Bios 6301: Assignment 2

Alexander Thiemicke

2015-10-08

50 points total.

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)

```
cancer.df = read.csv("cancer.csv")
```

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

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

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

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

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

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"
```

[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
```

[1] 350.69 5. Report the contents of the 172nd row. (2)

```
cancer.df[172, 1:8]
```

```
##     year                           site  state  sex  race mortality
## 172 1999 Brain and Other Nervous System nevada Male Black         0
##     incidence population
## 172         0      73172
```

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

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

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

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

[1] 23191 8. Find the subgroup with the highest incidence rate.(3)

```
which.max(cancer.df$rate)
```

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

[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

        max(x)
        sort(x)
        sum(x)
```

x <- c("5","12","7") max(x) [1] "7" sort(x) [1] "12" "5" "7" sum(x) Error in sum(x) : invalid 'type' (character) of argument #only sum(x) will produce an error meassage, because the data are character objects # and can not be summed up

```
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]
```

y <- c("5",7,12) y[2] + y[3] Error in y[2] + y[3] : non-numeric argument to binary operator #If a vector contains different data types, not only numbers, #the numbers in the vector can not be used for mathematical #calculations in this way anymore.

```
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[1,2] + z[1,3][1] 19 # The first command creates a data frame with 3 columns and 1 row, #The second command adds the values of the of the second and third #column in the first row.

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(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,c(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(c(0,1,1,1,0,1,1,1,0), ncol = 3, nrow = 3, byrow = TRUE)
```

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

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

```
f <- c(1,2,3,4)
matrix(c(f,f^2,f^3,f^4,f^5), ncol = 4, nrow = 5, byrow = TRUE)
```

```
##      [,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)

```
#n=0
#x=0
#for (i in seq (along=n)){
#h(x,n)=1+x+x^2+x^n=sum(i=0)^n = sum(x^i)
#}
```

```
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])
```

```
totala =  0
for (n in 1:999){
  if ((n%%3==0) || (n%%5==0)){
    totala <- totala +n
  }
}
totala
```

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

[1] 234167 1. Find the sum of all the multiples of 4 or 7 below 1,000,000. (2)

```
totalb = 0
for (n in 1:999999){
  if ((n%%4==0) || (n%%7==0)){
    totalb <- totalb +n
  }
}
totalb
```

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

[1] 178571071431

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

```r
#assure value is numeric
c <- numeric()
#define starting values
c[1] <- 1
c[2] <- 1
i <- 3
#create repeat loop, end if 3x15 as only every 3rd value in
#fibonacci series is even
repeat{
  c[i] <- c[i-1] + c[i-2]
  if (i > 45) break
  i <- i + 1
}

c <- c[1:i-1]

#add all even values together
sum(c[c %% 2 == 0])
```

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

[1] 1485607536

Some problems taken or inspired by projecteuler.