Bios 6301: Assignment 2

Yudong Cao

Grade: 53/50

(informally) Due Tuesday, 20 September, 1:00 PM

50 points total.

This assignment won't be submitted until we've covered Rmarkdown. Create R chunks for each question and insert your R code appropriately. Check your output by using the `Knit PDF` button in RStudio.

1. **Working with data** In the `datasets` folder on the course GitHub repo, you will find a file called `cancer.csv`, which is 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. (18 points)

   1. Load the data set into R and make it a data frame called `cancer.df`. (2 points)

```r
#setwd("~/Desktop/FA 16/BIOS 6301/hw")
cancer.df<-read.csv("cancer.csv")
cancer.df<-data.frame(cancer.df)
attach(cancer.df)
```

   2. Determine the number of rows and columns in the data frame. (2)

```r
nrow(cancer.df)
```

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

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

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

   3. Extract the names of the columns in `cancer.df`. (2)

```r
names(cancer.df)
```

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

   4. Report the value of the 3000th row in column 6. (2)

```r
cancer.df[3000,6]
```

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

   5. Report the contents of the 172nd row. (2)

```r
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
```

   6. Create a new column that is the incidence *rate* (per 100,000) for each row.(3)

```r
cancer.df[,'incrate']<-incidence/100000
```

JC Grading - 1 For incidence rate above should be incidence / population * 100000

~~~
7. How many subgroups (rows) have a zero incidence rate? (2)

```r
sum(cancer.df['incrate']==0)
```

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

```r
#23191 rows have zero incrate
```

8. Find the subgroup with the highest incidence rate.(3)

```r
which(cancer.df['incrate']==max(cancer.df['incrate']))
```

```
## [1] 21387
```

```r
#the 21387th subgroup has highest incrate
```
~~~

JC Grading - 1 syntax is fine but answer is incorrect b/c of how incidence rate was calculated

2. **Data types** (10 points)

    1. 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. (4 points)

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

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

```r
# no error; for characters, the first 'number' is compared to get max(x)=7
sort(x)
```

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

```r
# no error; for characters, the first 'number' is primarily sorted "12" "5" "7"

#sum(x)
# error; cannot sum up elements of character strings
```

    2. For the next two commands, either explain their results, or why they should produce errors. (3 points)

```r
y <- c("5",7,12)
# define y to be a vector with the character string "5", "7" and "12"

#y[2] + y[3]
```

```
# error because R cannot add two elements of character strings
```

3. For the next two commands, either explain their results, or why they should produce errors. (3 points)

```
z <- data.frame(z1="5",z2=7,z3=12)
# define z to be a data frame with a row of numeric elements 5, 7 and 12
z[1,2] + z[1,3]
```

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

```
# add up the 2nd and 3rd element of z, 7+12=19
```

3. **Data structures** Give R expressions that return the following matrices and vectors (*i.e.* do not construct them manually). (3 points each, 12 total)

   1. $(1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1)$

```
c(1:8,7:1)
```

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

   2. $(1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5)$

```
rep(1:5,time=1:5)
```

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

   3. $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$

```
1-diag(3)
```

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

   4. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 8 & 27 & 64 \\ 1 & 16 & 81 & 256 \\ 1 & 32 & 243 & 1024 \end{pmatrix}$

```
y=c(1:4)
x=matrix(c(y,y^2,y^3,y^4,y^5),nrow=5,byrow=T)
x
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    1    4    9   16
## [3,]    1    8   27   64
## [4,]    1   16   81  256
## [5,]    1   32  243 1024
```

4. **Basic programming** (10 points)

   1. 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. (5 points)

```r
h<-function(x,n) {
  h<-1
  for (i in 1:n) {
    h<-h+x^i
  }
  return(h)
}
```

2. If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The s

    i. Find the sum of all the multiples of 3 or 5 below 1,000. (3, [euler1])

```r
s<-0
for (i in 1:999) {
  if (i%%3==0 | i%%5==0) {
    s<-s+i
  }
}
s
```

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

    2. Find the sum of all the multiples of 4 or 7 below 1,000,000. (2)

```r
t<-0
for (j in 1:999999) {
  if (j%%4==0 | j%%7==0) {
    t<-t+j
  }
}
t
```

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

3. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting wi

```r
x<-c(1,2)
for (i in 1:100) {
  x[i+2]<-x[i]+x[i+1]
}
y<-0
for (j in 1:45) {
  if (x[j]%%2==0) {
    y<-y+x[j]
  }
}
y
```

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

Some problems taken or inspired by projecteuler.