

[Page One](#)[Campus  
Computing News](#)[Renew PRAS  
Accounts Now](#)[Accessing  
GroupWise E-  
mail From Home](#)[SPSS on CMS to  
be Dropped](#)[The UNIX  
Question](#)[Computer  
Virus!!!!](#)[Going  
Somewhere?](#)

RSS Matters

[The Network  
Connection](#)[List of the Month](#)[WWW@UNT.EDU](#)[Short Courses](#)[IRC News](#)[Staff Activities](#)

# RSS Matters

By [Rich Herrington](#), Research and Statistical Support Services

## Introduction to S-Plus 5.1 on UNIX (SOL)

The Research and Statistical Support group (RSS) has recently upgraded S-Plus on SOL from version 5.0 to version 5.1. In this column we will cover basic operations using S-Plus on SOL. This is part one of a three part series on using Splus on SOL.

### Connecting to S-Plus

In order to view graphical output on SOL a graphics device needs to be opened. You will need to have a x-window open or be using x-window emulator software. The College of Arts and Sciences (CAS) currently runs X-Win32 on the GAB, Terrell Hall, and Wooten Hall Servers. After starting X-Win32 on your local desktop, and before entering S-PLUS, the UNIX display needs to be routed to your IP address. To do this from the UNIX prompt, type:

```
xterm -ls -display 129.120.56.150:0 &
```

where the ipaddress is the address assigned to your machine (i.e. it is a number like 129.120.56.150:0). In WIN95/98 one can run the program "winipcfg" to get the ipaddress for your desktop PC. Before starting S-Plus, it would be a good idea to create a directory where all the data objects created in Splus can be stored. From the Unix prompt:

```
mkdir projects  
cd projects
```

Now create a chapter file within this directory:

```
Splus5 CHAPTER
```

This creates the hidden file ".Data" in this directory. This file will hold all of the objects created during your S-Plus session. This includes all objects that are created: scripts, data, graphs, etc. You can have multiple directories with different ".Data" files. Once the chapter file is created, S-Plus is invoked at the Unix prompt:

```
Splus5
```

You will see

```
S-PLUS : Copyright (c) 1988, 1999 MathSoft, Inc.  
S : Copyright Lucent Technologies, Inc.  
Version 5.1 Release 1 for Sun SPARC, SunOS 5.5 : 1999  
Working data will be in .Data  
> █
```

### Importing Data Files Into S-Plus

One method of getting data into S-Plus for data analysis and graphing is to import the data file. The S-Plus function that allows one to do this is the "importData" function.

Using the importData function, S-Plus can import from the following file types: ASCII, dBase, Microsoft Excel, FoxPro, Gauss, Informix, Gauss, Lotus, MatLab, Oracle, Quattro Pro, windows SPSS .sav and SPSS .por, windows SAS .sd2 and SAS .tpt, .xpt, STATA, Sybase, and Systat. You

can read a windows SPSS data file into S-Plus using the following:

```
data<-importData("data.sav")
```

The assignment operator "<-" assigns the output of the importData function into an objects named "data". To display the contents of your working directory use the objects function:

```
objects()
```

To display the contents of a data object, type the name of the object and hit return:

```
data
```

You see the following:

```
> data<-importData("data.sav")
> objects()
[1] ".Last.value"      ".Messages"      ".Random.seed"
[4] "atoq.1.46.dat"    "atoq.1.46b.dat" "atoq.col.100.sum"
[7] "atoq.col.200.sum" "atoq.col.300.sum" "atoq.col.400.sum"
[10] "atoq.col.457.sum" "atoq.col.sum"    "atoq.dat"
[13] "atoq2.dat"        "atoqprob.2.47.dat" "atoqprob.2.47b.dat"
[16] "atoqprob.dat"     "data"            "last.dump"
[19] "prim4"            "prim4.fac"       "rawprop.dat"
[22] "sum2"             "x"               "x.dat"
[25] "x.sum"
> data
  var00001 var00002 var00003
1         1         2         3
2         4         5         6
3         7         8         9
> █
```

## Using S-Plus in Batch Mode

An alternative to interactive processing in S-Plus is the "batch" processing of S-Plus scripts. This is accomplished by creating a file that contains S-Plus commands which is then submitted for processing. S-Plus has a command allowing one to shell out to the UNIX operating system. Upon completion, control reverts back to the interactive S-Plus session. Thus, one can edit a file and submit it for processing without exiting the S-Plus session. For example, one can issue the following UNIX commands at the UNIX prompt or use the exclamation mark (!) to "shell" out to the UNIX environment whereby the command is executed. For example:

```
richherr@sol /projects/sol/richherr % Splus5
S-PLUS : Copyright (c) 1988, 1999 MathSoft, Inc.
S : Copyright Lucent Technologies, Inc.
Version 5.1 Release 1 for Sun SPARC, SunOS 5.5 : 1999
Working data will be in .Data
> !pico example.ssc █
```

This invokes the "Pico" editor within the S-Plus session.

```

UW PICO(tm) 2.9          File: example.ssc          Modified
x<-rnorm(10)
x
█

[ Read 3 lines ]
^G Get Help   ^O WriteOut  ^R Read File  ^Y Prev Pg   ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify   ^W Where is   ^W Next Pg   ^U UnCut Text ^T To Spell

```

A file is created (example.ssc) which contains: 1st line - an assignment of 10 random numbers to the data object "x", 2nd line – the contents of "x" is listed. The file is saved by the key sequence "Ctrl-O", "Ctrl-X". Control reverts back to the S-Plus session. Once back in the S-Plus session. The file can then be submitted for batch processing. Once submitted the output can viewed within the S-Plus session using the UNIX command "more". For example:

```

> !Splus5 BATCH example.ssc example.out
> !more example.out
S-PLUS : Copyright (c) 1988, 1999 MathSoft, Inc.
S : Copyright Lucent Technologies, Inc.
Version 5.1 Release 1 for Sun SPARC, SunOS 5.5 : 1999
Working data will be in .Data
> x<-rnorm(10)
> x
 [1] 0.2718807 -0.2354967 0.1925765 -0.3835263 -0.4013729 0.1724819
 [7] 0.5381853 1.6538538 -0.8547855 0.1699303
>
> █

```

## Other Useful S-Plus Operations

### The History Function

The "history" function allows one to pull up a listing of commands that have been utilized during the current S-Plus session. In an S-Plus session:

```
> history(1)
1: X <- mvrnorm(100, 2, S = diag(rep(1 - 0.2, 2)) + 0.8)
2: X <- mvrnorm(nt, p = 4, S = diag(rep(1 - rho, 4)) + rho)
3: nt <- 100
4: X <- mvrnorm(nt, p = 4, S = diag(rep(1 - rho, 4)) + rho)
5: X <- mvrnorm(nt, p = 4, S = diag(rep(1 - rho, 4)) + rho)
6: X <- mvrnorm(nt, p = 4, S = diag(rep(1 - rho, 4)) + rho)
7: mvrnorm <- function(n, p = 1, u = rep(0, p), S = diag(p))
{
      Z <- matrix(rnorm(n * p), p, n)
      t(u + t(chol(S)) %** Z)
}
8: X <- mvrnorm(100, p = 4, S = diag(rep(1 - rho, 4)) + rho)
9: X <- mvrnorm(10000, p = 4, S = diag(rep(1 - rho, 4)) + rho)
10: rnorm(100000)
Selection: |
```

The history function displays the last 10 commands used during the S-Plus session. Other uses of the history function include:

```
history()
      # puts up a menu of last 10 expressions; choose one to evaluate
history(max=20) # choose one from the last 20
history("plot") # recent expressions containing word plot
history(plot)
history(call="plot", max=1) # re-evaluate the last call to plot
history(call=plot, max=1)
history(x, max=1, editor=T)
      # re-evaluate the last expression containing x, but let me edit
      # it first.
history(menu=F, eval=F, reverse=F, file="S.exprs")
      # write the last 10 expressions to a file in chronological order
history(call=f, eval=F, menu=F)
      # print but do not evaluate recent calls to f
again() # re-evaluate the last expression
value <- again(editor=T) # edit and then re-evaluate last expression;
      # assign the result under value
again(plot) # re-evaluate the last expression containing word plot
```

### **The Scan Function**

The scan() function can be used to read in numeric data into a matrix. Suppose the file "data2" contains data which is stored in 3 columns. The data would be read in as follows:

```
> data2.mat<-matrix(scan("data2"),ncol=3, byrow=T)
```

To read in character data, the argument what=character() must be specified:

```
>data2.mat<-matrix(scan("data2"),ncol=3, what=character(), byrow=T)
```

To quit from S-Plus use the following at the Splus prompt:

```
>q()
```

### **Next time**

Next time we'll cover S-Plus data structures, indexing operations for these data structures, and data subsetting. In the final installment we will cover programming constructs (e.g. writing functions, loops, branching operations).