RSS Matters

By Rich Herrington, Research and Statistical Support Services

Exploring S-Plus 5.1 on UNIX (SOL) - Part II

In the August 1999 issue of RSS matters we provided an introduction to using S-Plus on SOL (http://www.unt.edu/benchmarks/archives/1999/august99/rss.htm). In this issue we continue our exploration of S-Plus on SOL.

Constructing Expressions in S-Plus

To begin our session we must invoke S-Plus after logging onto SOL. Type "Splus5" at the UNIX prompt to start the S-Plus session. You should see the following screen:

```
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
>
```

S-Plus expressions are typed in at the ">" prompt. S-Plus will print out the results of the evaluation once the "Enter" key is pressed:

```
> 2+2
[1] 4
> sin(pi)
[1] 1.224647e-16
> sqrt(1000)
[1] 31.62278
>
```

An incomplete expression will lead to a second prompt, "+". You can continue with your expression at the second prompt:

```
> sqrt(
+ 100)
[1] 10
>
```

If the “+” prompt continues after pressing "Enter", then enter many “)” to get the “>” prompt back again. Then start your expression once again:

```
> sqrt(
+
+ )))))
Problem: Syntax error: No opening parenthesis before unbalanced (")")  on
        input line 3
>
```

Scalars and Assignments

The assignment operator is the sequence of characters, "<" (less than) and "-" (hyphen). Assigning the variable “weight” the value of 190 we use the following:

```
> weight<-190
> weight
[1] 190
>
```

Character values are inserted in quotes. If the quotes are omitted, S-Plus will look for a possibly non-existent data object called "Jim" to assign to the variable "person". The result is not printed until you enter the object name:

```
> person
[1] "Jim"
>
```

Vectors

The function "rnorm( )", returns a vector of random deviates from the normal distribution The "[n]" on the left shows where the row starts:

```
> rnorm(10)
 [1] -0.63147304  1.25447805 -0.84064508 -0.36729337  0.09650417 -0.76198708
 [7]  0.96427688 -2.32446837  0.10866023  0.73403810
> 
```

A single number is a vector of length 1. We can make vectors using the concatenation function, "c( )". Then we can assign the integers 1,2,3 to the vector x:

```
> mean(rnorm(10))
[1] -0.240037
> x<-c(1,2,3)
> x
[1] 1 2 3
>
```

We can create a vector of names. Also we can create a vector of sequential integers using the function, "a:b", where a is the starting integer and b is the ending integer:

```
> people<-c("Jim", "Sue", "Dave")
> people
[1] "Jim"  "Sue"  "Dave"
>
> seqvar<-5:10
> seqvar
[1]  5  6  7  8  9 10
> 
```

Object Names

Object names may contain letters, "abcDEF", or numbers, "0123456789", or a dot, ".". Examples of valid names: height, weight, x.var, .yvar, x.y.var, or x110. Objects names cannot use an underscore, a hyphen, begin with a number, or use reserved symbols. Examples of invalid object names: _xvar, y_var, x-yvar, 120xvar, T, F, or NA.

Handling Objects

We can list out all of the objects in our workspace:

```
> objects()
 [1] ".Last.value"          ".Random.seed"         ".nfs0788"
 [4] "X"                    "last.dump"            "mvrnorm"
 [7] "n"                    "nt"                   "nval"
[10] "people"               "person"               "poprho"
[13] "rcrit.crit"           "rcrit.pred"           "rho"
[16] "rpred.crit"           "rpred.pred"           "seqvar"
[19] "sim"                  "tabfid.cancor"        "tabfid.cancor.eigen"
[22] "tabfid.cancov.crit"   "tabfid.cancov.pred"   "tabfid.cor"
[25] "tabfid.cor.rob"       "tabfid.crit"          "tabfid.dat"
[28] "tabfid.pred"          "weight"               "x"
>
```

Objects remain until removed, even if one quits S-Plus:

```
> rm(x)
> x
Problem: Object "x" not found
>
```

Objects as Variables

Objects can be used in expressions:

```
> x<-1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> mean(x)
[1] 5.5
> y<-c(x, 10)
> y
 [1]  1  2  3  4  5  6  7  8  9 10 10
> length(y)
[1] 11
> 2*y
 [1]  2  4  6  8 10 12 14 16 18 20 20
>
```

Vector Arithmetic

Scalar Functions work on an element-wise basis. It is also possible to perform scalar and vector arithmetic:

```
> x<-1:5
> x^2
[1]  1  4  9 16 25
>
> 2*x
[1]  2  4  6  8 10
>
```

Logical Vectors

Expressions with relational operators return logical vectors, "T" is True, "F" is False:

```
> x<-rnorm(10)
> x
 [1] -0.4022230 -0.3696861 -1.8830429  1.6202351  0.4653652  1.5345344
 [7] -0.6967635  0.8779519  0.3089322 -1.0294022
> x<0
 [1] T T T F F F T F F T
>
```

Missing Values

A missing value is represented by "NA". Operations on NA return NA. The function is.na( ) checks for missing values:

```
> x<-c(1, NA, 3)
> x
[1]  1 NA  3
> x+1
[1]  2 NA  4
> sum(x)
[1] NA
> is.na(x)
[1] F T F
>
```

Vector Indexing

S-Plus uses brackets, [ ], to select elements of a vector. Negative indices remove elements:

```
> x<-c(2,4,6,8,10)
> x
[1]  2  4  6  8 10
> x[1]
[1] 2
> x[3:5]
[1]  6  8 10
```

```
> x[c(1,2,3)]
[1] 2 4 6
> x[-c(1:3)]
[1]  8 10
> 
```

Logical Indices

A logical index selects elements. Symbols for the logical operators are: "<" (less than), ">" (greater than), "<=" (less than or equal to), ">=" (greater than or equal to), " = =" (equal to), "!" (negation operator), "!=" (not equal to).

```
> x<-rnorm(5)
> x
[1] -0.6932126 -0.2386601  1.0713995  0.1983262  1.0289510
>
> x[x<=0]
[1] -0.6932126 -0.2386601
>
```

Replacement

You can use [ ] on the left hand side of an assignment, "<-" :

```
> x<-sample(1:8)
> x
[1] 6 4 7 1 2 3 5 8
> x[2]<-NA
> x
[1]  6 NA  7  1  2  3  5  8
```

Next Time

Next time we will cover matrices, arrays, and lists, among other topics. Good luck with S-Plus!