Bios 6301: Assignment 2

Yeji Ko

Due Tuesday, 21 September, 1:00 PM

50 points total.

Add your name as `author` to the file's metadata section.

Submit a single knitr file (named `homework2.rmd`) by email to michael.l.williams@vanderbilt.edu. Place your R code in between the appropriate chunks for each question. Check your output by using the `Knit HTML` 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)
    2. Determine the number of rows and columns in the data frame. (2)
    3. Extract the names of the columns in `cancer.df`. (2)
    4. Report the value of the 3000th row in column 6. (2)
    5. Report the contents of the 172nd row. (2)
    6. Create a new column that is the incidence *rate* (per 100,000) for each row.

    The incidence rate is the `(number of cases)/(population at risk)`, which in this case means `(number of cases)/(population at risk) * 100,000`. (3)

    7. How many subgroups (rows) have a zero incidence rate? (2)
    8. Find the subgroup with the highest incidence rate.(3)

```
# 1
cancer.df <- read.csv('/Users/yejiko/Downloads/cancer.csv')

# 2
nrow(cancer.df)
```

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

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

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

```
# 3
colnames(cancer.df)
```

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

```
# 4
cancer.df[3000,6]
```

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

```
# 5
cancer.df[172,]
```

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

```
# 6

cancer.df[,'rate'] <- (cancer.df[,'incidence'] * 100000)/cancer.df[,'population']

# 7: 23191 rows have a zero incidence rate
nrow(cancer.df[which(cancer.df$rate==0),])
```

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

```
# 8
max(cancer.df$rate)
```

```
## [1] 261.1599
```

```
cancer.df[which(cancer.df$rate == max(cancer.df$rate)),]
```

```
##      year     site                state  sex  race mortality incidence
## 5797 1999 Prostate district of columbia Male Black     88.93       420
##      population     rate
## 5797     160821 261.1599
```

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)

        ```
        max(x)
        sort(x)
        sum(x)
        ```

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

        ```
        y <- c("5",7,12)
        y[2] + y[3]
        ```

    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)
        z[1,2] + z[1,3]
        ```

```
# 1
x <- c("5","12","7")
max(x)
```

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

```
# there is no error but the maximum value is not "7".
# Even if we change the order of values in the vector, it still outputs "7"
# because each string represents some numeric values.
```

```
# However, we cannot understand what is behind the scene intuitively.

sort(x) # there is no error but this output is also incorrect.
```

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

```
# We cannot understand what is behind the scene intuitively.

#sum(x)
# it produces an error message
# because it cannot make a summation of character values

# 2
y <- c("5",7,12)
#y[2] +y[3]
str(y)
```

```
##  chr [1:3] "5" "7" "12"
```

```
# vector always contains the same type of values.
# Because the last two elements have changed to character, this produce errors.

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

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

```
str(z) # dataframe can contain different data types, so this does not produce errors.
```

```
## 'data.frame':    1 obs. of  3 variables:
##  $ z1: chr "5"
##  $ z2: num 7
##  $ z3: num 12
```

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. $(1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1)$
    2. $(1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5)$
    3. $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$
    4. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 8 & 27 & 64 \\ 1 & 16 & 81 & 256 \\ 1 & 32 & 243 & 1024 \end{pmatrix}$

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

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

```
# 2
rep(1:5, times = 1:5)
```

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

```
# 3
mat <- matrix(1,3,3)
diag(mat) <- 0
mat
```

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

```
# 4
mat <- matrix(NA,5,4)
for(i in 1:5){
  mat[i,] <- (seq(1:4)^i)
}
mat
```

```
##      [,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. As an example, use `x = 5` and `n = 2`. (5 points)

```
x <- 5
n <- 2
sum <- 0
for(i in 0:n){
  sum <- sum + x^i
}
sum
```

```
## [1] 31
```

```
1. 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

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

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

```
# 1
sum(c(seq(3,1000, by = 3), seq(5,1000, by = 5)))
```

```
## [1] 267333
```

```
table(seq(3,1000, by = 3)%%3==0) # all multiples of 3
```

```
##
## TRUE
##  333
```

```r
table(seq(5,1000, by = 5)%%5==0) # all multiples of 3
```

```
##
## TRUE
##  200
```

```r
# 2
sum(c(seq(4,1000000, by = 4), seq(7,1000000, by = 7)))
```

```
## [1] 196429428571
```

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

```r
fib <- c(1, 2)
i <- 2
sum <- 0
while (length(fib[fib %% 2 == 0]) < 15) {
  fib <- c(fib, fib[i-1] + fib[i])
  i <- i + 1
  if(fib[i]%%2 ==0){
    sum <- sum + fib[i]
  }
}

length(fib[fib %% 2 == 0])
```

```
## [1] 15
```

```r
sum
```

```
## [1] 1485607534
```

Some problems taken or inspired by projecteuler.