Bios 6301: Assignment 2

Erin Fey

04 October 2016

(informally) Due Tuesday, 20 September, 1:00 PM

50 points total.

This assignment won't be submitted until we've covered Rmarkdown. Create R chunks for each question and insert your R code appropriately. Check your output by using the `Knit PDF` 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)

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

```
cancer.df <- read.csv("https://github.com/fonnesbeck/Bios6301/raw/master/datasets/cancer.csv", header=T
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)

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

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

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

```
## [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
## 172 1999 Brain and Other Nervous System nevada Male Black         0
##     incidence population
## 172         0      73172
```

6. Create a new column that is the incidence *rate* (per 100,000) for each row.(3)

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

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

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

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

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

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

```
##      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)
3. 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"
```

The characters are sorted according to their first digit , therefore 7>5>1 and you return the value “7”

```r
sort(x)
```

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

Again the characters are sorted by their first digit 1<5<7, you return the values “12” “5” “7”

sum(x)

Error, the class of vector x is a character and you can only sum numeric complex or logical values “‘ 2. For the next two commands, either explain their results, or why they should produce errors. (3 points)

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

Since “5” is a character, the whole vector is read as a character because it is the least flexible and you return the values “5” “7” “12”

y[2] + y[3]

Error, again here you are trying to take the sum of two characters because when you call y[2] you are returning the element found at the second position of y but it is returned in character form

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)
```

We get a data frame with z1 z2 and z3 the column names and 5 7 12 in the first row #since we have assigned z as a data frame, this function converts character variables to factors

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

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

Addition works here because the values are numeric and you return a value of 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)

4. $(1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1)$

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

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

```r
x <- rep(1:5, c(1,2,3,4,5))
```

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

```
  matrix(1, 3, 3) - diag(3)
```

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

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

```
mx <- matrix(rep(1:4, 5),5,4,byrow = TRUE)
(mx <- mx ^ row(mx))
```

```
##      [,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)

5. 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. (5 points)

```
h <- function(x, n){
  sum = 0
  for (i in seq(n)){
    sum = sum + x^i
  }
  return(sum)
}
```

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)

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

```
sum <- function(x = 1000){
  sum = 0
  for (i in 1:x-1){
    if(i%%3 == 0 | i%%5 == 0)
      sum = sum + i
  }
  return(sum)
}
sum()
```

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

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

```
sum <- function(x = 1000000){
  sum = 0
  for (i in 1:x-1){
    if(i%%4 == 0 | i%%7 == 0)
      sum = sum + i
  }
  return(sum)
}
sum()
```

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

4. 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]) c(1, 2, 3, 5, 8, 13, 21, 34, 55, 89)

count <- 1 sum <- 2 while (count < 15) { i <- i+1 x<-c(x,x[i]+x[i+1]) if (x[i+1]%%2==0) { count <- c(count,x[i+1]) sum <- sum+1 }
sum “‘