Bios 6301: Assignment 2

Lingjun Fu

07 October, 2015

(informally) Due Thursday, 17 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)

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

```
library(RCurl)
```

```
## Loading required package: bitops
```

```
x <- getURL("https://raw.githubusercontent.com/fonnesbeck/Bios6301/master/datasets/cancer.csv")
cancer.df=read.csv(text = x)
```

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"] <- NA
cancer.df$incidence_rate <- cancer.df$incidence / 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 population
## 21387 2002 Breast california Female White   3463.74     18774   13690681
##       incidence_rate
## 21387        0.18774
```

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)`: "7". R sorts characters by the first digit (7>5>1)
        `sort(x)`: "12" "5" "7". R sorts characters by the first digit (1<5<7)
        `sum(x)`: x is a chracter, while sum needs arguments as numeric or complex or logical vectors.

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

        `y <- c("5",7,12)`: "5" "7" "12". All are assigned as chracters as the first element.
        `y[2] + y[3]`: the type of elements in y is chracter, which cannot be added.

    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)`: a data frame with "z1", "z2", "z3" as the column name and 5, 7, 12 as the first row
        `z[1,2] + z[1,3]`: 19. Both are numeric values.

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, 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(3)
```

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

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

```
pivot<-c(1:4)
matrix(c(pivot,pivot^2,pivot^3,pivot^4,pivot^5), nrow = 5, ncol = 4, byrow = T)
```

```
##      [,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. (5 points)

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

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_3or5 = function(x = 1000){
    sum = 0
    for (i in 1:x-1){
        if(i%%3 == 0 | i%%5 == 0)
        sum = sum + i
    }
    return(sum)
}
sum_3or5()
```

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

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

```
sum_4or7 = function(x = 1000000){
    sum = 0
    for (i in 1:x-1){
        if(i%%4 == 0 | i%%7 == 0)
        sum = sum + i
    }
    return(sum)
}
sum_4or7()
```

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

1. 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])

```
sum_Fibonacci = function(n = 15){
    f1 = 1
    f2 = 2
    count = 1
    sum = 2
    while (count < n){
        temp1 = f1
        temp2 = f2
        f1 = temp2
        f2 = temp1 + temp2
        if(f2%%2 == 0){
            count = count + 1
            sum = sum + f2
        }
    }
    return(sum)
}
sum_Fibonacci()
```

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

Some problems taken or inspired by projecteuler.