MATH3012: Statistical Methods II – S-Plus Work Sheet 1

1 What is S-Plus?

S-Plus is a statistical programming and analysis language, developed primarily at the AT&T research laboratories during the 1980s. As well as having the usual facilities for statistical modelling, data handling and graphical display, in common with many other statistical packages, S-Plus also allows the user extreme flexibility in manipulating and analysing data. It is the 'package of choice' for many statisticians.

To enable us to exploit the full flexibility of S-Plus, these worksheets will focus on the command-based implementation, whereby one enters S-Plus *directives* directly in the `Commands` window. S-Plus for Windows also utilises *pull-down menus* to perform some of the more common operations. Feel free to investigate these for yourselves.

S-Plus is an *object-oriented* language, which means that everything is stored as a particular type of object, with different operations being appropriate for different types of object. For example, *vectors* and *matrices* are both types of object in S-Plus. Data are usually stored in a *data frame* object, and results of statistical analyses are stored in an object of the appropriate type, for example when you fit a linear model, the results are stored as a *linear model* object.

2 Preliminaries

It is assumed that you are a registered user of the Information Services system, and are familiar with using the Windows machines in the Information Services clusters. Operations you should be familiar with include logging in, accessing programs and files, file handling, using disks, printing, logging out *etc.* If you are unfamiliar with any of these, then Information Services provide a wide range of introductory printed material, available in many of the clusters, to help you.

3 Starting S-Plus

Log-on to the computer and launch S-Plus from `Programs` and then `Statistics`. After starting you will see a large window containing two smaller windows: the `object browser` and the `commands window`. The object browser window is like 'the file manager' and lists all the data you have in your workspace. The default location of the workspace (`C:\users\S-Plus`) can be read from the first line of the commands window. You can switch between windows by **Windows** menu.

The top menubar can be used to perform a variety of tasks, e.g. obtaining summary statistics, fitting models etc. In this course, however, we shall mostly use the commands

window. Type `2+2` in the commands window and see what happens! This window can be used to perform basic arithmetic and thus eliminates the need to have calculators while using S-Plus.

You can obtain help from the `Help` button on the top menu or typing `?<command name>` where command name is the name of the command you want help on, e.g.
`> ?plot` or equivalently `> help(plot)`

Here and subsequently the very first `>`-sign in a command denote the prompt in the commands window, you do not have to type it again. If you do, you will see an error message.

The `args` command prints the list of arguments and defaults which can be passed to a command. For example, type `args(pnorm)`.

If you want to exit S-Plus, type
`> q()`
in the commands window or use select `Exit` off the `File` menu.

4 S-Plus commands and basics

S-Plus commands are always of the form `<commands>(<options>)`. For example, `qnorm(0.975)` gives the 97.5% quantile of the standard normal distribution. In the case where there are no options, e.g. the quit command `q()` you still need to add the brackets. This is because S-Plus thinks of its commands as functions. If you omit the brackets, then S-Plus thinks that you don't want to execute the command but simply want to see what's in it. Type `plot` and see what happens. You, perhaps, do not understand the output, but hopefully you will later.

The assignment operator in S-Plus is given as `<-`, i.e. the less than sign immediately followed by the minus sign (the hyphen), as opposed to the usual '=' symbol which is reserved for something else, we shall see later. For example we may write
`> x <- 2 + 2`
This is read as the object `x` comes from `2+2`.

You can insert comments by first typing the # character. For example, we could have said
`> x <- 2 + 2` # The output should be 4!

You can repeat or edit previous commands by using the up and down arrow keys (↑↓).

5 Course files and workspace

A number of data files, which we shall use during the course, are stored on the University central file server. To access these files, follow the path **Programs** → **System Tools** → **Search For Course Applications**. Highlight `MATH3012 Statistical Methods II` and

click `Install`. The data files are then copied automatically into folder `C:\users\S-Plus` and the windows explorer is launched showing the copied files. The data files are kept there so you can easily access data from S-Plus.

The default workspace is `C:\users\S-Plus`, it links to a `.data` sub-directory though. Type `search()` and see that this is listed in the first position. This will be erased when you log out of the computer. You can use a **permanent workspace** as follows. Suppose that we want to make `H:\math3012` as the permanent workspace for this course. To do this for the first time we follow the path **File** → **Chapters** → **New Working Chapter**. In the dialog box you type in `H:\math3012` and put the label `math3012`. Press ok and it will ask if you would like to create the directory. Upon answering yes you get the desired result; type `search()` again to see if you are successful it should have put math3012 after [1], i.e. the first database is math3012.

All the objects you create from now on will be written on the permanent work space in position 1 of the search path. When you enter S-Plus next time you come through the **File** → **Chapters** → **Attach/Create Chapter** and put `H:\math3012` in position 1. This will retrieve everything in the folder as you left it last time.

6 Import/export data

Probably the best way of entering data into S-Plus is as a *data frame* object. You can create a new data frame by following **File→New** and choosing `Data Set` A rectangular array appears, and you can type in the data directly. For example, you could enter Dataset 1 (welding data) as two columns of 21 observations each. Use the mouse or the arrow keys (←↑↓→) to move around.

There are several ways you can read data into S-Plus. From `File => Import Data` you can import different types of files, e,g. microsoft EXCEL files, MINITAB files etc. Sometimes it is also possible to copy and paste chunks of data from, e.g. EXCEL to an S-Plus data window. See also the top menu `Data`.

Let us illustrate with an example. Install the course files as mentioned in the previous section. Here we will read the data from the file `weld.txt`. You can read it in directly from the file by following **File→Import Data→From File**. Once you hit the `Browse` button you will see two weld data file along with many others, you choose the txt file. (You can choose the other one as well). If you read the data in directly from the file `weld.txt` then the default name for your data frame will be `weld`. Remember to give your data frame an informative name, for future reference. Once the data are imported, the two variables can be given names by double clicking on the grey bar above the first row of data. Call the columns of data `x` and `y`.

The `Export Data` sub-menu from the `File` can be used to save data in different formats.

7 S-Plus data types

The most common data types are as follows.

- **Vectors** are ordered strings of data values. A vector can be one of numeric, character, logical or complex types. For example: `x <- 5:15` puts the numbers 5, 6, ..., 15 in the vector `x`. You can access parts of `x` by calling things like:

  `> x[1]` # Gives the first element of `x`.

  `> x[2:4]` # Gives the elements `x[2], x[3], x[4]`.

  `> x[-(2:4)]` # Gives all but `x[2], x[3], x[4]`.

  There are various command for creating vectors. For example, investigate the vectors produced by the following commands

  ```
  > a1<-c(1,3,5,6,8,21)
  > a2<-seq(1,21,by=2)
  > a3<-seq(min(weld$x),max(weld$x),length=100)
  > a4<-rep(2,12)
  ```

- **Matrices** are rectangular arrays consisting of rows and columns. All data must of the same mode. For example, `y <- matrix(1:9, nrow=3)` puts the numbers 1 to 9 in the matrix `y`. You can access parts of `y` by calling things like:

  `> y[1, 2]` gives the first row second column entry of `y`.

  `> y[1,]` # Gives the first row of `y`.

  `> y[,2]` # Gives the second column of `y`.

  and so on.

- **Data frames** are rectangular arrays where columns could be of different types. For example, type `z <- kyphosis` and bring it up by double clicking on `z` on the object browser. (You can see what these are by typing `?kyphosis`). Columns of data are vectors and are denoted by `(data frame name)$(variable name)`, for example `weld$x` and `weld$y` in the example above. Try

  ```
  > mean(weld$x)
  > var(weld$x)
  > weld$x*weld$y
  > mean(weld$x*weld$y)
  > weld$x^2
  > weld$x-1
  ```

Note that calculations on vectors result in vectors, for example `weld$x*weld$y`, `weld$x^2` and `weld$x-1` above. Results of an S-Plus operation can be stored for future use in S-Plus by assigning the result to a new object using `<-`. For example,

```
> wmx <- mean(weld$x)
> wxy<-weld$x*weld$y
> wxm1<-weld$x-1
```

all create new S-Plus objects with the names given. `wxy` and `wxm1` are vectors of 21 observations, and `wmx` is a scalar.

These objects will now be visible using the `Object Explorer`. An alternative way of examining which objects are available to you is to type

```
> ls()
```

in the `Commands` window. This also lists some things which do not appear in the `Object Explorer`, such as any S-Plus functions you have written (more about that later).

You can also add columns to a data frame, for example

```
> weld$xy<-weld$x*weld$y
```

adds a new column to the data frame `weld`.

- **Lists** are used to collect objects of different types. For example, a list may consist of two data frames and three vectors of different size and modes.

Any unwanted objects can be removed from your disk by using the command `rm`. For example, to remove object `a1` type

```
> rm(a1)
```

Alternatively objects, or individual columns of a data frame, may be removed using the `Object Explorer`.

8 Naming conventions

S-Plus has many reserved words, for example `plot`. You must avoid giving those names to objects you create. If you do, then it will disable the corresponding library function and many things will not work.

You can see if S-Plus has reserved the word by asking it. For example if you wanted to know whether you can use `c` or `D` as your objects simply type it in the commands window. If you get an error message then it has not reserved the word or it is not in your work space. Then you can use the name. If it spits out something that does not look like your own creation then you cannot use it. You will know some reserved words or functions gradually as we learn more.

S-Plus is case sensitive. Hence `x` and `X` are different objects; `plot` is a built-in function but `Plot` is not.

9 Output routing

It is likely that you will want to save your S-Plus output for future use. In this case, it is useful to direct the output of any commands into a *Report* window. To do this, choose **Options→Text Output Routing**. and select `Report` under `Window for Normal Text Output`.

Now, the result of any command you type in the `Commands` Window appears in the `Report` window. You can edit the `Report` window as you work, for example adding annotation or deleting unwanted output. Before you quit S-Plus remember to save the `Report` window as a text (`.txt`) file.

10 Fitting linear models

Fitting linear regression models in S-Plus is quite easy. For example, to fit a simple linear regression to Dataset 1 (welding data) with Diameter (`weld$y`) as the response variable and Current (`weld$x`) as the explanatory variable, type

```
> lm(y~x, data=weld)
```

in the `Commands` window. The output contains a summary of important aspects of the regression model. However, much more information can be obtained by assigning the results of the linear model to an S-Plus object (a linear model object) using

```
> weld.lm1<-lm(y~x, data=weld)
```

The linear model object can then be used within a number of S-Plus commands. Try

```
> resid(weld.lm1)
> coef(weld.lm1)
> plot(weld.lm1)
> deviance(weld.lm1)
> fitted(weld.lm1)
> print(weld.lm1)
> summary(weld.lm1)
> anova(weld.lm1)
```

We may be interested in whether a quadratic model fits the data any better. You can fit a quadratic model by

```
> weld.lm2<-lm(y~x+x^2, data=weld)
```

or

```
> weld.lm2<-update(weld.lm1, . ~ . +x^2)
```

Does the quadratic model provide a significatly better fit? What about a cubic model? For your chosen model, investigate the residual plots.

11 Plotting

S-Plus is particularly flexible at enabling you to get the plot that you want. A simple scatterplot of Diameter against Current for Dataset 1 can be obtained by

```
> plot(weld$x,weld$y)
```

A nicer plot can be obtained by

```
> plot(weld$x,weld$y, xlab="Current", ylab="Diameter", las=1)
```

Can you see what the argument `las=1` has achieved? This is just one of a large number of arguments which can be supplied to an S-Plus graphical routine. To find out others, search for help on the function `par`.

We can add points or lines to this plot by using the commands `points` or `lines` respectively. For example, to put the best fit linear and quadratic curves on the plot, all we need to do is to create corresponding vectors of x and y values and use the `lines` command, as follows

```
> x<-seq(min(weld$x),max(weld$x),length=100)
> y1<-coef(weld.lm1)[1]+(coef(weld.lm1)[2]*x)
> lines(x,y1)
> y2<-coef(weld.lm2)[1]+(coef(weld.lm2)[2]*x)+(coef(weld.lm2)[3]*x*x)
> lines(x,y2,lty=2)
```

The argument `lty=2` ensures that the linear and quadratic curves are distinguishable. Note also that an alternative way of creating `y1` and `y2` here is to use

```
> y1<-predict(weld.lm1,data.frame(x))
> y2<-predict(weld.lm2,data.frame(x))
```

Note that each time S-Plus produces a new plot, it overwrites any existing plots in the current `graphsheet`. This is usually a good idea, as initially a statistical analysis may involve a large number of exploratory plots which you do not want to keep. If you produce a plot which you do want to keep, and which should not be overwritten, then use the command `graphsheet()`. A new `graphsheet` window appears and any subsequent plots will appear here.

If a command produces multiple plots, then the default is that each plot appears on a separate page. However, you can get an array of plots on the same page by using the command `par(mfrow=c(r,c))` where `r` is the number of rows of your array and `c` is the number of columns. For example, `par(mfrow=c(2,3))` produces a 2×3 array of plots. Try typing this, followed by `plot(weld.lm2)`. The residual plots for the quadratic model all appear on a single page.

To get back to one plot per page use `par(mfrow=c(1,1))`

Graphs can be printed by following **File→Print Graph Sheet**. You can also save graphs for future use. They can then be reopened in future sessions using **File→Open**

12 S-Plus on the web

There are many websites around the world offering help with learning S-Plus. The course website `http://www.maths.soton.ac.uk/staff/Sahu/teach/math3012/` links some. Please do have a look. Happy surfing!

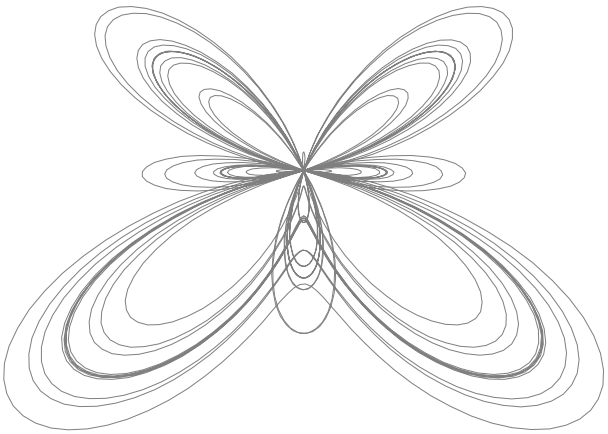

Figure 1: This is drawn using S-Plus! We will see howto later.