Bios 6301: Assignment 2

55/55

AUTHOR
Yiqing Pan

Due Tuesday, 17 September, 1:00 PM

50 points total.

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

Submit a single quarto file (named `homework2.qmd`) by email to huiding.chen@vanderbilt.edu. Place your R code in between the appropriate chunks for each question. Check your output by using the `Render` 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)

```
library(dplyr)
```

```
载入程序包：'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

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

```
cancer.df <- read.csv("cancer.csv")
```

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

```
nrow(cancer.df); ncol(cancer.df)
```

```
[1] 42120
```

```
[1] 8
```

```
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 <- cancer.df %>% mutate(incidence_rate = incidence/population *100000)

head(cancer.df, 10)
```

```
   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
7  1999 Brain and Other Nervous System  alaska Female    Black      0.00
8  1999 Brain and Other Nervous System  alaska Female Hispanic      0.00
9  1999 Brain and Other Nervous System  alaska Female    White      0.00
10 1999 Brain and Other Nervous System  alaska   Male    Black      0.00
   incidence population incidence_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          0      12710       0.000000
8          0      11664       0.000000
9          0     220036       0.000000
10         0      13900       0.000000
```

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

```
nrow(cancer.df %>% filter(incidence_rate == 0))
```

```
[1] 23191
```

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

```
cancer.df[which(cancer.df$incidence_rate == summary(cancer.df$incidence_rate)[6]), ]
```

```
     year     site                state   sex  race mortality incidence
5797 1999 Prostate district of columbia Male Black     88.93       420
     population incidence_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)
```

Command sum(x) will produce an error message, indicating that x is a invalid ‘type’ (character).

max(x) is a function that returns the maxima of the input values. It works since the characters will be coerced to numeric values and then the maximum will be computed.

sort(x) is a function that orders a vector into ascending or descending order. It can work since characters can be easily sorted lexicographically.

sum(x) is a function that returns the sum of all the values present in the arguments.sum(x) enters an error message since it requires numeric input from the start and does not handle implicit coercion. However, by adding “” to each numeric value here, we made x a character vector.

```
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]
    ```
```

This produces an error since one of the values we are trying to add up is non-numeric. We can not use non-numeric in arithmetic operation.

```
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] 19
```

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

With these 2 lines, we can get 19 as a result. This works since z is created as a dataframe with 1 row and 3 columns, in which z[1,2] refers to the value stored in the second column of the first row(numeric value 7) and z[1,3] refers to the value stored in the third column of the first row(numeric value 12). We can add up 2 numerical values using +, and the result is simply the addition of 7 and 12, which equals 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, times = 1:5)
```

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

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

```
        matrix(1, 3, 3) - diag(1, 3, 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
cbind(
  rep(1, 5),
  2^(1:5),
  3^(1:5),
  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 = 1
for(i in (1:n)) {
  h = h + x^i
}

h
```

```
[1] 31
```

```
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 sum of these multiples is 23. Write an R program to perform the following
calculations. (5 points)

    1.  Find the sum of all the multiples of 3 or 5 below 1,000. (3, [euler1]
(https://projecteuler.net/problem=1))
```

```r
multi_3 <- NULL
multi_5 <- NULL

for (i in (1:999)) {
  if (i%%3 == 0) {
    multi_3 <- c(multi_3, i)
  } else if (i%%5 == 0) {
    multi_5 <- c(multi_5, i)
  }
}

sum(multi_3, multi_5)
```

```
[1] 233168
```

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

```
sum_multi_4or7 = 0

for (i in (1:999999)) {
  if ((i%%4 == 0)|(i%%7 == 0)) {
    sum_multi_4or7 = sum_multi_4or7 + i
  }
}

sum_multi_4or7
```

```
[1] 178571071431
```

3. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be $(1, 2, 3, 5, 8, 13, 21, 34, 55, 89)$. Write an R program to calculate the sum of the first 15 even-valued terms. (5 bonus points, [euler2](https://projecteuler.net/problem=2))

```
even_val <- c(2)
i <- 1

a = 1
b = 2

while (i < 15) {
  new_a = b
  b = a+b
  a = new_a
  if (b %% 2 == 0) {
    i = i + 1
    even_val[i] = b

  }
}

sum(even_val)
```

```
[1] 1485607536
```

Some problems taken or inspired by projecteuler.