Bios 6301: Homework 2

Zi Ye

(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)

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

```
setwd('~/Downloads/Bios6301-master/datasets')
x <- data.frame(read.csv('cancer.csv'))
```

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

```
nrow(x)
```

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

```
ncol(x)
```

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

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

```
names(x)
```

```
## [1] "year"       "site"       "state"      "sex"        "race"
## [6] "mortality"  "incidence"  "population"
```

    4. Report the value of the 3000th row in column 6. (2)

```
x[3000,6]
```

```
## [1] 350.69
```

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

```
x[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)

```
incirate <- x$incidence/100000
x <- data.frame(x, incirate)
```

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

```
nrow(x[incirate==0,])
```

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

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

```
x[incirate==max(x$incirate),]
```

```
##       year   site      state    sex  race mortality incidence population
## 21387 2002 Breast california Female White   3463.74     18774   13690681
##       incirate
## 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)
sort(x)
sum(x)
```

Since the number within quote mark will become characters, `x` is not a combination of numbers but characters. `sum(x)` returns error message since it is impossible to `sum()` characters. `max(x)` can be used on characters but it only takes the first letter in each element into account ('12' becomes '1'), and so as `sort(x)`.

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]
```

It returns errors. The first element in `y` is a character. The `c()` function will switch all the elements to character if there are both numbers and characters. Therefore, `y[2]` becomes the character '7' and `y[3]` becomes the character '12', and it is impossible to add them together.

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]
```

It returns 19. As opposed to `c()` function, `data.frame()` function switchs characters to numbers when both characters and numbers are included in the data frame. So the second and third elements can be added up to 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(seq(8), seq(7, by=-1))
```

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

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

```
c(1,rep(2,2),rep(3,3),rep(4,4),rep(5,5))
```

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

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

```
m1 <- matrix(1, nrow=3, ncol=3)
diag(m1) <- 0
m1
```

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

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

```
m2 <- matrix(ncol=4, nrow=5, byrow=T)
for (i in 1:4) {
m2[,i]=i^(1:5)
}
m2
```

```
##      [,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 = 0
for (i in 0:n) {
h = h + x^i
}
h
```

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)

```
x = 0
for (i in 1:999) {
if (i%%3==0 | i%%5==0) {
x = x+i
}
}
x
```

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

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

```
y = 0
for (i in 1:(1e6-1)) {
if (i%%4==0 | i%%7==0) {
y = y+i
}
}
y
```

```
## [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)

```
z = c(1, 2)
even = c(2)
for (i in 3:1000) {
  z[i] = z[i-1] + z[i-2]
  if (z[i]%%2==0 & length(even) < 16) {
    even = c(even, z[i])
  }
}
sum(even)
```

```
## [1] 6293134512
```

Some problems taken or inspired by projecteuler.