1. Overview of S-PLUS and R

1.0 Introduction

S-PLUS (and its public domain equivalent R) is an integrated suite of software facilities for data analysis and graphical display. It offers:–

- an extensive and coherent set tools for statistics and data analysis
- a language for expressing statistical models and tools for using linear and non-linear statistical models
- graphical facilities for interactive data analysis and display
- an object-orientated programming language that can easily be extended
- an expanding set of publicly available libraries of routines for special analyses

S-PLUS is available as a commercial package from Insightful (formally known as MathSoft) and is an implementation of the language S developed at Bell Laboratories by Becker, Chamberlain and Wilks. R is a very similar implementation but is available free from many different websites. The prime differences between R and S-PLUS (apart from the cost!) are:

- R is an ***Open Source* system** — it is possible to examine the source code and determine precisely what variation on a statistical method has been implemented. This is less important for e.g. t-tests (although even for these there are *equal variance* or *unequal variance* versions of t-tests) but much more important for the more heuristic methods of *robust analysis* and semi-parametric methods, i.e. those modern methods based more on practical consideration than on mathematical theory.

- S-PLUS has *menus and dialogs* as well as a command-line interface, but R has only the command-line.

- S-PLUS has ways to edit graphs and more facilities for multi-panel plots.

- R is better at annotating with mathematical notation.

- R is small with many extensions, S-PLUS is monolithic.

- R runs on less powerful machines.

1.1 Some Features of R

1.1.1 R is a function language

All commands in R are regarded as *functions*, they operate on *arguments*, e.g. `plot(x, y)` plots the vector x against the vector y — that is it produces a scatter plot of x *vs.* y. Even Help is regarded as a function:— to obtain help on the function `plot` use `help(plot)`. To obtain general help use `help()`, i.e.use the function `help` with a null argument. To end a session in R use `quit()`, or `q()`, i.e. the function `quit` or `q` with a null argument. In fact the function `quit` can take optional arguments, type `help(quit)` to find out what the possibilities are.

1.1.2 R is an *object orientated* language

All entities (or 'things') in R are ***objects***. This includes vectors, matrices, data arrays, graphs, functions, and **the results of an analysis.** For example, the set of results from performing a two-sample t-test is regarded as a complete single object. The object can be displayed by typing its name or it can be summarized by the function `summary()`.

1.1.3 R is a *case-sensitive* language

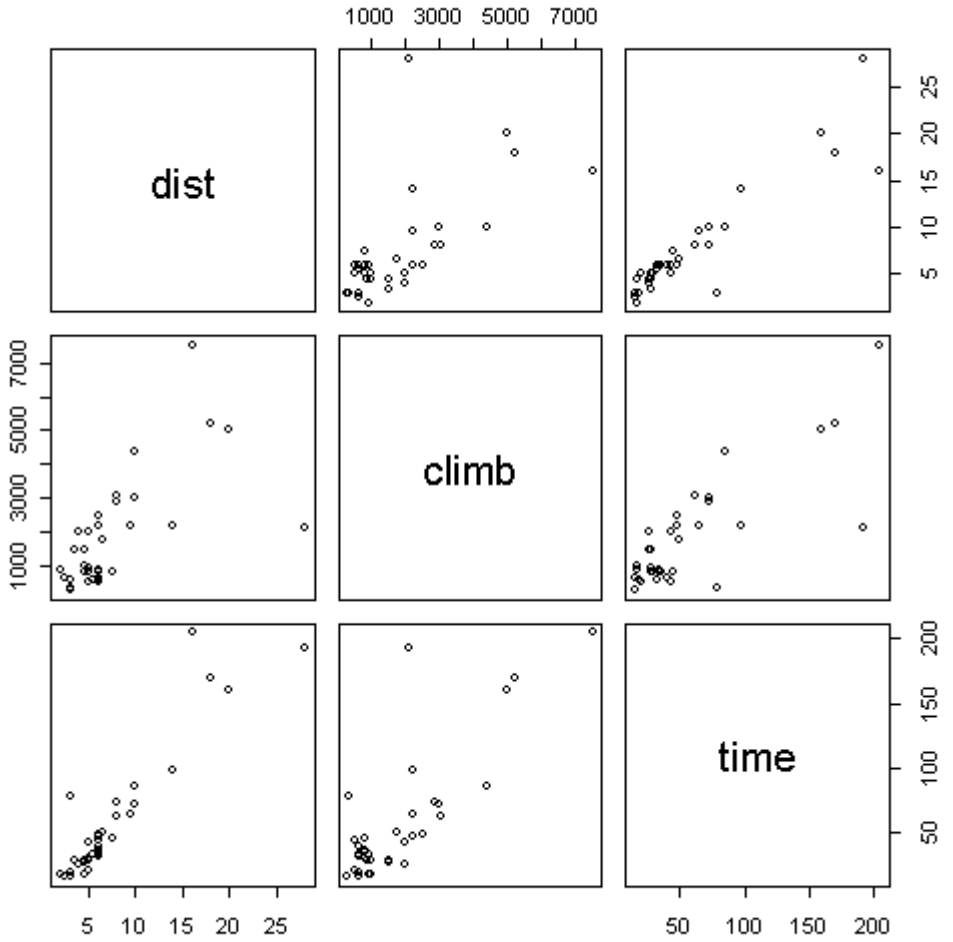
Note that R treats small letters and big letters as different, for example a two sample t-test is performed using the function `t.test()` but R does not recognize `T.test()`, nor `T.TEST()`, nor `t.Test()`, nor……

1.1.4 Brief Example

```
R : Copyright 2001, The R Development Core Team

Version 1.2.2  (2001-02-26)

R is free software and comes with ABSOLUTELY NO WARRANTY.

You are welcome to redistribute it under certain
conditions.

Type `license()' or `licence()' for distribution details.

R is a collaborative project with many contributors.

Type `contributors()' for more information.


Type `demo()' for some demos, `help()' for on-line help,
or

Type `q()' to quit R.


> library(MASS)

> data(hills)

> summary(hills)

      dist            climb          time

 Min.   : 2.000   Min.   : 300   Min.   : 15.95

 1st Qu.: 4.500   1st Qu.: 725   1st Qu.: 28.00

 Median : 6.000   Median :1000   Median : 39.75

 Mean   : 7.529   Mean   :1815   Mean   : 57.88

 3rd Qu.: 8.000   3rd Qu.:2200   3rd Qu.: 68.63

 Max.   :28.000   Max.   :7500   Max.   :204.62
```


```
> hills
                  dist climb    time
Greenmantle        2.5   650  16.083
Carnethy           6.0  2500  48.350
Craig Dunain       6.0   900  33.650
Ben Rha            7.5   800  45.600
Ben Lomond         8.0  3070  62.267
Goatfell           8.0  2866  73.217
Bens of Jura      16.0  7500 204.617
Cairnpapple        6.0   800  36.367
Scolty             5.0   800  29.750
Traprain           6.0   650  39.750
Lairig Ghru       28.0  2100 192.667
Dollar             5.0  2000  43.050
Lomonds            9.5  2200  65.000
Cairn Table        6.0   500  44.133
Eildon Two         4.5  1500  26.933
Cairngorm         10.0  3000  72.250
Seven Hills       14.0  2200  98.417
Knock Hill         3.0   350  78.650
Black Hill         4.5  1000  17.417
Creag Beag         5.5   600  32.567
Kildcon Hill       3.0   300  15.950
Meall Ant-Suidhe   3.5  1500  27.900
Half Ben Nevis     6.0  2200  47.633
Cow Hill           2.0   900  17.933
N Berwick Law      3.0   600  18.683
Creag Dubh         4.0  2000  26.217
Burnswark          6.0   800  34.433
Largo Law          5.0   950  28.567
Criffel            6.5  1750  50.500
Acmony             5.0   500  20.950
Ben Nevis         10.0  4400  85.583
Knockfarrel        6.0   600  32.383
Two Breweries     18.0  5200 170.250
Cockleroi          4.5   850  28.100
Moffat Chase      20.0  5000 159.833
>
```


```
> pairs(hills)
```



```
> cor(hills)
           dist     climb      time
dist  1.0000000 0.6523461 0.9195892
climb 0.6523461 1.0000000 0.8052392
time  0.9195892 0.8052392 1.0000000
```

```
>  lm(time~dist)
Error in eval(expr, envir, enclos) : Object "dist" not
found
```

```
> attach(hills)
> lm(time~dist)

Call:
lm(formula = time ~ dist)

Coefficients:
(Intercept)         dist
     -4.841        8.330
```


```
> summary(lm(time~dist))

Call:
lm(formula = time ~ dist)

Residuals:
    Min      1Q  Median      3Q     Max
-35.745  -9.037  -4.201   2.849  76.170

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.8407     5.7562  -0.841    0.406
dist          8.3305     0.6196  13.446    6e-15 ***
---
Signif. codes:  0  `***'  0.001  `**'  0.01  `*'  0.05
`.'  0.1  ` '  1

Residual standard error: 19.96 on 33 degrees of freedom
Multiple R-Squared: 0.8456,     Adjusted R-squared: 0.841
F-statistic:  180.8  on  1  and  33  degrees  of  freedom,
p-value: 6.106e-015

>
```


```
> data(shoes)
> shoes
$A
 [1] 13.2  8.2 10.9 14.3 10.7  6.6  9.5 10.8  8.8 13.3

$B
 [1] 14.0  8.8 11.2 14.2 11.8  6.4  9.8 11.3  9.3 13.6

> attach(shoes)
> t.test(A,B)

        Welch Two Sample t-test

data:  A and B
t = -0.3689, df = 17.987, p-value = 0.7165
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 -2.745046  1.925046
sample estimates:
mean of x mean of y
    10.63     11.04
```


```
> T.test(A,B)
Error: couldn't find function "T.test"
> t.test(a,b)
Error in t.test(a, b) : Object "b" not found
> summary(t.test(A,B))
            Length Class  Mode
statistic   1      -none- numeric
parameter   1      -none- numeric
p.value     1      -none- numeric
conf.int    2      -none- numeric
estimate    2      -none- numeric
null.value  1      -none- numeric
alternative 1      -none- character
method      1      -none- character
data.name   1      -none- character
>
>  mean(A)
[1] 10.63
> mean(B)
[1] 11.04
```


1.1.5 Comments on example

- 1:– The first command opened the library of routines and data sets `MASS`. There are many libraries of routines available in R and many can be downloaded from the various R websites listed in §0.1. To find out what libraries are available in your system type `library()` and you will obtain a list of them. To find out what routines are available in [for example] `MASS` type `library(help=MASS)`.

- 2:– The second command `data(hills)` made the data set `hills` available to the session. The base system of R and many of the available libraries come with example data sets for testing routines and for illustrations and `hills` is one of those that come in the library `MASS`. To find out what data sets are currently available to the session type `data()`. It is of course possible to read in data from files, not only ordinary ASCII text files but also files produced by most other packages such as Excel, SAS, SPSS, Minitab, STATA, …… . In addition data can be typed in direct from the keyboard.

- 3:– `summary(hills)` produced a basic summary of the *object* `hills`. Typing `summary(name-of-object)` will produce some sort of summary whatever type of object it is, though what is produced depends on the type of the object (i.e. whether it is a data set or the results of an analysis or whatever.

- 4:– `hills` produced a complete list of the object hills. Typing `name-of-object` will print it out, whatever sort of object it is. Note that this data set consists of three variables: `dist`, `climb`, `time`, and that the rows are labelled with names. These are the record times in minutes taken for ***hill races*** in Scotland. The distance `(dist)` is in kilometres and `climb` gives the total cumulative height in metres climbed in the race.

- 5:– Note the commands `pairs(hills)` and `cor(hills)` are functions operating on the ***object*** `hills`. However, the command `lm(time~dist)` which tried to fit a ***linear model*** of `time` on `dist` did not recognise the objects `dist` (and `time`) until the command `attach(hills)` was given, making the names of the objects (i.e. variables) inside `hills` available to the R-session. Then issuing the command again gives the basic results of fitting a linear model to the relationship between `time` and `dist`, whereas `summary(lm(time~dist))` gives a summary of the ***object*** `lm(time~dist)`.

- 6:– Finally, a further data set, `shoes`, is opened. Given are measures of the wear of shoes of materials `A` and `B` for one foot each of ten boys. Illustrated are the results of a Two Sample t-test of `A` *vs* `B` and reminders that R is ***case-sensitive***.

- 7:– In fact, it would be better to do a ***paired t-test*** on these data, since each boy is wearing material `A` on one foot and `B` on the other and since there is likely to be great differences between the different boys but not between the different feet of individual boys. This can be done by the same function `t.test()` on the differences, i.e. `t.test(A-B)`. In fact `t.test()` is an example of a ***generic function*** (as is `summary()`) whose result depends on the type of argument given to it

```
> t.test(A-B)

        One Sample t-test

data:  A - B
t = -3.3489, df = 9, p-value = 0.008539
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.6869539 -0.1330461
sample estimates:
mean of x
    -0.41
```


1.2 Summary so far

- The aim of this course is to give a flavour of recent developments in applied statistics that have been made possible by the development of a computer language **S** (implemented as the commercial package **S-plus** and as the free language **R**).

- It may seem at first as if the course is more about the computer package **R** than about statistics, but have patience — it really is about statistics.

- **R** is an *object-orientated* language providing facilities for manipulating *objects* such as vectors, matrices, data sets, results of analyses as well as inbuilt statistical procedures and integrated (and interactive) graphical facilities.

- **R** consists of a *base* system supplemented by various *libraries* of routines. Additionally, various standard data sets are included that can be used to illustrate the techniques. The extensive Help System can be used to find out what libraries are available, what each of them contains, what data sets are included and what the data refer to.

- §1.1.4 gives a record of a short **R** session with comments and explanations given in §1.1.5. These contain some key tools for getting started when using the system.