Bios 6301: Assignment 2

Lan Shi

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)

```
cancer.df = read.csv("/Users/lanshi/Desktop/21 FA/6301_Stats_Computing/Bios6301-main/datasets/cancer.csv
```

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

```
cat('There are ',dim(cancer.df)[1],'rows and ',dim(cancer.df)[2], ' columns')
```

```
## There are  42120 rows and  8  columns
```

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

```
colnames(cancer.df)
```

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

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

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

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

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

```
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. 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)

```
cancer.df[,'rate'] = cancer.df$incidence / cancer.df$population * 1e5
head(cancer.df)
```

```
##   year                          site   state    sex     race mortality
## 1 1999 Brain and Other Nervous System alabama Female    Black      0.00
## 2 1999 Brain and Other Nervous System alabama Female Hispanic      0.00
## 3 1999 Brain and Other Nervous System alabama Female    White     83.67
## 4 1999 Brain and Other Nervous System alabama   Male    Black      0.00
## 5 1999 Brain and Other Nervous System alabama   Male Hispanic      0.00
## 6 1999 Brain and Other Nervous System alabama   Male    White    103.66
##   incidence population     rate
## 1        19     623475 3.047436
## 2         0      28101 0.000000
## 3       110    1640665 6.704598
## 4        18     539198 3.338291
## 5         0      37082 0.000000
## 6       145    1570643 9.231888
```

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

```
sum(cancer.df$rate==0)
```

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

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

```
cancer.df[which.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)

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

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

```
sort(x)
```

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

- sum(x) will produce an error message, since the argument of this command should be numeric or complex or logical vectors.
- max(x): result is “7”, since it will compare the first letter of the each element first, if there are equal cases, it will start comparing second letter, etc., so “7” > “5” > “12”.
- sort(x): result is “12”,“5”,“7”, since sort will arrange elements in an increasing order.

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

- The error is because when combining characters and numeric values, numeric values will be coerced to charaters, then plus two characters will give an error.

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

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

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

- The result is 19, (=7+12), since variables combined by data.frame() will retain their own data type, so z[1,2] and z[1,3] are still numeric value 7 and 12, therefore the summation will give the result 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. $(1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1)$

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

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

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

```r
rep(1:5,1:5)
```

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

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

```r
matrix(1,nrow=3,ncol=3) - diag(1,nrow=3)
```

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

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

```r
matrix(rep(1:4,each=5),ncol=4)^(1:5)
```

```
##      [,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)

```r
x = 5
n = 2
h_x_n = 0
for (i in c(0,1:n)){
  h_x_n = h_x_n + x^i
}
h_x_n
```

```
## [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])

```
n = 1:(1000-1)
sum(n[(n%%3==0)| (n%%5==0)])
```

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

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

```
n = 1:(1e6-1)
sum(n[(n%%4==0)| (n%%7==0)])
```

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

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

```
Fib_seq = 1:2 # there is already one even-valued term: "2"
even_terms = 2

while (length(even_terms) < 15){
  Fib_len = length(Fib_seq)
  # find the last two terms
  last_two = Fib_seq[c(Fib_len-1,Fib_len)]
  Fib_seq = c(Fib_seq, sum(last_two))

  # check if the new term is even, if so, add the new term to the even_terms
  last_term = Fib_seq[length(Fib_seq)]
  if (last_term %% 2 == 0){
    even_terms = c(even_terms,last_term)
  }
}
#Fib_seq
#even_terms
sum(even_terms)
```

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

Some problems taken or inspired by projecteuler.