X3.

: GEN 40 X4; NOR 10 2   generates 40 observations from a normal
                        distribution with mean 10 and S.D. 2.

: gen 20 x5; poi 3.4    gives a sample of 20 integers from a
                        Poisson distribution with mean 3.4.
```

Several different samples from the same distribution can be generated simultaneously by simply specifying more than one column. So, for example, to produce 5 samples, each of size 60, from the exponential distribution with mean 2:

```
: gen 60 x1-x5; exp 2
```

This should be useful for the application concerning probability plotting mentioned in the Introduction to this chapter.

Teaching the Central Limit Theorem

A useful exercise for teaching the basic idea of the central limit theorem is to generate a large number of samples from some distribution, and examine the distribution of the sample means. Suppose, for instance, that we want 50 samples of size 5 from a normal distribution with mean 20 and variance 4. We could use the GENerate command to obtain 50 columns of length 5 and then calculate all 50 means with the STAtistic command. Alternatively, there is a well-known trick which can reduce the amount of work. We generate just 5 samples, each of size 50, and take the *rows*

as our 50 samples of size 5. Of course, we have to have the foresight to set up the worksheet to accomodate columns of sufficient length. The following commands produce the samples and save the means in X6.

```
: gen 50 x1-x5; nor 20 2
: x6 = (x1+x2+x3+x4+x5)/5
```

We can now examine the sampling distribution of the means in X6 using the HIStogram, or STEm, command, and calculate its mean and standard deviation etc.

15.4 RANDOM SAMPLES AND PERMUTATIONS

The GENerate command has a sub-command SAMple which can be used to extract random samples of data from a column in the worksheet. It can also be used to produce randomisations of a label column.

Sampling Without Replacement

Suppose X1 contains data from which we want to draw samples. The command

```
: GEN 8 X10; SAM X1
```

selects a pseudo-random sample without replacement from X1 (which must have length at least 8) and puts it into X10. Several independent samples can be taken simultaneously:

```
: GEN 8 X10-X14; SAM X1
```

Of course, 'without replacement' refers to each sample separately. *Different* samples may contain the same elements of X1.

Sampling With Replacement

In this case we have to specify the probabilities with which each element of X1 will be sampled. This is done by saving the probabilities in a column, say X2, and specifying it as a second argument to the SAMple sub-command:

```
: GEN 8 X10-X14; SAM X1 X2
```

In this case, there is no restriction on the length of X1.

Random Permutations

Random permutations are invaluable aids in planning the allocation of treatments, blocks, etc. in designed experiments. Like the

percentage points of distributions, random permutations are traditionally obtained from tables, but it is convenient to have them available from the computer. To produce a random permutation of the integers 1, 2, ..., 12, say, in X5, the command is

```
: GEN 12 X5; PER
```

or, again, if several independent permutations are required,

```
: GEN 12 X5-X9; PER
```

Another use of the SAMple sub-command is to obtain randomised permutations of the labels in a label column. Figure 15.1 illustrates the idea. L1 contains labels representing the 8 possible treatment combinations for a 2*2*2 complete factorial experiment. Suppose that there are to be 3 blocks, so 3 independent randomisations are required. The randomisations are contained in label columns L2-L4.

Figure 15.1

```
: GEN 8 L2-L4; SAM L1
: DIS L1-L4
```

Row	L1	L2	L3	L4
1	(1)	K	N	NPK
2	N	N	P	PK
3	P	NPK	K	NP
4	NP	P	NPK	P
5	K	(1)	(1)	K
6	NK	PK	NP	NK
7	PK	NP	PK	(1)
8	NPK	NK	NK	N

Chapter 16: MORE ADVANCED USE OF INSTAT

16.1 INTRODUCTION

This chapter introduces a few techniques which can be used to extend the facilities within INSTAT or to enable existing facilities to be used more easily. In particular, we discuss the use of some of the BBC micro's 'star' commands, INSTAT's system integers and facilities for writing 'macros' of stored INSTAT commands.

16.2 USING STAR COMMANDS

These are the commands provided by the BBC's disc system. The most useful are likely to be

*CAT

*EXEC filename

*DRIVE n

*KEY .....

Programming the function keys, in the course of an INSTAT session, provides a simple way to save a small series of commands, that are then executed by pressing the one key. For example, typing

KEY 3 DIS X1-X4 |M DES X1-X4 |M

programs these commands into function key f3.

*CAT, or just *., displays the directory of the current disc drive, and *DRIVE n changes the current drive to n. These commands, and several other star commands, behave as they usually do without INSTAT.

16.3 USING *EXEC

Figure 16.1 shows the commands and data supplied as the file EGS1 on the master disc to introduce INSTAT. To run the commands it is only necessary to enter INSTAT and then type

EXEC EGS1

Figure 16.1 Listing of file EGS1

```
NOTE  Only the first 3 letters of
NOTE  a command are required

WARn OFF supresses warnings
PAGe OFF for continuous scrolling
PAUse 2 secs between commands
CLOse any current Worksheet
CREate @WOODS
Simple example of a forest survey

NOTE  The CREate command reserves
NOTE  an area on the disc and
NOTE  calls the file '@.WOODS'

REAd X1 X2 X3 X4
11 110 163  273
15 129 190 319
42 127 210 337
104 90 135 225
105 89 103 192
168 102 117 219
94 137 216 353
68 131 184 315
5 142 206 348
60 145 23 382
125 90 126
125 90 126 216
29 146 223 369
111 109 128 237
EOD

NOTE  Column 1 is the plot number
NOTE  Col. 2 is no. of small trees
NOTE  Col. 3 is no. of large trees
NOTE  Col. 4 is total no. of trees

DISplay X1-X4 ;WID 8


NOTE  Some corrections to be made

INSert row 3 into X1-X4
88 122 206 328
EOD
NOTE  If correct, (X2+X3) equals X4

?X2+X3-X4
NOTE  11'th element of X3 is wrong
X3(11)=237
```

Figure 16.1 continued

```
DEScribe X4;PERcents 20 50 80
NOTE  PER shows the use of an
NOTE  optional subcommand

PLOt X2 X3
CLEar

NOTE  SCA works in MODe 7 & can be
NOTE  used when PLOt not available

SCAtterplot X2 by X3

NOTE  Seems X2 & X3 are linearly
NOTE  related, so find correlation

CORrelate X2 X3

RECode X1 'Region
1 96   1
97 168 2

NOTE  The data were from 2 regions.
NOTE  X5 (the first empty column)
NOTE  is named 'Region' & contains
NOTE  '1' or '2'

FACtor 'R  2
NOTE  Defines X5 to be a factor
NOTE  column with 2 levels

REPlot; BY X5
CLEar

NOTE  Use SCAtterplot again, if
NOTE  REPlot not available

SCAtterplot X5 X2 ;HEIght 5

NOTE  Shows the high correlation
NOTE  may be largely due to the
NOTE  differences between the regions

NOTE  All columns have automatically
NOTE  been saved on the disc so the
NOTE  analysis can easily be resumed
NOTE  after further thought
PAUse 0
PAGe ON
WARn ON
MENu
```

This file was entered and edited with one of the BBC's standard word processing packages. The 'full screen' editing facilities of any word processing package are also particularly useful when a reasonably large data set has to be entered. Data that have been output from another program in a form that can be edited by a word processing package can be input to INSTAT in a similar way.

It is often useful to turn WARnings and PAGe mode off when using an EXEC file. The commands are

```
: WARn OFF : PAGe OFF
```

If results will be displayed, the command

```
: PAUse 2
```

gives a 2 second delay after each command has been executed. If a longer delay is needed, press the <SHIFT> and <CTRL> keys together, to freeze the display. Hold these two keys down until you are ready to continue the run.

An EXEC file can be used to extend the commands that are available in INSTAT. For example, INSTAT has no commands for distribution free tests, though it does have the basic building blocks with SORt and RANk. Figure 16.2 shows how these can be used to provide the Mann-Whitney test for comparing two groups. It is assumed that the data are in X1, the treatment codes are in X2 and that the rest of the worksheet is empty. Once the file has been entered, Figure 16.3 shows that the test is given merely by typing : **EXEC MANNU*. The test data set is from Gregory [1978].

Figure 16.2 Listing of file MANNU

```
NOTE DATA Should be in X1
NOTE Treatment codes in X2
WARnings OFF
ECHo OFF
DISplay X1 X2
SORt X1 X2 to X3 X4
FACtor X4 2 levels
RANk X3 to X5
STAtistics X5;BY X4;SUM X6;COUnt X7
K1=X7(1)
K2=X7(2)
K3=X6(1)
K4=K1*K2+K1*(K1+1)/2-K3
TITle Mann Whitney U Statistic
TITle ------------------------
TITle K1 and K2 are the sample sizes
DISplay K1 K2
TITle K4 is Mann Whitney
DISplay K4
```

Figure 16.3 Results from running EXEC file MANNU

```
: *EXEC MANNU
: NOTE DATA Should be in X1
: NOTE Treatment codes in X2
: WARnings OFF
: ECHo OFF
: DISplay X1 X2

  Row            X1            X2

    1            25             1
    2            35             1
    3            38             1
    4            46             1
    5            15             2
    6            24             2
    7            29             2
    8            33             2
    9            36             2
   10            40             2

: SORt X1 X2 to X3 X4
: FACtor X4 2 levels
: RANk X3 to X5
: STAtistics X5;BY X4;SUM X6;COUnt X7

    Statistics for X5

  X4              Sum      Count
  Levels           X6         X7
    1              27          4
    2              28          6

: K1=X7(1)
: K2=X7(2)
: K3=X6(1)
: K4=K1*K2+K1*(K1+1)/2-K3

  Mann Whitney U Statistic
  ------------------------
  K1 and K2 are the sample sizes
: DISplay K1 K2
K1 =             4
K2 =             6

   K4 is Mann Whitney
: DISplay K4
K4 =             7
```

16.4 SYSTEM INTEGERS

Nine system integers are available, %1, %2, ..., %9. They are declared with the calculate command and may then replace any integer within a command. For example

```
: %1 = 4
: DISplay X%1
```

displays the data in X4.

This facility allows the EXEC file given in Figure 16.3 to be generalised (at the expense of readability!), so that the data and treatment codes can be in any columns. The worksheet must however have five free columns that can be used to store intermediate results. Figure 16.4 gives the rewritten file. To use it on a worksheet, which has the data in X3 and treatment codes in X4, type

```
: %1 = 3 : %2 = 4
: *EXEC MANNU2
```

Figure 16.4 Listing of file MANNU2

```
NOTE DATA Should be in X%1
NOTE Treatment codes in X%2
WARnings OFF
ECHo OFF
SORt X%1 X%2 to 'Data 'Group
FACtor 'Group 2 levels
RANk 'Data to 'ranks
STAtistics 'ranks;BY 'Group;SUM 'sum;COUnt 'count
%3='count(1):%4='count(2)
K1=%3*%4+%3*(%3+1)/2-'sum(1)
TITle Mann Whitney U Statistic
TITle ------------------------
TITle %3 and %4 are the sample sizes
TITle K1 is Mann Whitney
DISplay K1
```

Figure 16.5 Results from running EXEC file MANNU2

```
: *EXEC MANNU2
: NOTE DATA Should be in X%1
: NOTE Treatment codes in X%2
: WARnings OFF
: ECHo OFF
: SORt X%1 X%2 to 'Data 'Group
: FACtor 'Group 2 levels
```

Fig. 16.5 cont'd

```
: RANk 'Data to 'ranks
: STAtistics 'ranks;BY 'Group;SUM 'sum;COUnt 'count

   Statistics for ranks

 Group          Sum      Count
 Levels         sum      count
   1             27          4
   2             28          6


: %3='count(1):%4='count(2)
: K1=%3*%4+%3*(%3+1)/2-'sum(1)

  Mann Whitney U Statistic
  ------------------------
  %3 and %4 are the sample sizes

    K1 is Mann Whitney

 : DISplay K1
   K1 =              7
```

16.5 STORING COMMANDS WITHIN AN INSTAT WORKSHEET

The USE command provides an alternative to the *EXEC described above for storing commands that can then be executed automatically. The commands are first entered into strings (S1, S2, ...) with the ENTer command. A number of sub-commands are available with the USE command, giving more flexibility than with an EXEC file.

Figure 16.6 shows a simple example to RANk each of the 12 columns in the file @.RAIN10. Here the 'macro' is a single line (in S1) consisting of

```
: RANK X%1 into X%1 : %1 = %1 + 1
```

and after setting %1 = 2, the command

```
: USE S1 ;REPeat 12
```

ranks each of the 12 columns.

Figure 16.6 Example of Ranking, with the USE command.

```
     : CREate @RAINRAN; COL 20 10; STRing 5; CONstant 10
     Enter title for worksheet (or RETURN).
     Example of USE command
     : INPUT @RAIN10 X1-X13; INTO X1-X13

     :  DIS X2-X13; WID 5; FIX 0

Row   X2   X3   X4   X5   X6   X7   X8   X9  X10  X11  X12  X13

  1  89.  81. 112. 155. 590. 174. 292. 253. 336. 458. 121. 102.
  2  78.   1. 110.  74. 234. 247. 293.  60. 114. 265. 200. 133.
  3  59. 107.  62. 192. 420. 114.  49. 213.  90. 496. 218. 553.
  4 101. 127. 169. 160. 301. 110. 174. 139. 130. 232. 324.  92.
  5 115.  26. 106. 184. 161. 243. 156. 224. 484. 298. 197. 186.
  6  80.  12.  54.  87. 303. 214.  53.  99. 389. 520. 328. 104.
  7  36.  66. 214. 110. 264. 362. 239. 128. 157. 422. 197. 114.
  8   1.  73.  54. 416. 393. 194. 365. 141. 394.  45. 208. 121.
  9  91.  85. 157. 357. 181. 286. 145. 162. 155. 352. 329. 101.
 10  96.  18.  87. 403. 164. 100. 146. 109.  51. 448. 750. 313.

     : ENTer S1
     S1: RANK X%1 X%1 : %1=%1+1
     : ECHO OFF
     : WARN OFF
     : %1=2
     : USE S1; REPeat 12
     : DISplay X2-X13; WID 5

Row   X2   X3   X4   X5   X6   X7   X8   X9  X10  X11  X12  X13

  1   6    7    7    4   10    4    8   10    7    8    1    3
  2   4    1    6    1    4    8    9    1    3    3    4    7
  3   3    9    3    7    9    3    1    8    2    9    6   10
  4   9   10    9    5    6    2    6    5    4    2    7    1
  5  10    4    5    6    1    7    5    9   10    4    3    8
  6   5    2    2    2    7    6    2    2    8   10    8    4
  7   2    5   10    3    5   10    7    4    6    6    2    5
  8   1    6    1   10    8    5   10    6    9    1    5    6
  9   7    8    8    8    3    9    3    7    5    5    9    2
 10   8    3    4    9    2    1    4    3    1    7   10    9
```

A more substantial example, Figure 16.7, extends the ANOva command for data sets with missing values. Once set up, the commands

```
: USE S1 : USE S3 ;WHIle SSQ(X4)>0.001 : USE S5 S6
```

should repeat the ANOva on the randomized block data in X1, until the estimates of the missing values hardly differ on successive iterations. If necessary, the 'stopping rule' given above can be adjusted.

Once written it is always tempting to improve on a macro, or make it easier to use. For example, if S7 is entered as

```
: ECH OFF : WAR OFF : PAG OFF : VDU 21 : USE S1
: USE S3 ;WHI SSQ(X4) > 0.001 : VDU 6 : USE S3 S5 S6
```

then just typing

```
: USE S7
```

should execute the commands, turning off the display while the iterative stage is in progress.

The USE command is a very powerful facility. However, these commands have to be executed by INSTAT which is written largely in BASIC. Thus, execution of a complicated macro can be slow. We have found the macro facility to be most valuable for small sets of commands and also to help the emphasis that an introduction to computers does not inevitably have to involve learning a standard programming language.

Figure 16.7

```
    : NOTE Missing values in ANOVA
    : NOTE Example of Randomised Block
    : NOTE YVAriable in X1,
    : NOTE Block and treament factors in X2 and X3
    : NOTE Change ANOVA command in S3 for other designs
    : NOTE Row numbers of missing values in X6
    : NOTE Missing value code assumed = -9999

    : PAGe OFF
    : WARnings OFF
    : ECHo OFF
    : CREate @TESTMIS;COLumns 10 25; CONstants 10; STRings 10
   Enter title for worksheet (or RETURN).
   Missing Values in Analysis of variance

    : ENter S1-S6
S1: SEL X1X7 ;IF X1>-9998:%3=COU(X6):X8=MEA(X7);UNI %3:X10=0;
    UNI %3:X4=X8
S2: %1=X6(%2):X1(%1)=X8(%2):%2=%2+1:YVA X1
S3: %2=1:USE S2;REP%3:ANOX2X3;RESX9:%2=1:X10=X8:USE S4;REP %3
    :X4=X8-X10


S4: %1=X6(%2):X8(%2)=X8(%2)-X9(%1):%2=%2+1
S5: NOTE Error mean square is given by:?ESS/(EDF%-%3)
S6: NOTE Use this value to get revised SE's

    : NOTE try on data from EXPERI

    : INPUT @EXPERI X1-X3;INTO X1-X3
    : FAC X2 4 X3 3

    : NOTE Set just 1 obs. as missing
    : NOTE to compare with Mead & Curnow pp59-61
    : X1(7)=-9999
    : ENTer X6;DATa 7
    : USE S1: USE S3; WHIle SSQ(X4)>.001

   Number of cases = 11

   (first use of S3)

          ANOVA TABLE
 Source   DF      SS         MS          F
 ------------------------------------------
 X2        3    2266.5     755.49       1.7
 X3        2    3869.9       1935       4.4
 Error     6    2613.2     435.54
 ------------------------------------------
 Total    11   8749.64
 ------------------------------------------
```

Fig. 16.7 cont'd

```
      MAIN EFFECTS
      ------------

      X2                    X3
   Level    Mean       Level    Mean

       1  353.667          1  306.500
       2  321.667          2  346.205
       3  333.273          3  342.750
       4  318.667

 SE. diff.   17.040   SE.diff.   14.757

. . . . . . . . . . . . . . . . . . . . .

  (after 10 iterations)

           ANOVA TABLE

 Source   DF      SS       MS        F
 ---------------------------------------
 X2        3    2608.4   869.47    2.5
 X3        2    4946.4   2473.2    7.0
 Error     6    2111.4    351.9
 ---------------------------------------
 Total    11    9666.14
 ---------------------------------------

       MAIN EFFECTS
       ------------

       X2                    X3
    Level    Mean       Level    Mean

       1  353.667          1  306.500
       2  321.667          2  354.110
       3  343.813          3  342.750
       4  318.667

  SE.diff.   15.317    SE.diff.   13.265

 : ECH
 : USE S5 S6
 : NOTE Error mean square is given by

  Command - ?ESS/(EDF%-%3)
     422.28

 : NOTE Use this value to get revised SE's
```

REFERENCES

COCHRAN W.G. and COX G.M. [1957] - *Experimental Designs*, John Wiley and Sons, Inc., New York.

DRAPER N.R. and SMITH H. [1981] - *Applied Regression Analysis - 2nd Edition*, John Wiley and Sons, Inc., New York.

GREGORY S. [1978] - *Statistical Methods and the Geographer - 4th Edition*, Longman, London & New York.

MEAD R. and CURNOW R.N. [1983] - *Statistical Methods in Agriculture and Experimental Biology*, Chapman and Hall, London

POLLARD J.H. [1977] - *A Handbook of Numerical and Statistical Techniques*, Cambridge University Press, Cambridge

ROTHAMSTED EXPERIMENTAL STATION [1983] - *GENSTAT - A General Statistical Program*, Numerical Algorthms Group Ltd., Oxford

SIEGEL S. [1956] - *Nonparametric Statistics for the Behavioral Sciences*, McGraw-Hill

SNEDECOR G.W. and COCHRAN W.G. [1980] - *Statistical Methods - 7th Edition*, Iowa State University Press, U.S.A.

TUKEY J.W. [1977] - *Exploratory Data Analysis*, Addison-Wesley Publishing Co., Inc., U.S.A.

VELLEMAN P.F. and HOAGLIN D.C. [1981] - *Applications, Basics and Computing of Exploratory Data Analysis*, Duxbury Press, Wadsworth Inc.

APPENDICES

APPENDIX 1: ADAPTING INSTAT

If INSTAT is to be adapted for use on a single-sided disc drive, you will need at least temporary access to an 80 track double sided disc drive. If this is not possible, see READ ME.

1. Connect the double sided disc drive to the BBC microcomputer containing the INSTAT EPROM.

2. Make a copy of the INSTAT disc onto an 80 track double sided disc.

3. Place this disc in the 80 track drive, as side 0/2.

4. Switch off the 6502 second processor (if attached to the BBC)

5. Press <CTRL> and <BREAK> keys simultaneously.

6. Type **INSTAT* or **IN*.

7. From the 'Introductory Menu', (see Figure 2.1), choose option 8 and press <RETURN>.

8. From the 'Starting Options' menu, (see Figure 2.2), type 1. The flashing cursor will move to the RHS of the screen, opposite

1) Main drive for program files.

Type 0 and *<RETURN>*

The first line of the screen will now read

1) Main drive for program files 0

Type *<RETURN>*

9. The program will return you to the BASIC prompt >

10. This double sided disc is now ready to be used as a master disc, from which to create new discs to be used in a single sided disc drive, by using the following instructions.

Installing INSTAT with a disc drive of less than 400K.

INSTAT can be split up to fit on 3 100K 40 track discs.

Suggested files on each disc, assuming no 6502 Second Processor, are as follows:-

Disc 1	Disc 2	Disc 3
M.INSTAT	INS4 - INS8	INS19
INSDATA	INS11 - INS18	TXT1A
INS0 - INS3		TXT1B
INS9, INS10		HPTR1
ERR		data files

If a Second Processor is available, the file INSTAT needs to be added to disc 1, which is normally in the drive. With M.INSTAT, but not INSTAT, there is about 60k of this disc for data files. You are prompted to insert either of the other discs, when they are required.

It is suggested that you have several copies of disc 1, each containing different data files. Only one copy each of disc 2 and 3 should be needed.

Figure A.1 gives the commands which correspond to each of the program files in INSTAT and INS0 to INS19. If any of these commands are needed frequently, it is advisable to have the corresponding file on disc 1.

A user with a single sided 80 track drive (or a double density 40 track) has more flexibility. For example, with screen memory for plotting but without a second processor, the files

```
M.INSTAT
INSDATA
INS0 - INS5
INS7 - INS10
INS13, INS14
ERR
```

on the primary drive would give the user all the standard facilities plus plotting and regression.

Figure A.1 Commands on each Program File.

This table is essential if you wish to experiment with different program files on the disc.

File Name	Commands
INSTAT/M.INSTAT	AGAin, CLEar, ECHo, ERRor, HEAding, MODe, NOTe, PAGe, PAUse, QUIt, TITle, VDU, WARn
INS0	MENu
INS1	DISplay, ENTer, INSert, READ, RECode
INS2	CALculate, SELect, SHOw, USE
INS3	INFormation, INPut
INS4	DELete, INDicator, NORmal, OUTput, RANk, SORt
INS5	CORrelate, FACtor, INTeraction, LOCk, NAMe, REMove, TERms, UNLock, YVAriable
INS6	ANOva, ONEway
INS7	SCAtterplot
INS8	DEFine, HIStogram, LINe, PLOt, REPlot, SYMbol
INS9	HELp
INS10	CLOse, CREate, MACro, OPEn
INS11	PREsent
INS12	DEScribe, STAtistics
INS13	TABle
INS14	ADD, DROp, ESTimate, FIT, REFit
INS15	BOXplots, STEm-and-leaf
INS16	GENerate
INS17	GAMma, TINterval
INS18	FREquencies, PERcentages, PRObabilities
INS19	CONfigure, KEY

APPENDIX 2: Notes about the EPROM

The main command within the EPROM is *INSTAT

INSTAT can only be called from BASIC.

Calling other language ROMS from within INSTAT is not recommended - press the BREAK key to return to BASIC first.

You may however call ROMS with SERVICE calls, e.g. to do a screen dump.

Two screen dumps are included on the EPROM:-

1) *DTEXT which dumps the text on the screen to any printer in Modes 0, 1, 3 and 7.

2a) *DNEC a fast Mode 0 dump for the NEC 8023 printer. The dump uses the full width of the paper and uses approximately the same aspect ratio as the screen. A dump takes about 1 min. 50 secs. without printer buffer.

2b) *DEPSON a fast Mode 0 dump for EPSON and compatible printers. This uses approximately two-thirds of the width of the paper and reproduces the screen with a different aspect ratio.

These commands can be preceded by R (for Reading) to prevent clashes with other ROMS.

The screen dumps are not available outside INSTAT.

Note: If you wish to use your own screen dump you can use the buffer areas of memory from &900 to &AFF to *RUN a machine code screen dump stored on disc. Note that INSTAT uses these areas when processing commands, so a screen dump routine will subsequently be overwritten. Zero page locations &72 to &8F are free for the user. Locations &70, &71 and &F8 must not be used, as INSTAT uses them at all times.

APPENDIX 3: Fitting the INSTAT EPROM

1. EPROMs are very delicate and should be handled with care. Do not bend the pins and never put them in contact with plastic, artificial fibres or other sources of static electricity.

2. Make sure your BBC is NOT plugged into the mains.

3. Unscrew the two large screws at the top of the back panel. Also unscrew the two similar large screws located near the front feet on the underside of the microcomputer. These are sometimes labelled FIX.

4. Lift the lid off the computer. If the lid does not lift off easily, check that you have removed the correct four screws.

5. Unscrew the two nuts and washers which hold the keyboard down. These are located at either end of the plastic board under the keyboard. The screws attached to these nuts are positioned about two inches to the rear of the screws you removed in section 3. Note their position for reassembly.

6. The keyboard should remain attached to the BBC by a grey ribbon connection at all times. Lift the front of the keyboard so that the keyboard is swivelled about its rear edge and layed flat on the rear half of the BBC.

7. At the very bottom right of the circuit board there are five 28-pin sockets, which contain the operating system, EPROMS and ROMS. Insert the INSTAT EPROM in a free ROM position, making sure that the notch points towards the back of the BBC. Do not remove the INSTAT label from the top of the EPROM.

8. Place the keyboard back and screw it into place, with its two nuts and washers.

9. Replace the lid and refit all the screws.

Note The position of the 28-pin sockets are different in the circuit board for the BBC+ and the BBC Master.

APPENDIX 4: An Introductory Practical for the BBC micro.

1. Switch on the computer. It should give a "beep" and a small light should also come on. If nothing is displayed on the monitor, then switch this on as well.

2. This exercise is to give practice in using the machine as a typewriter.

a) Type PRINT 2 + 2

Then press the <RETURN> key. This signals it is the turn of the computer to do something.

It should say 4

b) Use the arrow keys and the <COPY> key to edit this line into

(i) PRINT 2 + 3 - then press <RETURN>
(ii) PRINT LOG(2 + 3) - then press <RETURN>
(iii) PRINT LOG(2 + 3) + 4.5 - then press <RETURN>

c) Note the use of the <DELETE> key to delete any mistakes on the line you are typing

d) Type PAINT 2 + 3 - then press <RETURN>

The machine should say 'Mistake' - i.e. it cannot understand the word PAINT. Edit the PAINT to PRINT and it is happy again.

3. Some commands to the machine are preceeded by a *

For example try

*KEY1 PRINT <RETURN>

It looks as though nothing has happened. Now press the red key marked f1 and PRINT should appear on the screen. These red keys are called function keys and can be used as typing aids.

4. a) Type the following. It is a simple program to find the mean of 3 numbers but is mainly of interest here to demonstrate more of the facilities of the machine. (Remember from now on to press the <RETURN> key at the end of each line.)

```
10 INPUT A, B, C
20 PRINT "THE MEAN IS " ; (A + B + C)/3
30 END
```

b) Now clear the screen by typing.

CLS then <RETURN>

c) Then type LIST to list the program on the screen.

d) To run the program type RUN
It should give you a ?
Type any number and then press the return key. Repeat this 3 times e.g.

```
? 14
? 16
? 30
```

It should now say THE MEAN IS 20

Try this a few times. Notice a number can be corrected with the <DELETE> key if the mistake is detected before <RETURN> is pressed, e.g.

? 14.426 then press DELETE 3 times and change it to 14.246

e) List the program and edit it to give the mean of four numbers. This is intended to give more practice in the use of the 'editing keys'.

```
10 INPUT A, B, C, D
20 PRINT "THE MEAN IS " ; (A + B + C + D)/4
30 END
```

- check this still works by typing RUN.

5.
a) The computer can operate in a number of different "modes".

Type MODE 2 then press return.

The screen should go blank. Type LIST. The program should now be displayed in large characters with just twenty across the screen.

Run the program in this mode. It should work equally well.

b) There are 8 different modes giving 20, 40 or 80 characters across. Try MODE 0 up to MODE 7 in turn and fill in the table below.

MODE	Number of characters across the screen
0	
1	
2	20
3	
4	
5	
6	
7	40

c) Mode 7 has some differences from all the other modes. In particular 5 of the keys on the right hand side of the keyboard appear differently on the screen in Mode 7. Type MODE 7 and try the different keys to fill in the table below.

Keys in Mode 0-6	Keys in Mode 7
]	
[	
}	
{	
¦	
\	
^	
~	

The other key that is slightly different is the number zero. This looks like O in Mode 7 and has a line through it (as on the keyboard) in Modes 0 to 6.

You must also be careful to distingush between the numbers 0 and 1 and the letters O, I and l. Mixing them up is not a problem to the computer - it will say 'Mistake' or perhaps give the wrong results - and this might be a serious problem to you.

6. If you switch the computer off (don't do so yet!) you will lose the program you have just typed. If it were valuable you might like to save a copy of it on a disc. Put one in the diskette drive.

a) Type SAVE "TEST" - you can give any name as long as it has 7 letters or less.

On pressing <RETURN> you should hear the disk drive operating.

b) Type LIST then <RETURN>. Notice the program is still in the machine. A copy has also been made on the disc. Now press the <BREAK> key and type LIST again. The program has now gone from the central memory of your machine. Luckily there is (I hope) a copy

on your disc. Type LOAD "TEST" to transfer a copy back to the central memory. Type LIST. It should now be there again.

7. Now a couple of * commands.

*CAT (short for catalogue)

should list the names of all the files on your disc. Now type

*DELETE TEST which deletes the file TEST from the disc.

Try *CAT again and you will see it has gone. Sorry about that! If you still want it, a copy is in the central memory and you will have to save it again.

8. Check you are starting to feel at ease with the computer. You should be able at least:

a) to use it as a simple calculator by just typing

PRINT 2 + 2 etc

b) to find where the different symbols are on the keyboard, reasonably quickly.

9. This question introduces the computer as a scientific calculator and gives more practice in the use of the function keys.

a) Evaluate the square root of (1/n1 + 1/n2) when n1 = 7 and n2 = 9, i.e. the square root of (1/7 + 1/9).

Alternative ways are as follows:

i) The most direct is to type

```
PRINT SQR(1/7 + 1/9)                    (Ans. 0.5040)
```

ii)
```
N1 = 7 : N2 = 9
PRINT SQR(1/N1 + 1/N2)
```

iii)
```
*KEY 1 PRINT SQR(1/N1 + 1/N2)¦M
```

Then press the red f1 key.

(Note the ¦ symbol is on the top right hand side of the keyboard above the \ symbol. When typed followed by M it gives an automatic carriage return.)

be done once. If there are a few more values of N1 and N2.

(e.g. type N1 = 7 :N2 = 23)

then, once they are reset, just press f1 again.

b) The symbol ^ is used to get powers. Hence evaluate

i) 12^2 + 14^2 + 17^2 + 19^2 (Ans. 990)

ii) (43) ^ (1/3) (Ans. 3.5034)

c) A number of useful functions are also available including

SQR, LN, LOG, EXP, SIN, COS, TAN.

For example, to give the height of the standardised Normal curve for different values of X try

*KEY2 PRINT 1/SQR(2*PI) * EXP(-0.5 * X * X)¦M

Then type X = 0 and, after the carriage return, press the function key f2.

You should get the answer 0.39894228.

Try a few other values, e.g. X=0.5 gives 0.3521
X=1 gives 0.2420
X=1.5 gives 0.1295
X=2 gives 0.0540

d) Notice therefore that the machine can be used as a powerful calculator. It has lots of "memories" to store intermediate results. It also has the function keys which can act as "memories" to store formulae. Finally the fact there is a proper keyboard and screen to display the calculations and the results means that there will be fewer mistakes than with an ordinary calculator.

Some of the important keys

<COPY> <DELETE> and the four arrow keys are useful for editing

<RETURN>

<ESCAPE> when the program is running, will allow you to ESCAPE from the current task, but leaves the program still in memory.

<BREAK> Resets the computer and loses whatever is in memory.

<CTRL> and <SHIFT> pressed simultaneously will 'freeze' the screen for as long as these two keys are held down.

Some useful BASIC commands

LOAD "filename"	to load a named file from the disc
RUN	to run a program that is in the computers central memory.
CHAIN "filename"	same as LOAD and RUN together.
SAVE "filename"	to save and name a file on the disc.
*KEY n	to store text in a function key fn
*DISC	to use the diskette drive with the machine
*DRIVE n	to use a particular disc drive (if you have more than one)
*EXEC filename	to execute a file containing commands as though they were typed on the keyboard
*SPOOL filename	to save a copy of the text that appears on the screen into a file
*TYPE filename	to look at an output file on the screen
*CAT (or *.)	to get a catalogue of the files on the disc
*DELETE filename	to delete a named file from the disc
*ACCESS filename L	to lock a file, preventing it from being overwritten or deleted
*ACCESS filename	to unlock a file