

Bios 6301: Assignment 2

Ying Ji

(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("~/Documents/biostat_computing/bios6301_homework/")
cancer.df<-data.frame(read.csv("cancer.csv"))
```

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

```
dim(cancer.df)
```

```
## [1] 42120      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)

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

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

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)

`max(x)`: returned "7", not the max number 12. The reason is that `x` is a character vector, so the order of elements is determined by the first character.
`sort(x)`: returned ("12", "5", "7"), sort return the order of the vector, from first number of each element.
`sum(x)`: produced an error "invalid type of argument", because the argument should be numeric type

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]
```

produced errors: non-numeric argument to binary operator

`y` is a vector with character "5" and numeric element 7 and 12, after combination using `c()` the type is character

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]
```

result is 19, because `z[1,2]` and `z[1,3]` are all of numeric type. Data.frame keep the original class of elements

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, 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,times=1:5)
```

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

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

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

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

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

```
a<-seq(4)
pmatrix<-rbind(a,a^2,a^3,a^4,a^5)
pmatrix
```

```
##      [,1] [,2] [,3] [,4]
## a      1    2    3    4
##      1    4    9   16
##      1    8   27   64
##      1   16   81  256
##      1   32  243 1024
```

4. Basic programming (10 points)

1. Let $h(x, n) = 1 + x + x^2 + \dots + x^n = \sum_{i=0}^n x^i$. Write an R program to calculate $h(x, n)$ using a for loop. (5 points)

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

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 sum of all these multiples is 23.

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

```
s<-0
for (i in 1:1000-1){
  if (i %%3 ==0|i %% 5==0){

    s=s+i

  }
}
s
```

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

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

```
s<-0

for (i in 1:1000000-1){
  if (i %%4 ==0|i %% 7==0){

    s=s+i

  }
}
s
```

```
## [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:

```
f<-numeric(50)
j<-1
f[1]<-1
f[2]<-2
s<-f[1]+f[2]
for (i in 3:50){
  f[i]<-f[i-2]+f[i-1]
  if (f[i] %%2 ==0 && j<15){
    s=s+f[i]
    j=j+1
  }
}
s
```

```
## [1] 1485607537
```

Some problems taken or inspired by [projecteuler](#).