THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2017
Campus: City

STATISTICS

Data Science Practice

(Time allowed: THREE hours)

NOTE:
- This examination consists of **11** questions.
- The marks for all questions sum to **100**.
- Questions are **NOT** worth equal marks.
- You should attempt **ALL** questions.
- For questions where you are required to write computer code, if you do not know the exact code, you can still gain some of the marks by writing an approximation of what the code should be.

1. **[5 marks]**

Figure 1 shows the content of an HTML file, `"2017-07-29.html"`. This is a simplified extract from the HOT 100 songs web site that was used for the web scraping lab.

Write five R expressions that use functions from the **xml2** package (and XPath expressions) to perform the following steps:

- Read the HTML file into R.
- Extract the song title from the HTML (output shown below).

```
[1] "Despacito"
```

- Extract the artist name from the HTML (output shown below; note that white space has been removed).

```
[1] "Luis Fonsi & Daddy Yankee Featuring Justin Bieber"
```

- Extract the song rank from the HTML (output shown below).

```
[1] "1"
```

- Extract the rank from the previous week from the HTML (output shown below; note that the result is more than one character value).

```
[1] "Last Week" "1"
```

```
<!doctype html>
<html class="" lang="">
<body>
<article class="chart-row chart-row--1" data-songtitle="Despacito">
  <div class="chart-row__primary">
    <div class="chart-row__history chart-row__history--steady"></div>
    <div class="chart-row__main-display">
      <div class="chart-row__rank">
        <span class="chart-row__current-week">1</span>
        <span class="chart-row__last-week">Last Week: 1</span>
      </div>
      <div class="chart-row__container">
        <div class="chart-row__title">
          <h2 class="chart-row__song">Despacito</h2>
          <a class="chart-row__artist" data-tracklabel="Artist Name">
            Luis Fonsi &amp;amp; Daddy Yankee Featuring Justin Bieber
          </a>
        </div>
      </div>
    </div>
  </div>
  <div id="chart-row-1-secondary" class="chart-row__secondary">
    <div class="chart-row__stats">
      <div class="chart-row__last-week">
        <span class="chart-row__label">Last Week</span>
        <span class="chart-row__value">1</span>
      </div>
      <div class="chart-row__top-spot">
        <span class="chart-row__label">Peak Position</span>
        <span class="chart-row__value">1</span> </div>
      <div class="chart-row__weeks-on-chart">
        <span class="chart-row__label">Wks on Chart</span>
        <span class="chart-row__value">26</span> </div>
    </div>
  </div>
</article>
</body>
</html>
```

Figure 1: The HTML file `"2017-07-29.html"`.

2. **[5 marks]**

Write a paragraph explaining the purpose of the `flatten()` function from the **jsonlite** package. You should provide at least one example of its use.

3. **[10 marks]**

Explain what each of the following shell commands is doing and, where there is output, what the output means. These commands were all run on one of the virtual machines that were used in the course.

```
pmur002@stats769prd01:~/$ mkdir exam

pmur002@stats769prd01:~/$ cd exam

pmur002@stats769prd01:~/exam$ ls -1 /course/AT/BUSDATA/ | wc -l
98973

pmur002@stats769prd01:~/exam$ ls -l /course/AT/BUSDATA/ | awk '{ print($5) }' > sizes.txt

pmur002@stats769prd01:~/exam$ head sizes.txt

343
345
345
345
436
437
438
531
438

pmur002@stats769prd01:~/exam$ grep --no-filename ',6215,' \
> /course/AT/BUSDATA/trip_updates_20170401*.csv > bus-6215-2017-04-01.csv

pmur002@stats769prd01:~/exam$ head bus-6215-2017-04-01.csv
8300033770-20170322104732_v52.21,30002-20170322104732_v52.21,6215,NA,252,6,7168,1490975922
8300033770-20170322104732_v52.21,30002-20170322104732_v52.21,6215,275,NA,7,8502,1490976035
8300033770-20170322104732_v52.21,30002-20170322104732_v52.21,6215,275,NA,7,8502,1490976035
8300033770-20170322104732_v52.21,30002-20170322104732_v52.21,6215,275,NA,7,8502,1490976035
8300033770-20170322104732_v52.21,30002-20170322104732_v52.21,6215,NA,293,9,8516,1490976233
8300033770-20170322104732_v52.21,30002-20170322104732_v52.21,6215,299,NA,9,8516,1490976239
8300033770-20170322104732_v52.21,30002-20170322104732_v52.21,6215,NA,319,10,8524,1490976349
8300033770-20170322104732_v52.21,30002-20170322104732_v52.21,6215,388,NA,10,8524,1490976418
8300033770-20170322104732_v52.21,30002-20170322104732_v52.21,6215,388,NA,10,8524,1490976418
8300033770-20170322104732_v52.21,30002-20170322104732_v52.21,6215,NA,403,11,8532,1490976523
```

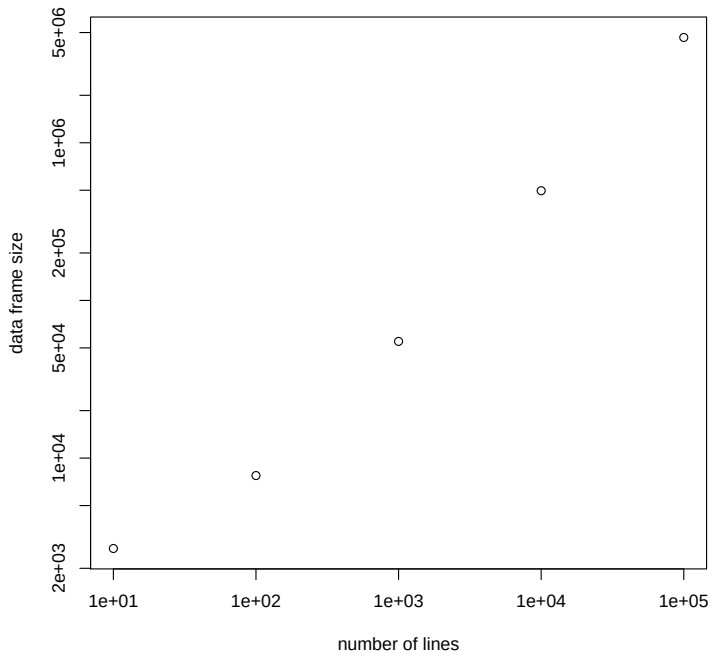
4. **[10 marks]**

The following R code was run on one of the virtual machines that were used in the course to investigate how much memory would be required to load a large CSV file with several million rows into R. The plot that this code produces is also shown.

Explain what the code is doing and **discuss** whether this will lead to a good estimate of the memory required to read the complete CSV into R. Is there another way to estimate the memory required (without reading the entire CSV file into R) ?

```
numLines <- 10^(1:5)
samples <- lapply(numLines,
                  function(i) {
                      read.csv("/course/AT/alldata.csv",
                               nrows=i, stringsAsFactors=FALSE)
                  })

plot(numLines, sapply(samples, object.size), log="xy",
     xlab="number of lines", ylab="data frame size")
```



5. **[10 marks]**

The following R code was run on one of the virtual machines that were used in the course to measure how much time is required to read different subsets of a large CSV file into R.

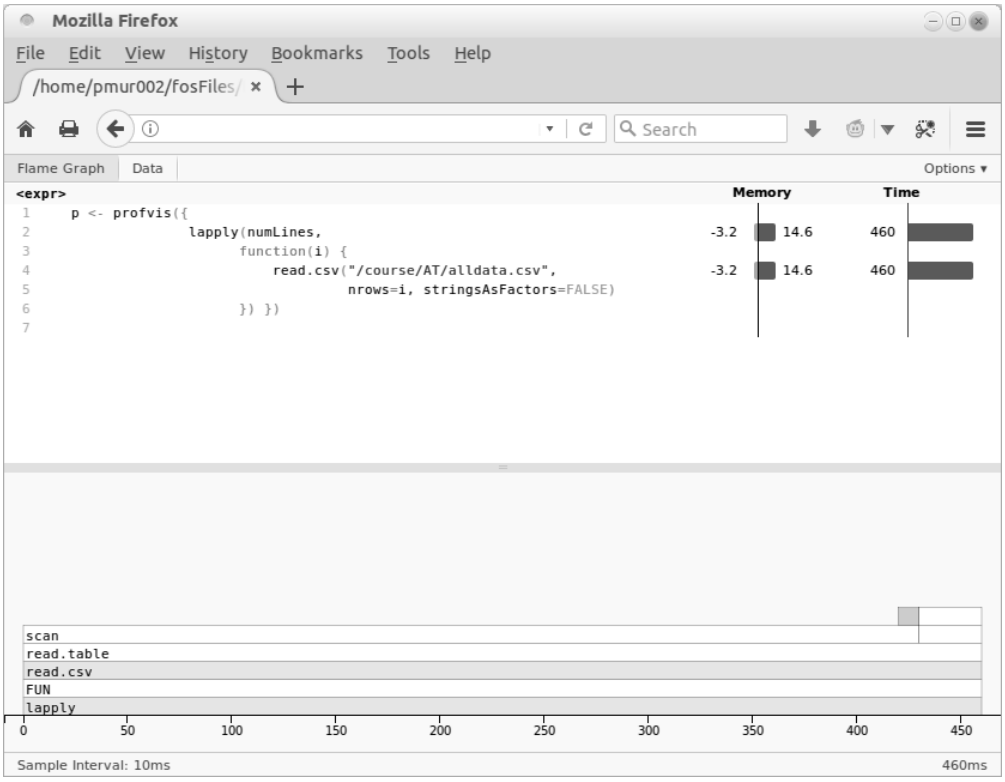
```
sapply(numLines,
       function(i) {
            system.time(read.csv("/course/AT/alldata.csv",
                        nrows=i, stringsAsFactors=FALSE))[1]
       })
```

The result of running this code is shown below.

```
user.self user.self user.self user.self user.self
    0.001     0.001     0.005     0.036     0.417
```

The following code was run to perform profiling. The profiling result is shown below the code.

```
library(profvis)
p <- profvis({
              lapply(numLines,
                     function(i) {
                         read.csv("/course/AT/alldata.csv",
                                  nrows=i, stringsAsFactors=FALSE)
                     }) })
htmlwidgets::saveWidget(p, "profile.html")
```



Explain what the timing and profiling results mean. **Suggest** how you could make the code run faster.

6. **[10 marks]**

Write R code to perform a parallel version of the `lapply()` call from Question 4. **Discuss** the advantages and disadvantages of using the `mclapply()` (forking) approach compared to the `makeCluster()` (socket) approach for this task. Also **discuss** whether load balancing would make sense for this task.

7. **[10 marks]**

Suppose we estimate the regression coefficients in a linear regression by minimizing

$$\sum_{i=1}^{n}(y_i - b_0 - b_1 x_{i1} - \cdots - b_k x_{ik})^2$$

subject to

$$\sum_{j=1}^{k} |b_j| \leq s$$

for a particular value of s. As we increase s from 0, indicate which of the following is correct. Justify your answer.

a. The training residual sum of squares ($SS_{residual}$) will increase initially, and then eventually start decreasing in an inverted U shape.

b. The training $SS_{residual}$ will decrease initially, and then eventually start increasing in a U shape.

c. The training $SS_{residual}$ will steadily increase.

d. The training $SS_{residual}$ will steadily decrease.

e. The training $SS_{residual}$ will remain constant.

8. **[10 marks]**

The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

obs	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	-1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using a K-NN classifier.

a. Compute the Euclidean distance between each observation and the test point, $X_1 = X_2 = X_3 = 0$. (*recall that the Euclidean distance between two points* (x_1, x_2, x_3) *and* (y_1, y_2, y_3) *is* $\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$).

b. What is our prediction with $K = 1$? Why?

c. What is our prediction with $K = 3$? Why?

d. If the Bayes decision boundary (optimal boundary) in this problem is highly non-linear, then would we expect the *best* value for K to be large or small? Why?

9. **[10 marks]**

I collect a set of data ($n = 100$ observations) containing a single predictor and a quantitative response. I then fit a linear regression model to the data, as well as a separate cubic regression, i.e. $Y = b_0 + b_1X + b_2X^2 + b_3X^3 + \varepsilon$.

a. Suppose that the true relationship between X and Y is linear, i.e. $Y = b_0 + b_1X + \varepsilon$. Consider the training residual sum of squares ($SS_{resdiual}$) for the linear regression, and also the training $SS_{residual}$ for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

b. Answer part (a) using test $SS_{residual}$ rather than training $SS_{residual}$.

10. **[10 marks]**

Explain how k-fold cross-validation is implemented. And what are the advantages and disadvantages of k-fold cross validation relative to the validation set approach?

11. **[10 marks]**

When the number of features p is large, there tends to be a deterioration in the performance of K-NN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is known as the **curse of dimensionality**, and it ties into the fact that non-parametric approaches often perform poorly when p is large.

a. Suppose that we have a set of observations, each with measurements on $p = 1$ feature, X. We assume that X is uniformly (evenly) distributed on $[0,1]$. Associated with each observation is a response value. Suppose that we wish to predict a test observation's response using only observations that are within 10% of the range of X closest to that test observation. For instance, in order to predict the response for a test observation with $X = 0.6$, we will use observations in the range $[0.55, 0.65]$. On average, what fraction of the available observations will we use to make the prediction?

b. Now suppose that we have a set of observations, each with measurements on $p = 2$ features, X_1 and X_2. We assume that (X_1, X_2) are uniformly distributed on $[0,1] \times [0,1]$. We wish to predict a test observation's response using only observations that are within 10% of the range of X_1 and within 10% of the range of X_2 closest to that test observation. For instance, in order to predict the response for a test observation with $X_1 = 0.6$ and $X_2 = 0.35$, we will use observations in the range $[0.55, 0.65]$ for X_1 and in the range $[0.3, 0.4]$ for X_2. On average, what fraction of the available observations will we use to make the prediction?

c. Now suppose that we have a set of observations on $p = 100$ features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation's response using observations within the 10% of each feature's range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?

d. Using your answers to parts (a)–(c), argue that a drawback of K-NN when p is large is that there are very few training observations "near" any given test observation.