THE UNIVERSITY OF AUCKLAND

SEMESTER TWO, 2020
Campus: City, NZ Online, Offshore Online,
UoA CLC - Northeast Forestry,
UoA CLC - Southwest University

STATISTICS

Advanced Data Science Practice

(Time allowed: TWO Hours)

INSTRUCTIONS

- Attempt ALL questions.
- Total marks are 70.

1. **[10 marks]**

This question relates to a CSV file called `"linux-prosper.csv"`. The first few lines of the file are shown below. There are many more lines like this in the file.

```
ID,age,sex,city,province
000-1-10449,,,Haibei Prefecture,Qinghai
000-1-1045,,,Pingdingshan City,Henan
000-1-10450,,,Haibei Prefecture,Qinghai
000-1-10451,40-49,male,,Saitama
000-1-10452,,,Xi'an City,Shaanxi
000-1-10453,,,Xi'an City,Shaanxi
000-1-10454,,,Xi'an City,Shaanxi
000-1-10676,,,Bexar County,Texas
000-1-10677,,,,Tianjin
000-1-10678,,,,Tianjin
000-1-10679,20-29,male,Haneda Airport,Tokyo
000-1-1068,,,Tongliang District,Chongqing
000-1-10680,40-89,,,Tokyo
000-1-10681,40-89,,,Tokyo
000-1-10682,40-89,,,Tokyo
000-1-10683,40-89,,,Tokyo
000-1-10684,40-89,,,Tokyo
000-1-10685,40-89,,,Tokyo
000-1-10686,40-89,,,Tokyo
```

(a) **Write a shell command** that uses `awk` to extract just the `id` and `province` for the rows of the CSV file for which the `age` column is an age range (i.e., the `age` column contains a dash character, like this: `40-89`).

The first few lines of output from your command would look like this:

```
000-1-10451 Saitama
000-1-10679 Tokyo
000-1-10680 Tokyo
000-1-10681 Tokyo
000-1-10682 Tokyo
000-1-10683 Tokyo
000-1-10684 Tokyo
000-1-10685 Tokyo
000-1-10686 Tokyo
```

[5 marks]

(b) **Explain** what the following shell command is doing and **write down the output** from the command.

```
head -20 linux-prosper.csv | grep Tokyo | wc -l
```

[5 marks]

2. **[10 marks]**

(a) The image below shows a web page located at the URL:
`https://www.stat.auckland.ac.nz/stats769/2020/donations.html`

Electoral Donations

The table below shows the total amount donated to each political party (for the top four parties) in the lead up to the 2014 general election.

Party	Amount Donated
National Party	$1,260,000
Labour Party	$670,000
Internet Party	$230,000
Maori Party	$220,000

The HTML code for that web page is shown below.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <style>
    body { padding: 150 }
    td.num { text-align: right }
  </style>
</head>
<body>
  <h1>Electoral Donations</h1>
  <p>
    The table below shows the total amount donated to each political
    party (for the top four parties) in the lead up to the
    2014 general election.
  </p>
  <table summary="Top four total donations by party">
    <tr>
      <td class="head">Party</td><td class="head">Amount Donated</td>
    </tr>
    <tr>
      <td>National Party</td><td class="num">$1,260,000</td>
    </tr>
    <tr>
      <td>Labour Party</td><td class="num">$670,000</td>
    </tr>
    <tr>
      <td>Internet Party</td><td class="num">$230,000</td>
    </tr>
    <tr>
      <td>Maori Party</td><td class="num">$220,000</td>
    </tr>
  </table>
</body>
</html>
```

Write R code that uses the **httr** package to download the web page and uses the **xml2** package, and one or more XPath expressions, to extract the numeric values from the table *as a numeric vector*.

Do NOT use the `rvest::html_table()` function for this question.

The result of your code would look like this:

```
[1] 1260000  670000  230000  220000
```

[5 marks]

(b) This question relates to the file `covid.json`, shown below.

```
{
    "ABW": {
        "continent": "North America",
        "location": "Aruba",
        "population": 106766.0,
        "data": [
            {
                "date": "2020-03-13",
                "new_cases": 2.0,
                "new_deaths": 0.0
            },
            {
                "date": "2020-03-20",
                "new_cases": 2.0,
                "new_deaths": 0.0
            },
            {
                "date": "2020-03-24",
                "new_cases": 8.0,
                "new_deaths": 0.0
            },
            {
                "date": "2020-03-25",
                "new_cases": 5.0,
                "new_deaths": 0.0
            }
        ]
    }
}
```

Write R code that uses functions from the **jsonlite** package to read the `covid.json` file and to create a data frame containing the `date`, `new_cases`, and `population` values.

The output of your code would look like this:

```
        date new_cases population
1 2020-03-13         2     106766
2 2020-03-20         2     106766
3 2020-03-24         8     106766
4 2020-03-25         5     106766
```

[5 marks]

3. **[10 marks]**

This question relates to a CSV file called `"linux-prosper.csv"`. The first few lines of the file are shown below. There are many more lines like this in the file.

```
ID,age,sex,city,province
000-1-10449,,,Haibei Prefecture,Qinghai
000-1-1045,,,Pingdingshan City,Henan
000-1-10450,,,Haibei Prefecture,Qinghai
000-1-10451,40-49,male,,Saitama
000-1-10452,,,Xi'an City,Shaanxi
000-1-10453,,,Xi'an City,Shaanxi
000-1-10454,,,Xi'an City,Shaanxi
000-1-10676,,,Bexar County,Texas
000-1-10677,,,,Tianjin
000-1-10678,,,,Tianjin
000-1-10679,20-29,male,Haneda Airport,Tokyo
000-1-1068,,,Tongliang District,Chongqing
000-1-10680,40-89,,,Tokyo
000-1-10681,40-89,,,Tokyo
000-1-10682,40-89,,,Tokyo
000-1-10683,40-89,,,Tokyo
000-1-10684,40-89,,,Tokyo
000-1-10685,40-89,,,Tokyo
000-1-10686,40-89,,,Tokyo
```

If the file `linux-prosper.csv` has 1,000,000 rows and we read it into R with `read.csv()`, **calculate** a rough estimate of the size (in terms of bytes) of the resulting data frame.

Explain your calculation and **describe any assumptions** that you are making in your calculation.

[10 marks]

4. **[10 marks]**

The following code reads a CSV file into an R data frame and generates some new R objects from values in the data frame.

```
> owid <- read.csv("owid-covid-data.csv")
> dim(owid)

[1] 47116    41

> head(owid$new_cases)

[1]  2 NA  2 NA NA NA

> cases <- owid$new_cases
> positive <- cases > 0
> pos_cases <- cases[positive]
> log_cases <- log(pos_cases)
> object.size(cases)

376976 bytes

> object.size(positive)

188512 bytes

> object.size(pos_cases)

252720 bytes

> object.size(log_cases)

252720 bytes
```

(a) **Explain** the size of each of the new objects (`cases`, `positive`, `pos_cases`, and `log_cases`).

[5 marks]

(b) **Write R code** that performs the same task, but uses functions from the **data.table** package to read the CSV file into a `data.table` object and uses the special **data.table** syntax to create a new column in the `data.table` that contains the logarithm of the `new_cases` column.

[5 marks]

5. **[10 marks]**

(a) This question is based on a text file, `path.txt`, that contains thousands of lines in the format shown below (a word, either `moveto` or `lineto`, followed by a space, followed by two numbers separated by a comma).

```
moveto -142.26372,227.62195
lineto 0.0,0.0
lineto 142.26372,227.62195
```

The R code below reads the file `path.txt` and extracts the numbers, creating a numeric vector `x` with the first number from each row and a numeric vector `y` with the second number from each row.

Identify any inefficiencies in the R code below and **write new R code** that performs the same task, but is more efficient.

```
> path <- readLines("path.txt")
> x <- numeric()
> y <- numeric()
> for (i in 1:length(path)) {
      coords <- strsplit(path[i], " ")[[1]][2]
      cx <- strsplit(coords, ",")[[1]][1]
      cy <- strsplit(coords, ",")[[1]][2]
      x <- c(x, as.numeric(cx))
      y <- c(y, as.numeric(cy))
  }
> head(x, 3)
[1] -142.2637    0.0000  142.2637
> head(y, 3)
[1] 227.6219   0.0000 227.6219
```

[5 marks]

(b) **Explain** what each expression in the following R code is doing and why the two time measurements shown in the output are different.

```
> cases <- read.csv("covid-cases.csv")

> train <- sample(1:nrow(cases), .8*nrow(cases))

> system.time(glm(deceased ~ agenum, cases[train,],
                  family="binomial",
                  control=list(trace=TRUE)))

Deviance = 80527.71 Iterations - 1
Deviance = 34497.45 Iterations - 2
Deviance = 20090.43 Iterations - 3
Deviance = 15772.11 Iterations - 4
Deviance = 14614.81 Iterations - 5
Deviance = 14376.48 Iterations - 6
Deviance = 14358.03 Iterations - 7
Deviance = 14357.87 Iterations - 8
Deviance = 14357.87 Iterations - 9

   user  system elapsed
  1.552   0.042   1.648

> start <- coef(glm(deceased ~ agenum, cases[train,],
                    family="binomial"))
> system.time(glm(deceased ~ agenum, cases[train,],
                  family="binomial",
                  start=start, control=list(trace=TRUE)))

Deviance = 14357.87 Iterations - 1
   user  system elapsed
  0.798   0.003   0.824
```

[5 marks]

6. **[10 marks]**

The following R code and output shows the execution time for three different sets of code that all produce exactly the same result.

```
> system.time(m1 <- mean(sapply(1:10, f)))

   user  system elapsed
 16.789   0.065  16.853

> system.time(m2 <- mean(unlist(mclapply(1:10, f, mc.cores=10))))

   user  system elapsed
 18.895   3.276   2.860

> system.time({
      cl <- makeCluster(10)
      clusterExport(cl, c("x", "y"))
      m3 <- mean(unlist(parLapply(cl, 1:10, f)))
      stopCluster(cl)
  })

   user  system elapsed
  0.251   0.091   3.053

> m1 == m2 && m2 == m3

 [1] TRUE
```

Explain the differences in `elapsed` and `user` time between the three sets of `system.time()` output.

Also **explain** the purpose of the `clusterExport()` function call and suggest why this may be necessary.

It is NOT necessary to know what values have been assigned to the symbols `f`, `x`, and `y` in order to answer this question.

[10 marks]

7. **[10 marks]**

This question relates to a data frame, `covid`, with two columns, `age` and `outcome`. There are many thousands of rows in the data frame, but just the first few rows are shown below.

```
> head(covid)
  age                                          outcome
1     critical condition, intubated as of 14.02.2020
2  78                                            death
3  61                                        discharge
4
5
6
```

The code below performs calculations with the values from the `covid` data frame using functions from the **rmr2** package. Some set up code has been left out for simplicity.

```
library(rmr2)

hdp.covid <- to.dfs(covid)

result <- mapreduce(input=hdp.covid,
                    map=function(k, v) {
                        deceased <- v$outcome %in%
                            c("dead", "Dead",
                              "death", "Death",
                              "Deceased",
                              "died", "Died")
                        keyval(deceased, as.numeric(v$age))
                    },
                    reduce=function(k, v) {
                        keyval(k[1], mean(v, na.rm=TRUE))
                    })

kv.result <- from.dfs(result)
kv.result
```

(a) **Explain** what each expression in the code is doing.

[5 marks]

(b) **Write down** what the output of the code would look like. You will not be able to know all of the exact values, but you can describe the overall structure of the output and what types of values will be returned.

Write R code that would perform an equivalent calculation using only standard R functions (no functions from add-on packages).

[5 marks]