BIOS 6301: Assignment 2

Charles Rhea

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)

```
library(readr)
cancer.df <- read_csv("~/Desktop/BIOS 6301 - Introduction to Statistical Computing/datasets/cancer.csv")
```

```
## Rows: 42120 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): site, state, sex, race
## dbl (4): year, mortality, incidence, population
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(cancer.df)
```

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

There are 42,120 rows and 8 columns

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

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

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

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

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

There names of the columns are: year, site, state, sex, race, mortality, incidence, and population

```
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)

The value of the 3000th row, 6th column - 350.69

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

```
## # A tibble: 1 x 1
##   mortality
##       <dbl>
## 1      351.
```

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

See output of row 172 below

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

```
## # A tibble: 1 x 8
##    year site                    state sex   race  mortality incidence population
##   <dbl> <chr>                   <chr> <chr> <chr>     <dbl>     <dbl>      <dbl>
## 1  1999 Brain and Other Nervou~ neva~ Male  Black         0         0      73172
```

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

See output below

```
cancer.df$incidence_rate <- ((cancer.df$incidence / cancer.df$population)*100000)
head(cancer.df)
```

```
## # A tibble: 6 x 9
##    year site     state sex   race  mortality incidence population incidence_rate
##   <dbl> <chr>    <chr> <chr> <chr>     <dbl>     <dbl>      <dbl>          <dbl>
## 1  1999 Brain a~ alab~ Fema~ Black       0          19     623475           3.05
## 2  1999 Brain a~ alab~ Fema~ Hisp~       0           0      28101           0
## 3  1999 Brain a~ alab~ Fema~ White      83.7       110    1640665           6.70
## 4  1999 Brain a~ alab~ Male  Black       0          18     539198           3.34
## 5  1999 Brain a~ alab~ Male  Hisp~       0           0      37082           0
## 6  1999 Brain a~ alab~ Male  White     104.        145    1570643           9.23
```

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

23,191 subgroups (rows) have an incidence rate = 0

```r
sum(cancer.df$incidence_rate == 0)
```

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

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

The subgroup with the highest incidence rate is row 5797

```r
which(cancer.df[, "incidence_rate"] == max(cancer.df[, "incidence_rate"]))
```

```
## [1] 5797
```

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)

The sum(x) command is going to result in an error because the values of vector 'x' are characters. This is because the 'sum' command requires numeric values.

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

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

```r
            sort(x)
```

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

```r
            #sum(x)
```

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

The vector 'c' contains two numeric and one character value. Because character values are the least flexible, this vector took on an overall character value, which will change the original numeric values to characters. As a result, we when attempt to add the second and third values together, we get an error because they are now character values.

```r
            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

Above we are creating a data frame as opposed to a vector, which can handle multiple data types. As a result, when we include character and numeric values, they retain their data type. In the calculation command, we are calling data frame 'z' and pulling the value in the 1st row, 2nd column (7) and the 1st row, 3rd column (12) and adding together. Since they're both numeric values, this command will work and produce our output of 19.

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

```
## [1] 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. $(1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1)$

```
x <- c(1:8, seq(7,1,-1))
x
```

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

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

```
y <- rep(1:5, times = 1:5)
y
```

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

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

```
z <- 1
mz <- matrix(z, ncol = 3, nrow = 3)
diag(mz) <- 0
mz
```

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

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

```
mn <- matrix(rep(1:4, each = 5), nrow = 5)
mo <- matrix(rep(1:5, each = 4), ncol = 4, byrow = TRUE)
mn^mo
```

```
##      [,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

out = 0
for (i in 0:n)
   out = out + x^i
out
```

```
## [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])

```
sum(unique(c(seq(3, 999, 3), seq(5, 999, 5))))
```

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

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

```
sum(unique(c(seq(4, 999999, 4), seq(7, 999999, 7))))
```

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

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

Some problems taken or inspired by projecteuler.