Statistical Computing

Webpage:

1. **Course notes:**

   http://web.thu.edu.tw/wenwei/www/cgi-bin/

then click on Course Notes and then click on

Statistical Computing (word , pdf).

2. **Online test and grades:**

   http://web.thu.edu.tw/wenwei/www

3. **Software: R:** http://www.r-project.org/

Chapter 1: The Splus (R) Language and Simple Programming

This course mainly uses the S-plus as a tool to convey some basic concepts in statistical computing. S-plus offers:

1. an extensive and coherent collection of tools for statistics and data analysis.
2. a language for expressing statistical models and tools for using linear and non-linear statistical models.
3. graphical facilities for data analysis and display either at a workstation or PC as well as hardcopy.
4. an effective objective-oriented programming language which can easily be extended by the user community.

Note:

The S system wins ACM software system award. The other software win this award including Unix, Tex, Postscript, TCP/IP, WWW, Apache,…. For details, see:

http://awards.acm.org/software%5Fsystem/

The ACM citation states that *"for the S system, which has forever altered how people analyze, visualize and manipulate data."*

1.1. The S language: Basics

1. Naming conventions:

(a) The upper case, lower case letters (A, B, …, Z, a, b, …, z), the digits $0-9$ in any non-initial position and also the period "." can be used.

Example (Splus):

```
A1= 1
a1= 5
A1
a1
b.x=3
b.x
9x=4         # digit  9  can not be put in the initial position
Error …..
```

Note:

1. # symbol marks the rest of the lines as comments
2. S-plus is case sensitive. That is, Var1 and var1 are distinct S name.

(b) "…", break, for, function, if, in, next, repeat, return, while, are reserved Identifiers. These reserved identifiers can not be used as a variable name.

(c) Avoid using system names; in particular c, q, s, t, C, D, F, I, T, diff, mean, pi, range, rank, tree, and var.

2. Data structure:

(a) Vector

(b) Matrix

(c) List (output)

(d) Data frame (input)

(a) Vector:

Example (Splus):

```
v1= c(1.1,2.2,3.3,4.4,5.5,6.6)
v2=c("a","b","c","d","e","f")       # declare v1 and v2 to be vectors
v1
v2
```

```
v1[3]                          # the third element of v1
v2[2]                          # the second element of v2
1:5
3:7
v1[2:4]                        # the elements of v1 from the second one to the
                               # fourth one
v2[3:5]                        # the elements of v2 from the third one to the
                               # fifth one
v1[c(1,3,5)]                   # the first, the third, and the fifth elements of v1
v2[c(2,3,6)]                   # the second, the third, and the sixth elements of v2
letters
letters[1:6]
names(v1)=c("dog","pig","cat","monkey","cow","sheep")
                               # give a name to each element of v1
v1
v1["dog"]
v1[c("pig","cow")]
v1[-c(1,3,4)]                  # the elements of v3 excluding the first, the third,
                               # and the fourth elements
v3=1:10
v3
v4=c(v1,v3)                    # v4 is the combination of v1 and v3
v4
length(v4)                     # the number of elements in v4
```

(b) Matrices:

There are several ways to construct a matrix.

1. **We construct a vector first, then convert this vector into a matrix.**

Example (Splus):

```
v3
dim(v3)=c(2,5)                 # specify v3 as a 2 by 5 matrix
v3
dim(v3)=NULL                   # the dimension of v3 is 0
v3
v5=matrix(v3,2,5,byrow=T)      # the matrix v5 is filled by row
v5
```

2. Specify the matrix directly.

The following example is to construct the matrix $\begin{bmatrix} 1 & 2 \\ 7 & 9 \\ 4 & 8 \end{bmatrix}$.

Example (Splus):

```
m1=matrix(0,3,2)
m1
m1[1,]=c(1,2)
m1[2,]=c(7,9)
m1[3,]=c(4,8)
m1
m1[1,2]
m1[,1]
m2=diag(c(1,3,4,8))
m2
ncol(m1)                    # number of columns of m1
nrow(m1)                    # number of rows of m1
```

(c) Lists:

A list is used to collect together items of different types.

Example (Splus):

```
v1
v2
m1
r1=list(mynumber=v1,myletter=v2,mymatrix=m1)
r1
r1$mynumber
r1$myletter
r1$mymatrix
r1$mynumber[3]
r1$mymatrix[3,2]
```

(d) Data Frame:

Data frame is a list of variable of the same length, but possibly of different types.

Example (Splus):

```
c1=c("car1","car2","car3")
c2=c("red","blue","white")
c3=c(80,75,60)
d1=data.frame(c1,c2,c3)
d1
```

Note:

Import Data Into Splus:

There are several ways to import data into Splus.

1. File → Import Data → From File
2. For quantitative data, use command "matrix".

Example (Splus):

```
mymatrix=matrix(scan("C://mymatrix.txt"),ncol=2,byrow=T)
mymatrix
```

3. Basic operations:

(a) Arithmetic operations

(b) Mathematical operations

(c) Logical operations

(a) Arithmetic operations:

Arithmetic operations include, + (addition), - (subtraction), * (multiplication), / (division), and ^ (power operator).

Example (Splus):

```
(1.5+2-3)*4
(4/2)
4^(1/2)
sqrt(4)
v1=1:3
v2=4:6
v1+v2                    # the usual vector addition
v1-v2                    # the usual vector subtraction
```

```
v1*v2                         # each element of v1 multiplied by the
                                # corresponding element of v2
v1/v2                         # each element of v1 divided by the
                                # corresponding element of v2
v1^2                            # square each element of v1
v1^v2
m1=matrix(1:4,2,2)
m1
m2=diag(1:2)
m2
m1+m2                           # the usual matrix addition
m1-m2                          # the usual matrix subtraction
m1*m2                           # each element of m1 multiplied by the
                                 # corresponding element of m2
m1/m2                          # each element of m1 divided by the
                                 # corresponding element of m2
m1^2                           # square each element of m1
m1^m2
v3=rep(0,9)
v3
v1+v3
```

(b) Mathematical operations:

1. **Some numerical functions, including abs, sign, log, log10, exp, sin, cos, tan, asin, atan, cosh, sinh, tanh.**

Example (Splus):

abs(-1.3)
sign(-2)
sign(4)
logb(9,3) # $log_3(9)$
log(9) # $ln(9)$
log10(100) # $log_{10}(100)$
exp(2) # e^2
exp(v1)
exp(m1)
sin(v1)
cos(m2)

```
pi
tan(pi/4)                # $tan\left(\frac{\pi}{4}\right)$
asin(1)                  # $sin^{-1}(1)$
acos(1)                  # $cos^{-1}(1)$
atan(1)                  # $tan^{-1}(1)$
```

2. Element matrix operations:

Example (Splus):

```
m1*m2
m1 %*% m2                # the usual matrix multiplication
t(m1)                    # the transpose of m1
m3=solve(m1)             # m3 is the inverse matrix of m1
m1 %*% m3
v7=c(2,3)
s=solve(m1,v7)           # solve the linear system $\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix}x=\begin{bmatrix}2\\ 3\end{bmatrix}$
s                        # s is the solution
e1=eigen(m1)             # the information of eigenvectors and
                         # eigenvalues of m1 is saved in e1
e1$vectors               # the eigenvectors of m1
e1$values                # the eigenvalues of m1
e1$vectors[,1]           # the first eigenvector of m1
e1$values[1]             # the first eigenvalue of m1
prod(e1$values)          # the determinant of m1
det(m1)                  # the determinant of m1
sum(e1$values)           # the trace of m1
m4=rbind(m1,m2)          # combine m1 and m2 by row
m4
m5=cbind(m1,m2)          # combine m1 and m2 by column
m5
```

(c) Logical operations:

Comparison and logical operators: ==, >, <, >=, <=, !, !=, &&, | |, &, !.

Example (Splus):

```
1 == 2
1 > -1
v1= 1:3
v2=5:3
v1>=v2
m1<m2
m1<=m2
(! 1 > 2)                # !: not
(! T)
(1 > 2) && (2 > 3)       # &&: and
(1 > 2) && (2 < 3)       # (condition  1) && (condition  2)=T as both
(1 < 2) && (2 < 3)       # conditions are true!!
(1 > 2) || (2 > 3)       # ||: or
(1 > 2) || (2 < 3)       # (condition  1) || (condition  2)=T as one of
(1 < 2) || (2 < 3)       # the two conditions is true
v3=1:10
v3>5
v3[v3>5]
(v3>4) && (v3<8)
(v3>4) & (v3<8)          # &: vectorized "and"
v3[(v3>4) & (v3<8)]
(v3>7) | (v3<3)          # |: vectorized "or"
v3[(v3>7) | (v3<3)]
```

4. Graphics:

(a) Simple graphical function

(b) Exploratory data analysis

(c) Higher dimension plot

(a) Simple graphical function:

The most commonly used graphical function: plot

Example (Splus):

```
corn.rain                               # rainfall measurement from 1890 to 1927
corn.yield                              # yearly corn yield from 1890 to 1927
plot(corn.rain,corn.yield)
plot(corn.rain,corn.yield,xlab=”rain (1890~1927)”,ylab=”yield (1890~1927)”)
title(“Rain versus Yield”)
rain1=corn.rain[1:19]
yield1=corn.yield[1:19]
rain2=corn.rain[20:38]
yield2=corn.yield[20:38]
plot(corn.rain,corn.yield,type=”n”)          # no plotting
points(rain1,yield1,pch=0)                   # the data from 1890 to 1908 are
                                             # imposed on the original figure
                                             # pch = 0~18
points(rain2,yield2,pch=1)                   # the data from 1909 to 1927 are
                                             # imposed on the original figure
points(rain2,yield2,pch=”C”)
rain1[1]
yield1[1]
text(9.6,25.5,”1890”)                        # add string “1890” to the figure
text(9.6,24.5,”1890”)                        # (9.6,25.5) or (9.6,24.5) is the
                                             # location of the inserted string
x=1:10
y1=3*x
y2=2*x+1

plot(x,y2, ylim=c(min(y1,y2),max(y1,y2)),type=”n”)
points(x,y1,pch=2)
points(x,y2,pch=3)
lines(x,y1,lty=1)
lines(x,y2,lty=2)                            #  lty = 1~8
legend(8,8,c(“y=3x”,”y=2x+1”),lty=1:2)       # add the legend to the figure
                                             # (8,8) is the location of the legend
```

Note:

The points and lines generated by the commands “points” and “lines” can be imposed on the figure generated by the command “plot”. Similarly, the text or string and the legend generated by the commands “text” and “legend” can also be imposed on the original figure.

```
plot(corn.rain,corn.yield,axes=F)              # an enclosing box, tick marks and
                                                # axis labels will be plotted.
axis(1,at=c(8,11,14),labels=c("little","med.","lots"))
axis(2,at=32,labels="median")
box()
par(mfrow=c(2,2))
plot(rain1,yield1)
plot(rain2,yield2)
plot(x,y1)
plot(x,y2)
graphics.off()
```

(b) Exploratory data analysis:

Histogram, box plot, stem-leaf plot, Q-Q plot (quantile-quantile plot)

Example (Splus):

```
corn.rain
hist(corn.rain)                         # the histogram
hist(corn.rain,nclass=5)                # histogram with specific number of
intervals
b=seq(6,18,by=0.5)
hist(corn.rain,breaks=b)                # histogram with specific breaks
hist(corn.rain,breaks=c(6,9,12,15,18))
boxplot(rain1,rain2)                    # Box plot
score=c(19,21,25,27,68,69,71,78,81,92)
stem(score)                             # stem-leaf plot
x=rnorm(100)
x
qqnorm(x)
qqline(x)
```

(c) Higher dimension plot:

3 D plots: contour, perspective, and image plots

Example (Splus):

```
switzerland
contour(switzerland)
voice.five
persp(voice.five)
persp(voice.five,eye=c(72000,350,30))
```