

# An Introduction to S-PLUS for Windows

by

Shuming Bao, China Data Center

TEL: (313)647-9610(O), EMAIL: sbao@umich.edu

## I. About S-PLUS

- An enhanced version of S language specially for exploratory data analysis and statistics.
- An integrated suite for data manipulation, data analysis and graphical display.
- An interpreted language, in which individual language expressions are read and then immediately executed.
- Object-oriented programming (method, class, and object)  
Example: draw.circle.
- Programming environment: the Commands windows and Script windows.

## II. Syntax of S-PLUS Expressions (case sensitive):

```
> sqrt(x)

> plot(corn.rain)

> a <- 7

> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m"
[14] "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"

> letters[3]
[1] "c"

> letters[-3]
[1] "a" "b" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m"
[14] "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"

> i <- 1:10
> i
[1] 1 2 3 4 5 6 7 8 9 10

> letters[ i < 3 ]
[1] "a" "b" "c"
```

## III. Data Objects

### *Data Modes:*

- **logical:** The values T (or TRUE) and F (or FALSE).
- **numeric:** Floating-point real numbers, integers, decimal fractions, or in scientific notation.
- **complex:** Complex numbers of the form  $a + bi$  ( $3+1.23i$ ).
- **character:** enclosed by matching double quotes (") or apostrophes ('), such as "Alabama".

```
> mode(car.gals)
[1] "numeric"
```

### *Basic types of data objects in S-PLUS:*

1. vector - a ordered set of values.

```
> a <- 1:10
```

## 2. matrix

```
> m1 <- matrix(1:12,ncol=3,byrow=T)
[,1] [,2] [,3]
[1,] 1 2 3
[2,] 4 5 6
[3,] 7 8 9
[4,] 10 11 12
```

## 3. array - a matrix with more than two dimensions.

```
> array(c(1:8,11:18,111:118), dim=c(2,4,3))
, , 1
[,1][,2][,3][,4]
[1,] 1 3 5 7
[2,] 2 4 6 8
, , 2
[,1][,2][,3][,4]
[1,] 11 13 15 17
[2,] 12 14 16 18
, , 3
[,1][,2][,3][,4]
[1,] 111 113 115 117
[2,] 112 114 116 118
```

## 4. list - a list of *components*, where each component can be a data object of different data types.

```
> grp <- c(rep(1,11),rep(2,10))
> heart.list <- list(group=grp, thw=thw, descrip="heart data")
> heart.list
$group:
[1] 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
$descrip:
[1] "heart data"
```

## 5. factor - categorical data

```
> classlist <- c("male", "female", "male", "male", "male",
+ "female", "female", "male", "female", "male")
> factor(classlist)
[1] male female male male male female female male
[9] female male
> levels(classlist)
[1] "male" "female"
```

## 6. time series - Defines a univariate or multivariate time series.

```
> freeny.x
lag quarterly revenue price index
[1,] 8.79636 4.70997
[2,] 8.79236 4.70217
[3,] 8.79137 4.68944
[4,] 8.81486 4.68558
[5,] 8.81301 4.64019
[6,] 8.90751 4.62553
[7,] 8.93673 4.61991
[8,] 8.96161 4.61654
[9,] 8.96044 4.61407
```

```
[10,]          9.00868      4.60766

> ts(freeny.x, start = 1962.25, frequency=4)
      lag quarterly revenue price index
1962.25      8.79636      4.70997
1962.50      8.79236      4.70217
1962.75      8.79137      4.68944
1963.00      8.81486      4.68558
1963.25      8.81301      4.64019
1963.50      8.90751      4.62553
1963.75      8.93673      4.61991
1964.00      8.96161      4.61654
1964.25      8.96044      4.61407
1964.50      9.00868      4.60766
```

7. data frame - generalized matrices that allow a mix of columns with different data modes.  
 > sample(row.names(solder),10)

```
380 L Thick A3 L7 2 0
545 L Thick B3 D4 2 0
462 L Thin  A3 D6 3 3
809 S Thick B6 L9 2 7
609 S Thick B3 L4 3 19
492 M Thin  A6 D6 3 8
525 S Thin  A6 L6 3 18
313 M Thin  A3 L6 1 1
408 M Thick A6 D7 3 11
540 S Thin  A6 L9 3 22
```

Combining data frames *by column*.

```
> oa.4.2p3
A B C
1 A1 B1 C1
2 A1 B2 C2
3 A2 B1 C2
4 A2 B2 C1

> run1 <- cbind(oa.4.2p3, resp=c(46, 34, 44, 30))
> run1
A B C resp
1 A1 B1 C1 46
2 A1 B2 C2 34
3 A2 B1 C2 44
4 A2 B2 C1 30
```

Combining data frames *by row*:

```
> rand.df1
norm unif binom
1 1.64542042 0.45375156 41
2 1.64542042 0.83783769 44
3 -0.13593118 0.31408490 53
4 0.26271524 0.57312325 34
> rand.df2
norm binom chisq
1 0.3485193 50 19.359238
2 1.6454204 41 13.547288
3 1.4330907 53 4.968438
4 -0.8531461 55 4.458559
```

```

> rbind(rand.df1[,c("norm","binom")],
+ rand.df2[,c("norm", "binom")])
norm binom
norm binom
1 1.64542042 41
2 1.64542042 44
3 -0.13593118 53
4 0.26271524 34
5 0.34851926 50
6 1.64542042 41
7 1.43309068 53
8 -0.85314606 55

```

Merging (or *joining*) data frames:

```

> authors
FirstName LastName Age Income Home
1 Lorne Green 82 1200000 California
2 Loren Blye 40 40000 Washington
3 Robin Green 45 25000 Washington
4 Robin Howe 2 0 Alberta
5 Billy Jaye 40 27500 Washington

```

```

> books
AuthorFirstName AuthorLastName Book
1 Lorne Green Bonanza
2 Loren Blye Midwifery
3 Loren Blye Gardening
4 Loren Blye Perennials
5 Robin Green Who_dun_it?
6 Rich Calaway Splus

```

```

> merge(authors, books, by.x=c("FirstName", "LastName"),
+ by.y=c("AuthorFirstName", "AuthorLastName"))
FirstName LastName Age Income Home Book
1 Loren Blye 40 40000 Washington Midwifery
2 Loren Blye 40 40000 Washington Gardening
3 Loren Blye 40 40000 Washington Perennials
4 Lorne Green 82 1200000 California Bonanza
5 Robin Green 45 25000 Washington Who_dun_it?

```

```

> attributes(auto)
$names:
[1] "Price" "Country" "Reliab" "Mileage" "Type"
$row.names:
[1] "AcuraIntegra4" "Audi1005" "BMW325i6"

```

```

$class:
[1] "data.frame"

```

#### **IV. Data Import and Export**

*Input/Export data type:* ACCESS, ASCII, DBASE, EXCEL, GAUSS, LOTUS, MATLAB, ODBC, PARADOX, QUATTRO, SAS, SAS\_TPT, SPLUS, SPSS, SPSS\_POR, STATA, SYSTAT.

*Loaded S-PLUS Object:* data frame.

*Export graph Type:* BMP, CGM, DXF, EPS TIFF, EPS, EPS PRINT, GIF, HGL, IMG, JPG, MET, PCL, X, PICT, SDW, TIFF, TGA, WMF, WPB, WPV.

```

export.data(FileName, FileType, DataSet, ...)
import.data(FileName, FileType, DataFrame, ...)
export.graph(Name, FileName, ExportType, ...)

```

## **Functions and Operators**

Components of S-PLUS functions: reserved word *function*, an *argument list* (which may be empty), and a *body*.

Ex:

```
> sqrt <- function(x) {x^0.5}
```

To create or edit a new function, call `fix function` as follows:

```
> fix(newfunc)
```

### ***Operators in S-PLUS***

\$	component selection
[ []	subscripts, elements
^	exponentiation
-	unary minus
:	sequence operator
%% %/% %*%	modulus, integer divide, matrix multiply
* /	multiply, divide
+ -	add, subtract
<> <= >= == !=	comparison
!	not
&   &&	and, or
~	formulas
<<- -> <- _	assignments

### ***Elementary Functions.***

sqrt	square root
abs	absolute value
sin, cos, tan	trigonometric functions
asin, acos, atan	inverse trigonometric functions
sinh, cosh, tanh	hyperbolic trigonometric functions
asinh, acosh, atanh	inverse hyperbolic trigonometric functions
exp, log	exponential and natural logarithms
log10	common logarithm
gamma, lgamma	gamma function and its natural logarithm
ceiling	closest integer not less than element
floor	closest integer not greater than element
trunc	closest integer between element and zero
round	closest integer to element
signif	round to specified number of significant digits
cumsum, cumprod	cumulative sum and product

### ***Summary Functions.***

min, max	Return, respectively, the smallest and largest values in their arguments.
range	Returns a vector of length two containing the minimum and maximum of all the elements in all its arguments.

mean, median	Return, respectively, the arithmetic mean and median of their arguments. An optional argument to mean, trim, allows you to discard a specified fraction of the largest and smallest values.
quantile	Returns user-requested sample quantiles for a given data set.
var	Returns the variance of a vector, the variance-covariance of a data matrix, or covariances between matrices or vectors.
cor	Returns correlation matrix of a data matrix or correlations between matrices or vectors.

**Logical and Comparison Operators.**

==	equal to	!=	not equal to
>	greater than	<	less than
>=	greater than or equal to	<=	less than or equal to
&	vectorized And		vectorized Or
&&	control And		control OR
!	not		

**S-PLUS constructions.**

if (cond) expression	Evaluates <i>cond</i> ; if true, evaluates <i>expression</i> .
if (cond) expr else condexpr condexpr.	Evaluates <i>cond</i> ; if true, evaluates <i>expr</i> . If false, evaluates <i>condexpr</i> .
ifelse(cond, expr1,expr2)	The ifelse function is a <i>vectorized</i> version of the if statement. It evaluates the condition and returns elements of <i>expr1</i> for TRUE elements and elements of <i>expr2</i> for FALSE elements.
switch(expr, ... )	Evaluates <i>expr</i> , which must evaluate to either character or numeric. The value of <i>expr</i> is compared to the remaining arguments. If it matches one of these arguments exactly, the value of the evaluated argument is returned as the value of switch.
break	Terminates current loop and passes control out of the loop.
next	Terminates current iteration and immediately starts next
iteration of the loop.	
return(expr)	Terminates current function and immediately returns the
value of <i>expr</i> .	
stop(message)	Signals an error condition by terminating evaluation of the current function, printing the character string " <i>mes sage</i> " as an error message, and returning to the S-PLUS prompt.
while (cond) expr	Evaluates <i>cond</i> ; if true, evaluates <i>expr</i> , then goes back to the top of the loop, evaluating <i>cond</i> again.
repeat expr	A simpler version of while, repeat performs no tests. It simply repeats <i>expr</i> indefinitely. Because they have no natural termination, repeat loops should have some breaks built in.
for (name in expr1) expr2	Evaluates <i>expr2</i> once for each <i>name</i> in <i>expr1</i> . Although for loops are widely used in most programming languages, they are generally less efficient in S-PLUS than calculations done by means of vectorized arithmetic.

**V. Graphics**

S-PLUS has a wide variety of editable graphics plot types. Users can easily generate the graph by using the point-and-click interface.

### ***2D Line and Scatter Plots***

Scatter - Line - Scatter w/Line - Line/Isolated Points - High Density - Line w/Text as Symbols - Bubble - Color Plot - Bubble Color - Loess - Smoothing Spline - Robust - Dot Plot - 2D Time Series - Linear Fit - Polynomial Fit - Exponential Fit - QQ Quantile Plots - QQ Normal Plot w Line - QQ Plot without Line - Box Plot - Pie Chart - Histogram - Density - Histogram/Density - Bar Plots - Grouped - Bar - Stacked Bar - Bar w/Error - Grouped Bar w/Error- Bar Origin Base - Horizontal Bar - Stacked Horizontal Bar-High-Low-Close Plots - Error Bar Plots - Vertical Error Bar - Area Plot - Scatter Plot Matrix - Contour - Filled Contour - Levels Plot

### ***3D Line and Scatter Plots***

Scatter - Line - Line with Scatter - Drop Line Scatter - Regression - Regression with Symbols - Coarse Surface - Data Grid Surface - Spline - Coarse Filled Surface - Data Grid Filled Surface - Filled Spline Surface - 8 Color Draped Surface - 16 Color Draped Surface - 32 Color Draped Surface - Bar - 3D Contour Plot - Filled Contour

## **VI. S+SpatialStats**

S+SpatialStats is an added module to S-PLUS. It is the first comprehensive, object-oriented software package for the analysis of spatial data. provides tools specifically developed for the exploratory data analysis and modeling of three broad classes of spatial data: geostatistical data, point patterns and lattice data:

1. Geostatistical data - spatial data observed at fixed locations, such as shellfish abundance along a coastline, pollution readings at monitoring stations, and weather observations at weather stations;
2. Point patterns - spatial data consisting of a finite number of locations in an observed spatial region, such as locations of earthquake epicenters, and outbreak of disease sites; and
3. Lattice data - data with observations associated with a spatial region, where the regions can be regularly or irregularly spaced, such as cancer rates in various states (irregular) or political poll data by county (irregular) or precinct, or image data for radiation measures on a pixel grid (regular).

The current S+SpatialStats module includes the following functions:

Geostatistical Data: Contour plots, 3-D point clouds, Variogram plots and boxplots, Directional variograms and correlograms for exploring anisotropy, Empirical variogram estimation including robust methods, Variogram models including spherical and exponential, Ordinary and universal kriging, Kriging prediction at arbitrary locations with standard errors, Parametric and nonparametric trend surfaces.

- Estimating Variogram
- Fitting Theoretical Variogram Models
- Performing Ordinary and Universal Kriging
- Simulating Geostatistical Data

Point Patterns: Point maps that include region boundaries, Spatial randomness tests, Ripley's K-functions, Simulation of spatial random processes, Local intensity estimation.

- Examine point pattern data for complete spatial randomness
- Estimate the intensity of a spatial point pattern
- Calculate Ripley's K-functions
- Simulate a spatial point process

Lattice Data: "Binning" of high density data into a regular lattice of counts, Geary and Moran spatial autocorrelation coefficients, Spatial regression models including conditional and simultaneous autoregressive models, Nearest neighbor search.

- Define spatial neighbors
- Test lattice data for spatial autocorrelation
- Test for local spatial associations
- Model lattice data using spatial regression
- Simulate lattice data

To load SpatialStats into S-PLUS:

```
> module(spatial)
```

To make the S+SpatialStats automatically loaded when S-PLUS is started:

```
> .First <- function(){module(spatial)}
```

## **VII. Some Basic Commands**

as.data.frame() - Convert a S-PLUS object into a data frame.

```
> x <- as.data.frame(x)
```

as.double() - returns an object as a double.

```
> x <- as.double(x)
```

as.integer() - returns an object as an integer.

```
> x <- as.integer(x)
```

as.numeric() - returns an object as a numeric object.

```
> x <- as.numeric(x)
```

attach() - Adds a directory or other object to the S-PLUS search list or returns the current search list.

```
> attach("d:\\todd\\_Data", pos = 1)
```

data.class() - returns the class of the object.

```
> data.class(x)
```

detach() - Removes a database from the search list.

```
> detach("d:\\todd\\_Data")
```

fix() - Invokes an editor to create or edit a S-PLUS function.

```
> fix(my.function)
```

ls() - List datasets in current working directory.

rm() - Removes object (x) from the working directory.

```
> rm(x,y,z)
```

search() - Returns the names of the search list.

**REFERENCES:**

MathSoft, 1997, S-PLUS 4 User's Guide, Data Analysis Products Division, MathSoft, Inc., Seattle, WA.

MathSoft, 1997, S-PLUS 4 Programmer's Guide, Data Analysis Products Division, MathSoft, Inc., Seattle, WA.

MathSoft, 1997, S-PLUS 4 Guide to Statistics, Data Analysis Products Division, MathSoft, Inc., Seattle, WA.

MathSoft, 1996, S+SpatialStats User's Manual for Window and Unix, Data Analysis Products Division, MathSoft, Inc., Seattle, WA.

MathSoft's Web Site: <http://www.mathsoft.com>