Assignment 2

Ya-Chen Lin

2015 09 12

Question 01:

Working with data in the datasets folder on the course GitHub repo, you will find a file called cancer.csv, a dataset in comma-separated values (csv) format. This is a large cancer incidence dataset that summarizes the incidence of different cancers for various subgroups.

```
#i:Load the data set into R and make it a data frame called cancer.df
cancer.df <- data.frame(read.csv('cancer.csv', fill=TRUE, header = TRUE))
#ii:Determine the number of rows and columns in the data frame
nrow(cancer.df)
```

```
## [1] 42120
```

```
ncol(cancer.df)
```

```
## [1] 8
```

```
#iii:Extract the names of the columns in cancer.df
colnames(cancer.df)
```

```
## [1] "year"       "site"       "state"      "sex"        "race"
## [6] "mortality"  "incidence"  "population"
```

```
#iv:Report the value of the 3000th row in column 6
cancer.df[3000,6]
```

```
## [1] 350.69
```

```
#v:Report the contents of the 172nd row.
cancer.df[172,]
```

```
##     year                          site  state  sex  race mortality
## 172 1999 Brain and Other Nervous System nevada Male Black         0
##     incidence population
## 172         0      73172
```

```
#vi:Create a new column that is the incidence rate (per 100,000) for each row.
cancer.df$incidence_rate <- ( cancer.df$incidence )/100000
colnames(cancer.df)
```

```
## [1] "year"           "site"           "state"          "sex"
## [5] "race"           "mortality"      "incidence"      "population"
## [9] "incidence_rate"
```

```r
#vii:How many subgroups (rows) have a zero incidence rate?
sum(cancer.df$incidence_rate == 0)
```

```
## [1] 23191
```

```r
#viii:Find the subgroup with the highest incidence rate
cancer.df[which.max(cancer.df$incidence_rate),]
```

```
##       year   site      state    sex  race mortality incidence population
## 21387 2002 Breast california Female White   3463.74     18774   13690681
##       incidence_rate
## 21387        0.18774
```

Question 02:

i.Create the following vector: x <- c("5","12","7"). Which of the following commands will produce an error message? For each command, Either explain why they should be errors, or explain the non-erroneous result.

```r
x <- c("5","12","7")
class(x)
```

```
## [1] "character"
```

x consists of three "character" arguments

```r
max(x)
```

```
## [1] "7"
```

Within maxima command, it can return numeric or character arguments. Thus we can get value "7" as character.

```r
sort(x)
```

```
## [1] "12" "5"  "7"
```

Sort can return objects with a numeric, complex, character or logical vectors. Thus we can get the order of x.

```r
#sum(x) will give an error which couldn't be shown in PDF.
```

Sum can only return numeric or complex or logical vectors. Thus in this case where x consists only character value, we will receive an error message.

ii:For the next two commands, either explain their results, or why they should produce errors.

```r
y <- c("5",7,12)
#y[2] + y[3] will give an error message which couldn't be shown in PDF
```

y is cassified as character variable. + is only used in numerics not character. Thus by adding them together, we will recive the error message:"non-numeric" argument.

iii:For the next two commands, either explain their results, or why they should produce errors.

```r
z <- data.frame(z1="5",z2=7,z3=12)
z[1,2] + z[1,3]
```

```
## [1] 19
```

```r
class(z[1,2])
```

```
## [1] "numeric"
```

```r
class(z[1,3])
```

```
## [1] "numeric"
```

When characteric arguments are passed to data frame, they are numeric arguments, as in the class commands. Thus, we can use + operator to add two numeric values together.

Question 03:

Data Structure: Give R expressions that return the following matrices and vectors (i.e. do not construct them manually).

```r
#i.
c(x <- 1:8, rev(x[-length(x)]))
```

```
##  [1] 1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
```

```r
#ii
rep(1:5, c(1:5))
```

```
##  [1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
```

```r
#iii
x<- c(0,1,1,1,0,1,1,1,0)
matrix(x,nrow=3, ncol=3)
```

```
##      [,1] [,2] [,3]
## [1,]    0    1    1
## [2,]    1    0    1
## [3,]    1    1    0
```

```r
#iv:
matrix(c(1^(1:4), 2^(1:4), 3^(1:4), 4^(1:4)), nrow =4, ncol=4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    1    4    9   16
## [3,]    1    8   27   64
## [4,]    1   16   81  256
```

Question 04:

i:Let $h(x,n) = 1 + x + x^2 + \ldots + x^n = \sum_{i=0}^{n} x^i$. *Write an R program to calculate* $h(x,n)$ *using a for loop.*

```r
#A<- NULL
#B <- rep(x,(n+1))
#for(i in 0:n){
  #A[(i+1)] <-  B[(i+1)]^i
#}
#sum(A)
```

iia:Find the sum of all the multiples of 3 or 5 below 1,000.

```r
x <- c(1:999)
sum(which(x%%3 == 0 | x%%5 == 0))
```

```
## [1] 233168
```

iib:Find the sum of all the multiples of 4 or 7 below 1,000,000

```r
x <- c(1:999999)
sum(as.numeric(which(x%%4 == 0 | x%%7 == 0)))
```

```
## [1] 178571071431
```

iii:Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be $(1, 2, 3, 5, 8, 13, 21, 34, 55, 89)$. *Write an R program to calculate the sum of the first 15 even-valued terms.*

```r
B <- c(1,2,rep(0,50))
for (i in 3:52){
B[i] <- B[i-1] + B[i-2]
}
evenB <- B[which(B %% 2 == 0)]
sum(evenB[1:15])
```

```
## [1] 1485607536
```