Splus/R Introduction

You can quit Splus by typing `q()`

On Unix, Splus can be accessed by typing `splus` or `Splus5 –e` for the newer version. The –e flag will allow you to use emacs commands to scroll and move through text.

The **assignment operator** in Splus is `<-`

As an example, if we wanted to assign the variable `x` the value of 2

We would type: `> x<-2`

Splus will not write to the screen unless prompted to do so. Typing `>x` will prompt Splus to tell you that `x` has been assigned the value 2.

Splus is case sensitive, so the variable `X` is different from the `x` we just assigned.

Programs are written usually as `.q` or `.r` files. In Splus this is not important what you actually name them.

To run them, you will need to type `>source("filename")`

To access help documents about specific Splus functions, one can type `>help(function)` or `?function`

To find functions, you can type `>help.start()`

**Entering vectors:**
`>a<-c(1, 2, 3, 4, 5, 6, 7, 8, 9)`

will assign the variable `a` to a vector of length 9. This could also be accomplished by typing `a<-c(1:9)`

There are many shortcuts like this in Splus. The `c` stands for concatenate.

`>A<-rbind(a,a)` would result in a 2 by 9 element vector where both rows are `a`. There is also a command `cbind`, which would give us the transpose of `A` from our example. `R` is for row, `c` is for column.

To call an element of the matrix, one can type `>A[i,j]` where `i` and `j` are appropriate integers.

For a vector, one need only enter one argument.

**Matrix operations:**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
</tr>
<tr>
<td><code>%*%</code></td>
<td>Multiplication</td>
</tr>
<tr>
<td><code>t()</code></td>
<td>Transpose</td>
</tr>
<tr>
<td><code>%/%</code></td>
<td>Division</td>
</tr>
</tbody>
</table>

To take the inverse of a matrix `A` in Splus, we can use the `>solve(matrix)` command.

The default in Splus is to do element-wise operations. If we wanted to multiply the two arrays of the same size element-wise, say `A` and `B`, we can enter `>A*B`

**Relations:**

<table>
<thead>
<tr>
<th>Relation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;</code></td>
<td>Less than</td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td>Greater than</td>
</tr>
<tr>
<td><code>&lt;=</code></td>
<td>Less than or equal</td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>Greater than or equal</td>
</tr>
<tr>
<td><code>==</code></td>
<td>Equal</td>
</tr>
<tr>
<td><code>!=</code></td>
<td>not equal.</td>
</tr>
</tbody>
</table>

**Logical Operators:**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&amp;</code></td>
<td>and</td>
</tr>
<tr>
<td>`</td>
<td>`</td>
</tr>
<tr>
<td><code>~</code></td>
<td>not.</td>
</tr>
</tbody>
</table>

These are useful when using loops.

**Loops** that require statements of more than one line must be enclosed by brackets.

For, while, and if loops are the most common types of loops.
>x<-rep(0,10)  \hspace{1cm} \text{(initialize array x as a vector of length 10 with entries 0)}
>for (i in 1:10)
>  { \hspace{1cm} \text{would result in a vector of powers of 2, namely 2 4 8 16…We did not need brackets here since our statement was only one line, but they were included for illustrative purposes.}}
>x[i]<-2^i;
>
would result in a vector of powers of 2, namely 2 4 8 16…We did not need brackets here since our statement was only one line, but they were included for illustrative purposes.

while loops take on the form
>while relation
>  statement

The following will compute the smallest nonnegative integer n such that $2^n > a$, where a is some fixed constant.
>n<-0;
>while (2^n<a)
>  n<-n+1;
>n

As an example of an if statement, the following will check whether a given integer g is even. If it is it will add one to its value.
>if  \hspace{0.1cm} ((g\%\%2)==0) \hspace{0.1cm} \text{(modulus operator)}
>  g<-g+1;

Sub-matrices: As an example, say we wanted to remove the upper left two by two sub-matrix from a matrix X, and name it Y. The command would be
>Y<-X[1:2, 1:2]

If we wanted to assign the variable z the first column of X, we would type
>z<-X[1,]

Without specific arguments after the comma, Splus will take all the entries in the respective row or column.

Random Arrays:
The Splus command `runif(j)` will result in an array of random numbers between 0 and 1 of length j. If we need a vector of random numbers between 1 and 3, we could enter `runif(j,1,3)`.

Graphs: Commands such as plot, scatter, and hist are available to graph data. More information about these commands can be found using the help command. If you are using Splus on a Unix machine, you will need to open a graphics window before graphing. This is accomplished by the command `motif()`

Saving Sessions: Splus automatically saves your session, meaning all the variables you have created.

Recording Sessions: The command `sink name` will record the session into the file `name`. Typing `sink` will end the record.

Reading Data: The function `read.table(file_name, header=T)` can be used to read data into Splus. If your file does not include a header (variable names), then you would want to enter header=F as the second argument. To find out what the names of your variables are, you can type `names(array)` To use the columns of the array separately, you can use `attach(array)` As an example, say we have an array named data that we have read in. data contains variables V1 and V2. If you wanted to plot a histogram of V1, you can either first attach the data frame, by typing `attach(data)` followed by `hist(v1)` or alternatively you can simply type `hist(data$V1)` without using the attach command. The dollar sign lets us use various parts of a data frame. This especially comes in handy when we are using some of the statistical functions such as regression.

Useful Commands:
The command `ls()` will list all of the variables in the current work space

Useful References:
There are numerous Splus primers available on the world wide web. The site [http://www.ci.tuwien.ac.at/R/](http://www.ci.tuwien.ac.at/R/) is recommended.