Preface

This guide was originally written for graduate students in Statistics at the University of Oxford. The first versions were based closely on notes by Dr. Bill Venables of the Department of Statistics at the University of Adelaide, but have been updated to reflect later versions of S, the extension of S-Plus and local facilities. Several sections, in particular 4, 6 and 11, remain closest to Dr. Venables' original material. This guide will no longer be updated following the publication of Venables & Ripley (1994) [see p. 1]. Where new features are introduced in S-Plus, their first use will be described following the introduction of S-Plus and local facilities. Extensions of S-Plus and local facilities are described in Appendix 4. Current information on the extension of S, the University of Oxford's system, is available at http://www.stats.ox.ac.uk. The extended version was made freely available to users of the Department of Statistics at the University of Oxford.

The guide is intended for users of S-Plus, but much of it will be relevant to users of S and to S-Plus. The guide is intended for readers familiar with S-Plus, but readers familiar with S will find the material on the latter interesting. The terminology of this guide is intended to be precise, only referring to S-Plus rather than S for features unique to S-Plus.
Introduction

1.1 Starting and Finishing

This is waiting for input from you. After a short while (and, the first time, an initialization message) you get the S-Plus prompt. The NEWS language.

1.2 Getting Help

This window can be closed, but the window is not discarded. If an expression is given as a command, it is evaluated, printed, and the value is discarded. An assignment also evaluates an expression and passes the value to the command.

Expressions and assignments are evaluated by a lexical scanner. Elementary commands consist of either simple strings or a single expression. The help window can be extended by writing new functions, which then can be used exactly as built-in functions (see also page 3).

S-Plus is a function language whose organization differs from release to release.

The NEWS system was radically re-designed in the 1988 release and known as S, with added functions, produced by the StatSci Division of MathSoft in Seattle.

S is a statisticallanguage developed at AT&T's Bell Laboratories. It is not the intention of this guide to replace the books. Rather these notes are intended as a brief introduction to the capabilities of the S language.

1.3 Using and Finding

S-Plus is a statisticallanguage with a very simple syntax. Like most function languages, S-Plus has an inbuilt help facility similar to the man facility of Unix. To get more information on an any specifc named function or dataset, for example

> ?S-Plus

other books include...

1.4 Publishing and Printing

If you prefer, a separate help window (which can be left up, or removed by quiting) can be obtained by the argument

> help()
It is not advisable to quit S-Plus windows from the frame menu. It is not advisable to quit S-Plus windows from the frame menu. Graphics are printed by holding down the right button on the S+ menu (see 6) and releasing over the print item. This will print to the nearest laser printer (orthatselectedbyyour TU environment variable). To record a session cut-and-paste to a "#" window, then remove your mistakes (if any) and save as a Unix file.

Datasets

Datasets are stored in a directory %Z. They are permanent, so all the objects you create are retained until explicitly deleted. (As the directory name %Z begins with % it will normally be hidden in file listings from Unix by %.) If there is a %Z directory in the current directory when S is invoked, that directory is used rather than %Z. This provides one way to organize your S, using separatedirectoriesforeach project. In S, to get a list of names of the objectscurrently defined use the command

You own functions are also stored in %Z. To find out whether an object is a function or dataset, and what is in it, just type its name at the prompt, e.g.

This prints out the function, dataset, etc. In the later versions of S it may print a short summary of the object. To get the full details, use

When S looks for an object, it searches inturn through a sequence of directories known asthe search list. Usually the first entry in the search list is the %Z subdirectory of the current working directory. The names of the directories currently on the search list can be found by the function

Thenames of the objectsheld in any directory on the search list can be displayed by giving the function an argument. For example

liststhe contents of the second directory in the search list. Normally these secondsixth and seventh directories contain standard datasets. Extrasearch directories can be added to this list with the function and removed with the function, detailsof which can be found in the manuals of the package.

Warning

The function remove(· · ·) can be used to remove objects with non-standard names.

(For a list of standard names see %.) It is recommended that the function remove is available and checked in any awkward situation. More than one directory can be searched for the "data" directory in the order listed with the function (· · ·).
4.2 Vector Arithmetic

The operations of the algebra of vectors are: addition, subtraction, and scalar multiplication. The operations are all defined in terms of a vector, and a vector is defined as a set of numbers.

Addition of two vectors 

\[ \mathbf{a} + \mathbf{b} = (a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n) \]

Subtraction of two vectors 

\[ \mathbf{a} - \mathbf{b} = (a_1 - b_1, a_2 - b_2, \ldots, a_n - b_n) \]

Scalar multiplication of a vector 

\[ c \mathbf{a} = (ca_1, ca_2, \ldots, ca_n) \]

The dot product of two vectors 

\[ \mathbf{a} \cdot \mathbf{b} = a_1b_1 + a_2b_2 + \cdots + a_nb_n \]

The cross product of two vectors 

\[ \mathbf{a} \times \mathbf{b} = (a_2b_3 - a_3b_2, a_3b_1 - a_1b_3, a_1b_2 - a_2b_1) \]

The magnitude of a vector 

\[ \|\mathbf{a}\| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2} \]

The angle between two vectors 

\[ \cos \theta = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \]

4.3 Simple Data Manipulation

The basic data objects in R are vectors, arrays, lists, and data frames.
4.3 Generating Regular Sequences of Numbers.

Logical vectors are generated by coercion. For example,

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]

Logical vectors are coerced to numeric vectors by coercion, hence

\[
(1, 2, 3) \rightarrow 1, 2, 3
\]

Logical vectors are coerced to character vectors by coercion, hence

\[
(1, 2, 3) \rightarrow \text{true, false, true}
\]
4.7 Arrays

```r
> x = c(1, 2, 3)
> y = c(4, 5, 6)
> z = x + y
> z
```

In this case, `z` is a vector of length 6, containing the element-wise sum of `x` and `y`.

4.8 A vector of character strings

```r
> vec = c("apple", "banana", "cherry")
> vec[3]
```

In this case, `vec[3]` returns "cherry".

4.9 Indexing and Modifying SubsetsofaDataset

```r
> data <- read.csv("data.csv")
> data[1,]
```

This returns the first row of the dataset.

4.10 More generally, subsetsofavector(oranyexpressionthatevaluatestoavector)maybesubse-
lectedbyspecifyingtheelementinsquarebrackets,eg.

```r
> data[2,]
```

In this case, `data[2,]` returns the second row of the dataset.

4.11 Vectors of character strings

```r
> names <- c("John", "Jane", "Bob")
> names[2]
```

In this case, `names[2]` returns "Jane".

4.12 Elementsofavectorcanbenamed(aswellasnumbered)byassigningacharactervector

```r
> names <- c("John", "Jane", "Bob")
> names[2] <- "Jo"
> names
```

This changes the name of the second element to "Jo".

4.13 Indexing and Modifying SubsetsofaDataset

```r
> data <- read.csv("data.csv")
> data[1,]
```

This returns the first row of the dataset.

4.14 More generally, subsetsofavector(oranyexpressionthatevaluatestoavector)maybesubse-
lectedbyspecifyingtheelementinsquarebrackets,eg.

```r
> data[2,]
```

In this case, `data[2,]` returns the second row of the dataset.

4.15 Vectors of character strings

```r
> names <- c("John", "Jane", "Bob")
> names[2] <- "Jo"
> names
```

This changes the name of the second element to "Jo".
and the structure can be treated as a two-dimensional array:

\[ \begin{pmatrix}
  a & b \\
  c & d
\end{pmatrix} \]

Components can also be accessed individually, either by giving the component name as a character string in place of the number in double quotes, or as a similar array with data vector the result of an element-by-element operation on the data vector. The dimension vectors of operand lists generally need to be equal in the number of elements, and the result of a cross-product of a numeric vector, a logical vector, a matrix, or a function.

The matrix multiplication operator is \( \times \) and can be used as well as a function. The names of components may be abbreviated down to the minimum number of letters needed.

Data frames were introduced in the August 1991 release of R and can be thought of as closely related to data bases. They are comprised of a number of objects, each an object of a different type, and may always be referred to as such. If the names are omitted, the components are numbered only. There is no particular need for the components to be of the same mode or type, and, for example, lists, data frames, or numeric are allowed. Component names may be abbreviated, and may always be referred to as such. If the names are omitted, the components are numbered only. There is no particular need for the components to be of the same mode or type, and, for example, lists, data frames, or numeric are allowed. Component names may be abbreviated, and may always be referred to as such. If the names are omitted, the components are numbered only. There is no particular need for the components to be of the same mode or type, and, for example, lists, data frames, or numeric are allowed.
Reading data into S

Character vectors read into columns are automatically read as factors (see §10), unless specified within a function.

```
character vectors read into factors are automatically read as factors
```

```
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.02</td>
<td>1.04</td>
<td>1.06</td>
</tr>
<tr>
<td>2.08</td>
<td>2.10</td>
<td>2.12</td>
</tr>
<tr>
<td>3.14</td>
<td>3.16</td>
<td>3.18</td>
</tr>
<tr>
<td>4.20</td>
<td>4.22</td>
<td>4.24</td>
</tr>
<tr>
<td>5.26</td>
<td>5.28</td>
<td>5.30</td>
</tr>
</tbody>
</table>
```

```r
cchars <- c("A", "B", "C")
cnames <- c("one", "two", "three")
```

```r
> cchars
[1] A B C
>
```

```
characters are read by row.
```

If they were named vectors.

```
Dataframes can be read from a file by the read.table command. The data should be a table in one of a number of formats.
```

```r
> The argument
>
```

```r
> function.Thedata®leshouldbeavectorfile.
```

```
Matrices are usually read by row, as follows.
```

```
To read from a file specify its name as the first argument, for example
```

```
A vector is read as a character vector.
```

```
A data frame can be created from vectors and matrices by the `read.table` function. For example:
```

```r
> > dat <- matrix(c(10:4, 5:1), ncol = 2)
> > dat
     [,1] [,2]
[1,]   10    5
[2,]    9    4
[3,]    8    3
[4,]    7    2
[5,]    6    1
>
```

```
> character vectors read into columns are automatically read as factors (see §10), unless specified within a function.
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```
are as follows:

* A command is issued to construct a plot from data. For example:
  
  1. The str of a point plot, where x and y are vectors giving the x- and y-coordinates, respectively.
  2. A command is issued to construct a plot from data. For example:

6 Chapters

2 Column Data

The function \( u \) and \( v \) refer to the function part of the command.

Other parameters control the background characteristics of all subsequent plots and

differentiate a command by adding a value to the function part (\( \cdot \cdot \cdot \)). There are two
higher levels of these.

Components of the command

are usually specified by a call to the function part (\( \cdot \cdot \cdot \)).

The function \( u \) and \( v \) refer to the function part of the command.

Approximate minimum and maximum values for

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)

\( \cdot \cdot \cdot \)
places selected by mouse clicks. This is somewhat accurate if you draw on the plotting area in
the conventional way. The area within which the mouse actions are made is known as
the grid. The grid is used to determine where the mouse clicks are located.

6.2 Some Basic Plotting Functions

6.2.1 Text

The function text() is often used with text to add information to plots. It is

> text(x, y, "text", cex=1)

6.2.2 Points

The function points() is used to add points to already plotted data. It is

> points(x, y)

6.2.3 Lines

The function lines() is used to add lines to already plotted data. It is

> lines(x, y)

6.2.4 Polygons

The function polygon() is used to add polygons to already plotted data. It is

> polygon(x, y)

These are some examples of how dynamic manipulation of graphs allows these

6.4 Brush and Spin

Brush and spin

> text(x, y, "text" )

6.2.5 Equal-Sized Squared Plots

Equal-sized squared plots

> points(x, y)

6.5 Text

Text

> text(x, y, "text")

6.3 Interaction with Plots

Interaction with plots

> points(x, y)

6.6 Plotting

Plotting

> plot(x, y)

6.7 Draw

Draw

> plot(x, y)

6.8 Show

Show

> plot(x, y)
6.5 Equally-scaled plots

Figure 1: Screen dump of an open Source window displaying plots with different highlights for the three groups.

Figure 2: A histogram of feature with two density estimates overlaid.

7 Statistical Summaries

7.1 Arithmetical Summaries

Statistical summaries such as mean, median and variance are available. The var function will take a
Other styles of boxplot are available—see the help pages.

4. Other styles of boxplot are available—see the help pages.

7.3 Boxplots

A box plot is a way to look at the overall shape of a set of data. The central box shows the

Distributions available are:

\( \text{Weibull} \), \( \text{uniform} \), \( \text{STable} \), \( \text{Poisson} \), \( \text{negativebinomial} \), \( \text{log-normal} \), \( \text{exponential} \), \( \text{chisquare} \), \( \text{Cauchy} \), \( \text{t} \), \( \text{correlated} \), \( \text{geometric} \), \( \text{exponential} \), \( \text{chisquare} \).

Figure 3: Boxplots for means of 2000 data.

The mean and median of the data can be computed (roughly) from the plot.

Descriptive statistics for the heights of the cows:

\( \text{Mean} = 105.0, \text{Median} = 105.0, \text{Minimum} = 97.85, \text{Maximum} = 116.45, \text{Range} = 18.6, \text{Standard Deviation} = 4.96 \)
The function makes use of the probability distribution of the samples and computes the appropriate set of probabilities for the plot.

The function replaces one of the samples by a sample from the same distribution. This can be applied to each sample in turn and construct two samples. The function generates a function plot for the samples of two samples and generates the quantile function of the dataset.

The function makes use of the probability distribution of the samples and computes the appropriate set of probabilities for the plot.

The function replaces one of the samples by a sample from the same distribution. This can be applied to each sample in turn and construct two samples. The function generates a function plot for the samples of two samples and generates the quantile function of the dataset.

The function makes use of the probability distribution of the samples and computes the appropriate set of probabilities for the plot.

The function replaces one of the samples by a sample from the same distribution. This can be applied to each sample in turn and construct two samples. The function generates a function plot for the samples of two samples and generates the quantile function of the dataset.

The function makes use of the probability distribution of the samples and computes the appropriate set of probabilities for the plot.

The function replaces one of the samples by a sample from the same distribution. This can be applied to each sample in turn and construct two samples. The function generates a function plot for the samples of two samples and generates the quantile function of the dataset.

The function makes use of the probability distribution of the samples and computes the appropriate set of probabilities for the plot.

The function replaces one of the samples by a sample from the same distribution. This can be applied to each sample in turn and construct two samples. The function generates a function plot for the samples of two samples and generates the quantile function of the dataset.

The function makes use of the probability distribution of the samples and computes the appropriate set of probabilities for the plot.

The function replaces one of the samples by a sample from the same distribution. This can be applied to each sample in turn and construct two samples. The function generates a function plot for the samples of two samples and generates the quantile function of the dataset.
Many of these have alternative methods. For example, the methods of Chapter 7.

Figure 4: Histogram and example CDF of the permutation distribution of the test in the sample.

If there was some small drops in it it is possible: see, for example, R. Haffty (1994, Chapter 7).

The sample size is quite small, and one might wonder about the validity of the distribution.

The density and CDF of the permutation distribution of the test in the sample.

\[ \text{Density of differences is not equal to 0} \]

\[ t = 0.4999 \quad df = 39 \quad p\text{-value} = 0.0485 \]

\[ \text{Pairwise t-test} \]

\[ t = 0.69 \quad p\text{-value} = 0.4966 \]

\[ \text{Pairwise t-test} \]

\[ t = 0.69 \quad p\text{-value} = 0.4966 \]

\[ t = 0.69 \quad p\text{-value} = 0.4966 \]
Handling Categorical Data

10. The Function 

and the variables created by

After the assignment the sampled entries are calculated by

Suppose further we needed to calculate the missing entries of the mean scores. To do this we

and the variable selected are


> for(x in 2:3)

> sapply(x, mean)

> x = x[1]  # vector

> x

To select and to redefine the complementary levels by the levels

function

and redefine the missing entries of the levels.

In conclusion the following command:  

10.1 The Function and Levels

Some of the functions for extracting mods of the factor levels in applicable special ways.
Writing Your Own Functions

Loops and Conditional Execution
This shows how to extract information from a file by the use of auxiliary functions. There are no such functions in R, but they are based on object-oriented principles.

Writing Your Own Functions

A group of commands.

The syntax of a linear model is

The output of a linear model is

13.1 Model Formulae

A model formula couples a y-variable with a model expressed in a terminology very similar to that of GLM and CENSIT. The form is

13 Statistical Models
The analysis of one-way layouts is best illustrated by an example. The above data on the

where given:

\[ \text{test for linearity of response} \]

The parameterization of the chemical groups of 10 patients after oral administration

The parameterization of the chemical groups of 10 patients after oral administration
Consider designed experiments. Consider the chance Box-Plot.

13.3 Designed Experiments

\[
\begin{align*}
\text{I & II} & & \text{I & II} & & \text{I & II} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{align*}
\]

The \( p \)-value is 0.646-15.
The weight gain (in kg) over 12 weeks is given in the table below:

<table>
<thead>
<tr>
<th>Student</th>
<th>Initial Weight (kg)</th>
<th>Final Weight (kg)</th>
<th>Weight Gain (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>60.0</td>
<td>65.0</td>
<td>5.0</td>
</tr>
<tr>
<td>B</td>
<td>70.0</td>
<td>75.0</td>
<td>5.0</td>
</tr>
<tr>
<td>C</td>
<td>80.0</td>
<td>85.0</td>
<td>5.0</td>
</tr>
<tr>
<td>D</td>
<td>90.0</td>
<td>95.0</td>
<td>5.0</td>
</tr>
<tr>
<td>E</td>
<td>100.0</td>
<td>105.0</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Now consider a linear regression. Six different linear models in order of diminishing complexity:

1. **Boxcox(x)**
2. **Boxcox(x)** + **polynomial**
3. **Boxcox(x)** + intercept
4. **Boxcox(x)** + intercept + **polynomial**
5. **Boxcox(x)** + intercept + **polynomial** + **increasing**
6. **Boxcox(x)** + intercept + **polynomial** + **increasing** + **decreasing**

A more detailed view of (4) is in the function boxcox in the library `mgcv`.

```r
library(mgcv)
model <- gam(y ~ x, family = boxcox(model = boxcox(x)), data = mydata)
summary(model)
```

The best command given above for the console (Enter + Tab)?

```r
summary(model)
```

The **Boxcox** function in R allows for a variety of transformations. One common transformation is the Box-Cox transformation, which is a family of power transformations. The transformation is defined as:

\[
Y' = \begin{cases} \frac{Y^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(Y) & \text{if } \lambda = 0 \end{cases}
\]

where \(Y\) is the original variable, \(Y'\) is the transformed variable, and \(\lambda\) is the transformation parameter. The Box-Cox transformation is useful for stabilizing variance and normalizing distributions, which can improve the performance of regression models.

The `mgcv` package in R provides a convenient way to apply the Box-Cox transformation. The `gam` function can be used to fit a generalized additive model with a Box-Cox transformed response variable. The `family` argument in the `gam` function allows specifying the Box-Cox transformation by setting the `family` to `boxcox`. This approach is particularly useful when dealing with non-normal or heteroscedastic data.
13.4 Generalized Linear Models

An example of several points need further explanation:

\[
\begin{aligned}
&\text{logit}(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \epsilon
\end{aligned}
\]

where \( p \) is the probability of success, \( \epsilon \) is the error term, and the \( \beta \)'s are the coefficients.

The link function connects the linear predictor to the mean of the response. Common link functions include the logit, probit, and identity link.

One common approach is to use a linear model with a logit link function:

\[
\text{logit}(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k
\]

where \( p \) is the probability of success, \( x_i \) are the predictors, and the \( \beta \)'s are the coefficients.

The logit link function is defined as:

\[
\text{logit}(p) = \ln \left( \frac{p}{1-p} \right)
\]

A more general approach involves using a non-linear link function:

\[
\text{logit}(p) = \theta(x)
\]

where \( \theta(x) \) is the link function.

There are several advantages to using a non-linear link function:

- It allows for more flexibility in modeling the relationship between the predictors and the response.
- It can capture non-linear relationships between the predictors and the response.
- It can handle data with heteroscedasticity.

The function to fit the non-linear link function is called the inverse link function:

\[
\theta^{-1}(\text{logit}(p)) = p
\]

Once the inverse link function is determined, the model can be fit using maximum likelihood estimation.

An example of this approach is the following:

\[
\text{logit}(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \epsilon
\]

where \( p \) is the probability of success, \( \epsilon \) is the error term, and the \( \beta \)’s are the coefficients.

There are several advantages to using this approach:

- It allows for more flexibility in modeling the relationship between the predictors and the response.
- It can capture non-linear relationships between the predictors and the response.
- It can handle data with heteroscedasticity.

The function to fit the non-linear link function is called the inverse link function:

\[
\theta^{-1}(\text{logit}(p)) = p
\]

Once the inverse link function is determined, the model can be fit using maximum likelihood estimation.
function for multiple regression. 3 The only general procedure, however, is to use stepwise regression and drop or select terms, step by step. This may be repeated until the model is satisfactory. The basic function takes a data frame.

### 13.5 Updating and Selecting Models

<table>
<thead>
<tr>
<th>chdcholesterol</th>
<th>serum</th>
<th>bloodpressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The occurrence of coronary heart disease is important to detect. As this has been discussed, we consider the logistic model of coronary heart disease. The parameterization used is

\[
\text{logit}(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots
\]

where \(p\) is the probability of occurrence. The logistic function gives an estimate of the probability of occurrence.
Figure 7: Discriminant analysis

Principal components analysis (v3.2)
Mahalanobis distances

The classical methods based on variance-covariance matrices

Matrix methods

Multivariate analysis

The measures here are data which compare balance matrices (also used in cascade) and provide particularly rich functions for exploratory multivariate analysis, such as PLS.

14 Multivariate analysis
42 Sources of Libraries

Black Chest refers to libraries and volumes

which are collected together in Carnegie

volumes and made freely available. An archive of libraries is maintained at Carnegie

Library and made publicly available. An archive of libraries is maintained at Carnegie

Library and made publicly available. An archive of libraries is maintained at Carnegie

Library and made publicly available.

A.2 Sources of Libraries

To find more about a section, use

sectional instructions.

For more information on each section or page, please visit:

http://www.library.gov.au

If you have any questions, please contact the libraries.

Libraries are institutions that store, organize, and disseminate information and knowledge.
A.2 Sources of Libraries

Mellon as a service to the statistical profession by Mike Meyer. To obtain details of his courses by e-mail send a message to

fptto@mpl.com, ede. With user ede, ede's

read index

read index

read index

by ede send a message to

Mellon as a service to the statistical profession by Mike Meyer. To obtain details of his courses

A.3 Sources of Libraries