Introduction to S-PLUS

Jonathan D. Courtney
(courtney@pixel.cps.msu.edu)

Pattern Recognition & Image Processing Laboratory
Department of Computer Science
Michigan State University

January 25, 1997

1 Introduction

S-PLUS is an interactive tool for data analysis including plotting, projection, clustering, and other operations that are useful in exploratory data analysis and pattern recognition. It also provides a XWindow-based graphical interface and extensive help system.

To use S-PLUS, you must be running XWindows (either X11 or OpenWindows). Make sure your PATH includes /soft/prip/bin/ (Splus executable is in /soft/prip/bin/Splus).

1.1 Getting Started

1. Start S-PLUS. (Again, note the capitalization.)
   
   % cd $  
   % Splus  
   
   You will get a $ prompt in the terminal window, and S-PLUS will start a help window and an output window.

2. Configure S-PLUS. Create an initialization function named .First() to be run upon startup.
   
   > .First <- function() {
   + options(gui = "motif")
   + help.start()
   + motif()
   }  

   This will create a file named .First in a directory called .Data under your home directory.

   options() Sets the GUI to be Motif-like.
   help.start() Starts the online S-PLUS help system.
   motif() Starts an S-PLUS graphical output window.

3. Try a simple plot. The following example plots Fisher’s Iris data set, using the first two features of the patterns. There are 150 patterns and 4 features: the first 50 patterns are of the Setosa species, the next 50 are of the Versicolor species, and the last 50 are of Virginia.
   
   > x <- iris[,1,]  
   > y <- iris[,2,]  
   > labels <- c(rep("S", 50), rep("C", 50), rep("V", 50))  
   > plot(x, y, type="n", xlab="First feature", ylab="Second feature")
Functions:

iris  Fisher's Iris dataset, organized as an S-PLUS data frame. There are three dimensions to the data frame: the first is the pattern number, the second is the feature number, and the third is the pattern class. Thus, iris[2, 1, 3] returns the sepal length of the second pattern of the Virginica species. Furthermore, iris[n, 1] returns the n'th feature of all the patterns.
c()  Concatenates objects into a list.
rep()  Replicates objects.
plot()  Draws data points to the output window; x and y give the X and Y coordinates of the data, respectively, type="n" provides for the points to be labeled, and xlab and ylab are the X and Y coordinate axes labels, respectively.
text()  Writes the pattern labels to the plot.

To read data from a file, use scan():

> !tail +4 /user/prip/data/gdata/iris > /tmp/iris
> data <- matrix(scan("/tmp/iris"), ncol=4, byrow=T)
> !rm /tmp/iris

!command  Executes command in a Unix shell. Note that data files in PRIP Lab standard format must have the first 3 header lines removed for use in S-PLUS.

scan()  Reads in the data file.
matrix()  Creates a 2D matrix from the data. The parameter ncol specifies the number of columns of the matrix (in this case, corresponding to features), and byrow=T specifies that the patterns are given row-by-row in the file.

4. Print the plot. Select Printing... from the Options menu in the S-PLUS plot window. Specify the correct printer using the -P printer option in the lpr command sequence, and then select Print to output the plot. See man lpr(1) for more information.

5. Make a function. You can save the previous S-PLUS code as a function to be executed or edited later, using the procedure function().

> IrisPlot <- function() {
+ x <- iris[, 1,]
+ y <- iris[, 2,]
+ labels <- c(rep("S", 50), rep("C", 50), rep("V", 50))
+ plot(x, y, type="n", xlab="First feature", ylab="Second feature")
+ text(x, y, labels)
+ }

Now the code is stored as a function called IrisPlot. To view the code, do

> IrisPlot

and the code will be printed to the screen. To edit the code, type

> IrisPlot <- vi(IrisPlot)

which will invoke the vi editor on the function. User-created functions are stored in your .Data directory. To run the function, type

> IrisPlot()

Functions may take arguments, as in
> fscanf <- function(file) {
+ scan(file)
+ }

Finally, note that the file .First in the .Data directory is also a function (named .First()), and may be edited to customize your S-PLUS configuration.

6. Quit S-PLUS.
> q()

2 Examples

The following example functions perform some useful data analysis operations. You are encouraged to consult the online S-PLUS help facility for detailed information on the commands used.

2.1 Viewing Multidimensional Data

> MultiDim
function()
{
data <- rbind(iris[,1], iris[,2], iris[,3])
brush(cbind(data, rep(1:3, c(50, 50, 50))))
}

To exit brush(), press the Quit box in the output window.

2.2 Chernoff Faces

> ChernoffFaces
function()
{
data <- rbind(iris[,1], iris[,2], iris[,3])
faces(data, head = "Iris Data")
}

2.3 Principal Component Analysis

> PrincipalComponents
function()
{
data <- rbind(iris[,1], iris[,2], iris[,3])
labels <- c(rep("S", 50), rep("C", 50), rep("V", 50))
comps <- prcomp(data)
comp1 <- comps$x[, 1]
comp2 <- comps$x[, 2]
plot(comp1, comp2, type = "n", main = "Iris data, Principal Components", xlab = "First", ylab = "Second")
text(comp1, comp2, labels)
}
2.4 Discriminant Analysis

> DiscriminantAnalysis
function()
{
data <- rbind(iris[,1], iris[,2], iris[,3])
dis <- descr(data, 3)
dv <- data $%*% dis$vars
x <- dv[, 1]
y <- dv[, 2]
lab <- c(rep("S", 50), rep("C", 50), rep("Y", 50))
plot(x, y, type = "n", main="Iris data, Discriminant Analysis", xlab = "First discriminant variable", ylab = "Second discriminant variable")
text(x, y, lab)
}

2.5 Hierarchical Clustering

2.5.1 Single Link

> SingleLink
function()
{
data <- rbind(iris[,1], iris[,2], iris[,3])
h <- hclust(dist(data), method = "connected")
plclust(h, main = "Iris data, Single Link")
}

2.5.2 Complete Link

> CompleteLink
function()
{
data <- rbind(iris[,1], iris[,2], iris[,3])
h <- hclust(dist(data), method = "complete")
plclust(h, main = "Iris data, Complete Link")
}

2.6 Partitional Clustering

> KMeans
function()
{
data <- rbind(iris[,1], iris[,2], iris[,3])
centers <- rbind(data[1, ], data[51, ], data[101, ])
km <- kmeans(data, centers)
plot(data[, 1], data[, 2], type = "n", main = "Iris data, K-Means Clustering")
text(data[, 1], data[, 2], km$cluster)
}
3 Additional Information

For more information, check the online help facility available with S-PLUS, or try the *S-PLUS User's Manual* volumes I and II, found in the PRIP laboratory.