Bios 6301: Assignment 2

Jonathan Lifferth

Due Tuesday, 19 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 marisa.h.blackman@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)

```r
cancer.df <- read.csv('~/R_projects/Bios6301/datasets/cancer.csv')
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
## 1        19     623475
## 2         0      28101
## 3       110    1640665
## 4        18     539198
## 5         0      37082
## 6       145    1570643
```

```
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
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)

```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 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 th

```r
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
## 1        19     623475
## 2         0      28101
## 3       110    1640665
## 4        18     539198
## 5         0      37082
## 6       145    1570643
```

```r
round(.168, 2)
```

```
## [1] 0.17
```

```r
cancer.df['incidence rate'] <- cancer.df[,'incidence'] / cancer.df[,'population']

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 incidence rate
## 1        19     623475   3.047436e-05
## 2         0      28101   0.000000e+00
## 3       110    1640665   6.704598e-05
## 4        18     539198   3.338291e-05
## 5         0      37082   0.000000e+00
## 6       145    1570643   9.231888e-05
```

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

cancer.df[2,‘incidence rate’] == 0

```r
zero_rate <- 0
zero_rate
```

```
## [1] 0
```

```r
for (i in 1:nrow(cancer.df)) {
  if (cancer.df[i,'incidence rate'] == 0) {
    zero_rate <- zero_rate + 1}
}
zero_rate
```

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

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

```r
highest_subgroup <- 0
highest_i <- 0
highest_rate <- 0

for (i in 1:nrow(cancer.df)) {
  if (cancer.df[i,'incidence rate'] > highest_rate) {
    highest_rate <- cancer.df[i,'incidence rate']
    highest_subgroup <- paste(cancer.df[i,'state'], cancer.df[i,'sex'], cancer.df[i,'race'])}

}

highest_rate
```

```
## [1] 0.002611599
```

```
highest_subgroup
```

```
## [1] "district of columbia Male Black"
```

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)

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

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

```
max(x)
```

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

```
?max()
# max(x) returned "7" because "Character versions are sorted lexicographically" by max()

sort(x)
```

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

```
# sort(x) returned "12" "5"  "7" because the first character of "12" is "1" -- the function does not tr

#sum(x)
# sum(x) returned "Error in sum(x) : invalid 'type' (character) of argument" because characters cannot
```

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

y <- c(“5”,7,12)

y[2] + y[3]

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

# I was surprised to see the error "Error in y[2] + y[3] : non-numeric argument to binary operator" bec
# evidently, the type of these values became "character" when they were combined with "5"

typeof(y[3])
```

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

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

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

```
z <- data.frame(z1="5",z2=7,z3=12)
z
```

```
##   z1 z2 z3
## 1  5  7 12
```

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

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

```
# this command did not produce an error because the values were arranged in a data.frame, allowing them
```

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(seq(1:8), seq(from = 7, to = 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, times = 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
x <- matrix(rep(1, 9), nrow=3)
x
```

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

```r
x2 <- matrix(c(0,1,1,1,0,1,1,1,0), nrow=3)
x2
```

```
##      [,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
x <- matrix(c((1:4), ((1:4) ^ 2), (1:4) ^ 3, (1:4) ^ 4, (1:4) ^ 5), nrow=5)
x
```

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

# n starts at 0 and increases with each iteration

sum <- 0
for (i in 0:n) {
  sum <- sum + x^i
#  print(sum)
#  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])

```r
sum <- 0
for (i in 1:1000) {
  if (i%%3 == 0) {
  sum <- sum + i
} else if (i%%5 == 0) {
  sum <- sum + i
}
}

print(sum)
```

```
## [1] 234168
```

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

```r
sum <- 0
for (i in 1:1000000) {
  if (i%%4 == 0) {
  sum <- sum + i
} else if (i%%7 == 0) {
  sum <- sum + i
}
}

print(sum)
```

```
## [1] 178572071431
```

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

```r
term_count <- 1
term_sum <- 2
a <- 1
b <- 2

while (term_count < 15) {
  num <- a + b
  if (num %% 2 == 0) {
    term_sum <- term_sum + num
    term_count <- term_count + 1
    a <- b
    b <- num
  } else {
    a <- b
    b <- num
    }
}

term_sum
```

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

Some problems taken or inspired by projecteuler.