Problem Set 2

Stephanie Kim

September 22, 2014

Problem 1

Download the zipped csv file from the URL provided in the problem set and save it at the directory.

```
url="http://www.stat.berkeley.edu/share/paciorek/AirlineData2006-2008.csv.bz2"
download.file(url, destfile="/Users/stephaniekim/stat243-fall-2014/AirlineData.bz2")
```

Without unzipping, we open a connection that can read the file AirlineData.bz2 to AirlineData.

```
AirlineData <- bzfile("AirlineData.bz2", open="r")
```

Since the file is too huge, we are going to read the file block by block.
To set the size of the blocks, we first measure how many rows there are in this file.

```
## We randomly assign the size of block we are reading as 20000 lines.
## So we are reading 20000 lines at a time.

randomblocks <- 20000

## Then we set the initial value of the variable lines as 0.

lines <- 0

## Add the number of lines to the initial value until there is no remaining lines to count.
##Once we count 20000, we add 20000 to the previous value of lines.

(while((linesadd <- length(readLines(AirlineData,randomblocks))) > 0 )    lines <- lines+linesadd)

## Then we should subtract one line since we are not counting the header.

lines=lines-1

## After this we have to close the connection.

close(AirlineData)
```

Now we choose the number of blocks we are going to use.

```
## Print the value of lines which is the number of rows in our file; 21604867.

lines

## Now we know how many rows we have. I decided to set my block as 30000.

blocks=30000
```

```
## Make a variable itrNums=the ceiling of lines/blocks.

itrNums <- ceiling(lines/blocks)
```

Then we open the connections.

```
## We open the connection that can read the file AirlineData.bz2 to AirlineData2 again.

AirlineData2 <- bzfile("AirlineData.bz2", open="r")

## This time, we open the connection that can write data to the 3 files.
## We will save the output of the for loop later.
## We are making it 3 so that we can save data for each 3 years separately.

Air2006 <- bzfile("SFO2006.csv.bz2", open="w")
Air2007 <- bzfile("SFO2007.csv.bz2", open="w")
Air2008 <- bzfile("SFO2008.csv.bz2", open="w")
```

Let's run a for loop.

```
## Now we run a for loop for i=1,2,...,itrNums(# of rows divided by # of blocks.

for (i in 1:itrNums) {

## We read the AirlineData.bz2 through the connection AirlineData2 and save it to the data frame Airlin

  Airline<-read.table(AirlineData2, header = TRUE, sep = ",", nrows = blocks)

## Subset the data frame Airline by finding SFO in 18th column and save it another data frame SFOAirlin

 SFOAirline<-Airline[which(as.character(Airline[,18])=="SFO"),]

## Stratify the data frame SFOAirline by categorizing whether the 1st column is 2006,2007, or 2008 and

SFO2006<-SFOAirline[which(SFOAirline[,1]=="2006"),]

SFO2007<-SFOAirline[which(SFOAirline[,1]=="2007"),]

SFO2008<-SFOAirline[which(SFOAirline[,1]=="2008"),]

## Now we write a table based on the data frames.

write.table(SFO2006, file=Air2006, append=TRUE, quote=FALSE, sep=",", row.names=FALSE, col.names=FALSE)

write.table(SFO2007, file=Air2007, append=TRUE, quote=FALSE, sep=",", row.names=FALSE, col.names=FALSE)

write.table(SFO2008, file=Air2008, append=TRUE, quote=FALSE, sep=",", row.names=FALSE, col.names=FALSE)

 }
```

Lastly, we need to close the connection.

```
close(Air2006)

close(Air2007)

close(Air2008)
```

Problem 2

(a) MyFuns is a vector of 3 function outcomes. Since j is not in the for loop of MyFuns function, it does not go through the loop but just return a vector of 3 function outcomes. And here we can observe that in the for loop, the outcome is 1 when i=1, and it is updated to 2 when i=2, and it is updated again to 3 when i=3. So the final outcome is 3. Thus, the result of the first evaluation is the vector of 3 function outcomes of which the each entry is the final outcome, 3.
(b) In this case, since i is in the for loop, it does go through the loop. So the result of the second evaluation is the vector of 3 functions where the first entry is return(i=1)=1, the second entry is return(i=2)=2, and the third entry is return(i=3)=3. The value of 'i' is being found from global environment.
(c) myFuns is a vector of length 3. Since our final function is processed after for loop of f[[i]] function, all 3 entries of myFuns vector are the final result of the for loop (just like in 2(a)). Thus, all entries are i=3. Therefore, regardless of whether we are using i or j as an iteration variable, the result is a vector of length 3 with each entry=3. The value of 'i' is being found from local environment.

Problem 3

The frame number 0 is the global environment. All variables, functions and objects exist.
The frame number 1 is the environment defined by sapply. Only function (x) and variables 0 to 3 exist.
The frame number 2 is the environment defined by function function(x). Only function ls(x) and the variable x exist.
The frame number 3 is the environment defined by ls() function. Only variable x exist.