Assignment 2

James Lopez

October 11, 2016

Grade: 48/50

1.

part i.

```
#setwd("~/Bios 6301")
cancer.df <- read.csv("cancer.csv", header = TRUE)
```

part ii.

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

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

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

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

```
# There are 42120 rows and 8 columns in cancer.df
```

part iii.

```
colnames(cancer.df)
```

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

```
#The column names of cancer.df are "year", "site", "state", "sex", "race", "mortality", "incidence", an
```

part iv.

```
cancer.df[3000,6]
```

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

```
# The value in the 3000th row and 6th column in cancer.df is 350.69
```

part v.

```
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
```

part vi.

```
cancer.df[,'incidence rate'] <- cancer.df[,'incidence']/100000
```

JC Grading - 1 For incidence rate above should be incidence / population * 100000

part vii.

```
sum(cancer.df[,'incidence rate'] == 0)
```

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

```
#There are 23191 subgroups with a zero incidence rate
```

part viii.

```
which.max(cancer.df[,'incidence rate'])
```

```
## [1] 21387
```

```
cancer.df[21387,]
```

```
##       year   site      state    sex  race mortality incidence population
## 21387 2002 Breast california Female White   3463.74     18774   13690681
##       incidence rate
## 21387        0.18774
```

```
#The subgroup with the highest incidence rate is in row 21387 of cancer.df, during the year 2002 in Cal
```

JC Grading - 1 syntax is fine but answer is incorrect b/c of how incidence rate was calculated

2.

part i.

The sum() function should produce an error because it only works with numeric data types. The quotations around the numbers in x assigns the numbers as characters instead of numerics.

```
x <- c("5","12","7")
max(x)
```

```
## [1] "7"
```

```
sort(x)
```

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

```
#sum(x) **Not included as an R statement in rmd due to compiling issues
```

The max is “7” because “12” begins with a 1 and “5” begins with a 5. Since x is a character vector, the “numbers” will be treated like characters and be sorted by the first character in the whole string. “Max” will not produce an error.

Because x has numbers as characters, the value of 12 has the smallest value because it starts with a 1, while 7 has the largest value because it starts with a 7. So “Sort” will not produce an error.

part ii.

The next two commands will produce an error message because the elements in vector y do not have the same data type. The numeric elements 7 and 12 will be coerced into characters due to the presence of “5” Because the vector will coerce the 7 and 12 into characters, the + operator will not work.

```
#y <- c("5",7,12)
#y[2] + y[3]

# NOTE: Not included as R statements due to compiling issues
```

part iii.

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

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

The + operator returned a number 19. z was made into a data frame, which allows the elements of z to retain their original data type. z[1,2] and z[1,3] called for the numeric elements in z, so the + operator was able to be used to return numeric 19.

3.

part i.

```
c(1:8, 7:1)
```

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

part ii.

```
rep(1:5, 1:5)
```

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

part iii.

```r
m <- matrix(1, nrow = 3, ncol = 3)
for(i in 1:3) {
  m[i,i] <- 0
}
m
```

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

part iv.

```r
n <- matrix(c(1:4), nrow = 4, ncol = 4, byrow = TRUE)
for (i in 1:4) {
  n[i,] <- n[i,]^i
}
n
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    1    4    9   16
## [3,]    1    8   27   64
## [4,]    1   16   81  256
```

4.

part i.

I made a function that tried to follow the equation posted. To demonstrate the function, I randomly selected both the base and the exponential order between 1-10.

```r
h <- function(x,n){
n <- sample.int(10,1)
x <- sample.int(10,1)
for(i in 1:n) {
  print(sum(x^(1:i)))
}
}
h(x,n)
```

```
## [1] 10
## [1] 110
```

part ii. a.

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

```
## [1] 233169
```

part ii. b.

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

```
## [1] 178571071432
```

JC Grading +0 Bonus