Computing in Epidemiology and Biostatistics

Input, output, and plotting with R

Wan-Yu Lin

Input from a file

```
setwd("D:/Comp/1")
list.files()      # list the files in the working directory
Seizure1 <- read.csv('seizure.csv')
Seizure2 <- read.table('Seizure.txt')

install.packages("data.table")      # Fast aggregation of large data
library(data.table)
Seizure1 <- fread('seizure.csv')
Seizure2 <- fread('Seizure.txt')

# For a larger data set
starttime <- Sys.time()
Seizure1 <- read.table('D:/Comp/3/example.txt')
Sys.time()-starttime

# fread is similar to read.table but faster and more convenient
starttime <- Sys.time()
Seizure1 <- fread('D:/Comp/3/example.txt')
Sys.time()-starttime
```

Furthermore, we can use scan to read a vector of values from a file.

```
scan(file = "Seizure.txt", what = double(), n = -1, sep = "", skip = 0, quiet = FALSE)
```

scan returns a vector.

file gives the file to read from.

what gives an example of the mode of data to be read, with a default of double() for numeric data.

n gives the number of elements to read. If n = -1 then scan keeps reading until the end of the file.

sep allows you to specify the character that is used to separate values, such as ",". The default "" has the special meaning of allowing any amount of white space (including tabs) to separate values. Note that a newline/return always separates values.

skip is the number of lines to skip before you start reading, default of 0. This is useful if your file includes some lines of description before the data starts.

quiet controls whether or not scan reports how many values it has read, default FALSE.

Ex14: Please use "scan" to read all elements in "seizure.csv".

Output texts

```
x <- "Citroen SM"
y <- "Jaguar XK150"
z <- "Ford Falcon GT-HO"
(wish.list <- paste(x, y, z, sep = ", "))
```

```
x <- 7
n <- 5
# display powers
cat("Powers of", x, "\n")              # cat displays concatenated character
cat("exponent result\n\n")             # \n for a new line
result <- 1
for (i in 1:n) {
   result <- result * x
   cat(format(i, width = 8),format(result, width = 10),"\n", sep = "")
}
```

```
source("D:/Comp/4/powers.R")          # call an R script file
```

Functions **format** and **paste** also take vector input. Thus the program above could be vectorised as follows:

```
cat(paste(format(1:n, width = 8), format(x^(1:n), width = 10), "\n"), sep = "")
```

What if you remove "paste" from the above code?

```
cat(format(1:n, width = 8), format(x^(1:n), width = 10), "\n", sep = "")
```

Output to a file

```
write.table(Seizure1, 'Test1.txt', sep=' ', row.names=FALSE, col.names=FALSE, quote=FALSE, na='x', append=FALSE)
write.table(Seizure1, 'Test1.csv', sep=',', row.names=FALSE, col.names=FALSE, quote=FALSE, na='x', append=FALSE)
```

Plotting in R

```
bmi <- read.csv('D:/Comp/4/BMIrepeated.csv')
x <- seq(0,9,3)
y <- cbind(bmi$BMI0, bmi$BMI1, bmi$BMI2, bmi$BMI3)

par(mfrow = c(1,2))
plot(x,y[1,],type="b",lwd=1,col=1,lty=1,pch=1,ylim=c(15,50),axes =
F,xlab="months",ylab="BMI",main="Placebo group")
axis(1, at = x, labels = seq(0,9,3))
axis(2)

for(subj in 2:10){
   lines(x,y[subj,],lty=1,lwd=1,col=subj,type="b",pch=subj)
}
legend("topright",bty="n",
c("ID1","ID2","ID3","ID4","ID5","ID6","ID7","ID8","ID9","ID10"),lty=1,col=(1:10),lwd=1,pch=(1:10))
```

Ex 15: Please plot the BMI curves for ID51-ID60, to the right of ID1-ID10. Please put a title as “Drug group” and make a legend.

Box plot

```
par(mfrow = c(1,1))
boxplot(bmi$BMI0,bmi$BMI1,bmi$BMI2,bmi$BMI3,col="orange")
```

Histogram

```
hist(bmi$BMI0,col="blue",main="BMI at baseline",xlab="BMI")
```

Plot for multiple columns

```
x <- seq(0.5,2,0.01)
y <- cbind(exp(x),log(x))
matplot(x,y,col=(3:4),pch=(1:2),lty=(1:2),type="b",frame=F)     #matrix plot
```

Pie chart

```
# Simple Pie Chart
subject <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pie(subject, labels = lbls, main="Pie Chart of Countries")
```

Pie chart with percentages

```
subject <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pct <- round(subject/sum(subject)*100)
lbls <- paste(lbls, pct) # add percentages to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(subject,labels = lbls, col=rainbow(length(lbls)), main="Pie Chart of Countries")
```

3D pie chart

```
install.packages("plotrix")
library(plotrix)
subject <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pie3D(subject,labels=lbls,explode=0.1,main="Pie Chart of Countries ")
```

Ex S2: Please add percentages to the above 3D pie chart.

Hwk 4 (totally 10 points):

Ex S1: Please calculate “factorial(10)” by using a “for” loop and a “while” loop, respectively. (4 points)

Hint:

```
> factorial(10)
[1] 3628800
> 1*2*3*4*5*6*7*8*9*10
[1] 3628800
```

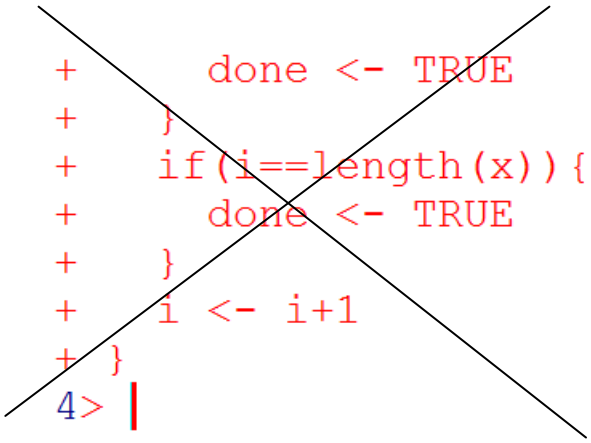
Revisit Ex 12: x <- c(3600, 5000, 12000, NA, 1000, 2000, 600, 7500, 1800, 9000), please end the loop once you meet the missing value, and please tell me which observation is the missing value. (2 points)

Requirements:

1. Please use “while” to build a “do…until…loop”.
2. Please use “cat” and the output should be

```
+      done <- TRUE
+    }
+    if(i==length(x)){
+      done <- TRUE
+    }
+    i <- i+1
+ }
4
>
```

=> the cursor should be put to the new line.



=> Not acceptable.

Ex 15: Please plot the BMI curves for ID51-ID60, to the right of ID1-ID10. Please put a title as “Drug group” and make a legend. (please see page 3 of this handout) (2 points)

Ex S2: Please add percentages to the 3D pie chart on page 4 of this handout. (2 points)