[redacted]

[redacted]

by

Scott Greenberg

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

(i) (25 Points) [redacted]

In this question, we work with the `parlgov-development.db` SQL data base from `http://www.parlgov.org/#data`. See this web site for further details on the data and all available background information. However, you should work directly with the version of `parlgov-development.db` that has been posted in *HW03_Data.zip* in Canvas because we have no control if and when the original data base on the web site gets updated!

For question parts (c) to (j), use the following template:

```
sql <- "SELECT ...;"

query <- dbSendQuery(parlcon, statement = sql)
qdata <- fetch(query, n = -1)
qdata
```

Note that all question parts can (and **must**) be answered via proper SQL queries. You cannot do any further modification of the query results in R. However, if you are not sure whether your query produces the correct result, first read in the entire tables into R and then do the proper subsetting in R. Recalling the similarity between R expressions and SQL queries, translate the R subsetting into SQL queries if you can't think of a SQL query directly.

As your final answer, only include the SQL queries and the surrounding final R code from the template above, but none of the additional R code you may have used to test and verify your SQL queries.

Always show your final R code. Also show the results of your queries.

(a) (1 Point) Load all R packages you need for this homework question in this part of the question. Establish a connection (called parlcon) to the local copy of the SQL data base on your computer.

Answer:

```
library(RSQLite)
driver <- dbDriver("SQLite")
path.one <- "C:/Users/Scott/Downloads/HW03_Data/"
path.two <- "HW03_Data/parlgov-development.db"
path <- paste0(path.one, path.two)
parlcon <- dbConnect(driver, dbname = path)
```

(b) (1 Point) Which tables are in the data base?

Answer:

```
dbListTables(parlcon)
```

```
##  [1] "cabinet"                       "cabinet_party"
##  [3] "country"                       "election"
##  [5] "election_result"               "external_commissioner_doering"
##  [7] "external_country_iso"          "external_party_benoit_laver"
##  [9] "external_party_castles_mair"   "external_party_chess"
## [11] "external_party_cmp"            "external_party_ees"
## [13] "external_party_euprofiler"     "external_party_huber_inglehart"
## [15] "external_party_ray"            "info_data_source"
## [17] "info_id"                       "info_table"
## [19] "info_variable"                 "party"
## [21] "party_change"                  "party_family"
## [23] "party_name_change"             "politician_president"
## [25] "sqlite_stat1"                  "view_cabinet"
## [27] "view_election"                 "view_party"
## [29] "view_variable"                 "viewcalc_country_year_share"
## [31] "viewcalc_election_parameter"   "viewcalc_parliament_composition"
## [33] "viewcalc_party_position"
```

(c) (2 Points) We want to figure out from the `country` table for which countries the data base offers data. To do so, load the first 5 entries from this table (in the given order).

Answer:

```
sql1 <- "SELECT * "
sql2 <- "FROM country LIMIT 5;"
sql <- paste0(sql1, sql2)
query <- dbSendQuery(parlcon, statement = sql)
qdata <- fetch(query, n = -1)
qdata
```

```
##   id        name name_short oecd_accession_date eu_accession_date iso_numeric
## 1  1    Slovakia        SVK          2000-12-14        2004-05-01         703
## 2  5       Japan        JPN          1964-04-28              <NA>         392
## 3  7  Luxembourg        LUX          1961-12-07        1958-01-01         442
## 4  8 Netherlands        NLD          1961-11-13        1958-01-01         528
## 5  9      Norway        NOR          1961-07-04              <NA>         578
##   cmp euprofiler ees      morgan castles_mair huber_inglehart ray chess
## 1  96         32  SK        <NA>                     SLOVAKIA  NA    28
## 2  71         NA                                        JAPAN  NA    NA
## 3  23         21  LU  LUXEMBOURG                                9    NA
## 4  22         24  NL NETHERLANDS          NLD     NETHERLANDS  10    10
## 5  12         NA          NORWAY          NOR          NORWAY  15    35
##   benoit_laver public
## 1           SK      1
## 2           JP      1
## 3           LU      1
## 4           NL      1
## 5           NO      1
##                                                                          comment
## 1                                                                          <NA>
## 2 Needs information or comment in coding rules for largest party that won no seats.
## 3
## 4
## 5
##   old_countryID
## 1           703
## 2           392
## 3           442
## 4           528
## 5           578
```

(d) (2 Points) Load all country names and country abbreviations from the `country` table. Sort the results in alphabetical order of the country names.

Answer:

```
sql1 <- "SELECT name, name_short FROM country "
sql2 <- "ORDER BY name ASC;"
sql <- paste0(sql1, sql2)

query <- dbSendQuery(parlcon, statement = sql)
qdata <- fetch(query, n = -1)
qdata
```

```
##              name name_short
## 1       Australia        AUS
## 2         Austria        AUT
## 3         Belgium        BEL
## 4        Bulgaria        BGR
## 5          Canada        CAN
## 6         Croatia        HRV
## 7          Cyprus        CYP
## 8  Czech Republic        CZE
## 9         Denmark        DNK
## 10        Estonia        EST
## 11        Finland        FIN
## 12         France        FRA
## 13        Germany        DEU
## 14         Greece        GRC
## 15        Hungary        HUN
## 16        Iceland        ISL
## 17        Ireland        IRL
## 18         Israel        ISR
## 19          Italy        ITA
## 20          Japan        JPN
## 21         Latvia        LVA
## 22      Lithuania        LTU
## 23     Luxembourg        LUX
## 24          Malta        MLT
## 25    Netherlands        NLD
## 26    New Zealand        NZL
## 27         Norway        NOR
## 28         Poland        POL
## 29       Portugal        PRT
## 30        Romania        ROU
## 31       Slovakia        SVK
## 32       Slovenia        SVN
## 33          Spain        ESP
## 34         Sweden        SWE
## 35    Switzerland        CHE
## 36         Turkey        TUR
## 37 United Kingdom        GBR
```

(e) (2 Points) Which time span is covered by the `election` table? Only return the earliest and latest dates.

Answer:

```
sql1 <- "SELECT MAX(date), MIN(date) "
sql2 <- "FROM election LIMIT 1;"
sql <- paste0(sql1, sql2)

query <- dbSendQuery(parlcon, statement = sql)
qdata <- fetch(query, n = -1)
qdata

##    MAX(date)  MIN(date)
## 1 2020-03-02 1900-05-27
```

(f) (3 Points) What were the early elections in Sweden, Latvia, Austria, and the United Kingdom? Assume that "1" represent TRUE, i.e., an early election. Use country_id from the `election` table and id from the `country` table to join the two tables. Return the country_id, date, country name, and early. Sort by country name and date within country (in chronological order).

Answer:

```
sql1 <- "SELECT E.country_id, E.date, C.name, E.early "
sql2 <- "FROM election AS E LEFT JOIN country AS C "
sql3 <- "ON E.country_id = C.id "
sql4 <- "WHERE (C.name = 'Sweden' OR C.name = 'Latvia' OR "
sql5 <- "C.name = 'Austria' OR C.name = 'United Kingdom') "
sql6 <- "AND E.early = 1 ORDER BY C.name, E.date;"
sql <- paste0(sql1, sql2)
sql <- paste0(sql, sql3)
sql <- paste0(sql, sql4)
sql <- paste0(sql, sql5)
sql <- paste0(sql, sql6)

query <- dbSendQuery(parlcon, statement = sql)
qdata <- fetch(query, n = -1)
qdata
```

```
##    country_id       date           name early
## 1          59 1920-10-17        Austria     1
## 2          59 1953-02-22        Austria     1
## 3          59 1956-05-13        Austria     1
## 4          59 1959-05-10        Austria     1
## 5          59 1962-11-18        Austria     1
## 6          59 1966-03-06        Austria     1
## 7          59 1971-10-10        Austria     1
## 8          59 1979-05-06        Austria     1
## 9          59 1983-04-24        Austria     1
## 10         59 1986-11-23        Austria     1
## 11         59 1990-10-07        Austria     1
## 12         59 1995-12-17        Austria     1
## 13         59 2002-11-24        Austria     1
## 14         59 2006-10-01        Austria     1
## 15         59 2008-09-28        Austria     1
## 16         55 1998-10-03         Latvia     1
## 17         55 2011-09-17         Latvia     1
## 18         35 1958-06-01         Sweden     1
## 19         44 1923-12-06 United Kingdom     1
## 20         44 1924-10-29 United Kingdom     1
## 21         44 1931-10-27 United Kingdom     1
## 22         44 1951-10-25 United Kingdom     1
## 23         44 1955-05-26 United Kingdom     1
## 24         44 1966-03-31 United Kingdom     1
## 25         44 1974-10-10 United Kingdom     1
## 26         44 1979-05-03 United Kingdom     1
## 27         44 2017-06-08 United Kingdom     1
```

```
## 28         44 2019-12-12 United Kingdom     1
```

(g) (3 Points) Take a closer look at the most recent early election from the United Kingdom. Return the country_id, date, country name, and the Wikipedia link.

Keep in mind that you must obtain the final result through your SQL query. You are not allowed to display the tail of your resulting data frame or limit fetch to one single record in R.

Answer:

```
sql1 <- "SELECT E.country_id, MAX(E.date), C.name, "
sql2 <- "E.wikipedia FROM election AS E "
sql3 <- "LEFT JOIN country AS C ON E.country_id = C.id "
sql4 <- "WHERE C.name = 'United Kingdom' "
sql5 <- "AND E.early = 1;"
sql <- paste0(sql1, sql2)
sql <- paste0(sql, sql3)
sql <- paste0(sql, sql4)
sql <- paste0(sql, sql5)


query <- dbSendQuery(parlcon, statement = sql)
qdata <- fetch(query, n = -1)
qdata

##   country_id MAX(E.date)           name
## 1         44  2019-12-12 United Kingdom
##                                                          wikipedia
## 1 https://en.wikipedia.org/wiki/2019_United_Kingdom_general_election
```

(h) (3 Points) Which country had cabinets lead by (Robert) Menzies? Return the country name, cabinet name, and start_date. Sort by start_date (in chronological order).

Answer:

```
sql1 <- "SELECT Ca.start_date, Co.name, Ca.name "
sql2 <- "FROM cabinet AS Ca "
sql3 <- "LEFT JOIN country AS Co ON Ca.country_id = Co.id "
sql4 <- "WHERE Ca.name LIKE '%Menzies%' "
sql5 <- "ORDER BY Ca.start_date ASC;"
sql <- paste0(sql1, sql2)
sql <- paste0(sql, sql3)
sql <- paste0(sql, sql4)
sql <- paste0(sql, sql5)

query <- dbSendQuery(parlcon, statement = sql)
qdata <- fetch(query, n = -1)
qdata

##    start_date      name        name
## 1  1939-04-26 Australia   Menzies I
## 2  1940-03-14 Australia  Menzies II
## 3  1940-10-28 Australia Menzies III
## 4  1949-12-19 Australia  Menzies IV
## 5  1951-05-11 Australia   Menzies V
## 6  1954-07-09 Australia  Menzies VI
## 7  1956-01-11 Australia Menzies VII
## 8  1958-12-10 Australia Menzies IIX
## 9  1961-12-22 Australia  Menzies IX
## 10 1963-12-18 Australia   Menzies X
```

(i) (3 Points) Which country had a cabinet lead by (Sanna) Marin? When? Just return the country name and the start_date. If you are uncertain whether you got the correct information, google this politician (and possibly look at his/her Wikipedia page to see whether the information you extracted from the data base is correct — but don't tell anyone that I suggested to look at Wikipedia ☺).

Answer:

```
sql1 <- "SELECT Ca.start_date, Co.name "
sql2 <- "FROM cabinet AS Ca "
sql3 <- "LEFT JOIN country AS Co ON Ca.country_id = Co.id "
sql4 <- "WHERE Ca.name LIKE '%Marin%' "
sql5 <- "ORDER BY Ca.start_date ASC;"
sql <- paste0(sql1, sql2)
sql <- paste0(sql, sql3)
sql <- paste0(sql, sql4)
sql <- paste0(sql, sql5)


query <- dbSendQuery(parlcon, statement = sql)
qdata <- fetch(query, n = -1)
qdata

##   start_date    name
## 1 2019-12-10 Finland
```

(j) (4 Points) Which cabinet immediately followed the final cabinet led by (Wilfried) Martens? Return the country name, the cabinet name, and the cabinet start_date.

You should use one single SQL query to answer this question. Do some google searches if necessary to learn how the result from one query can be further used in an extended query. There should be multiple ways to do so. Feel free to look at a Wikipedia page to see whether the information you extracted from the data base is correct.

You are **not** allowed to do any of the following: Directly use the name of the immediate successor (obtained from a Wikipedia or other search), directly enter a date, or directly enter the country name and then extract the k^{th} cabinet from this country.

You will lose some points if you need more than one SQL query, store intermediate results in some R data structure, or use information you found on the web in your queries.

Answer:

```
sql1 <- "SELECT Ca.start_date, Co.name, Ca.name "
sql2 <- "FROM cabinet AS Ca "
sql3 <- "LEFT JOIN country AS Co ON Ca.country_id = Co.id "
sql4 <- "WHERE Ca.start_date >"
sql5 <- "(SELECT MAX(Ca.start_date) "
sql6 <- "FROM cabinet AS Ca "
sql7 <- "LEFT JOIN country AS Co ON Ca.country_id = Co.id "
sql8 <- "WHERE Ca.name LIKE '%Martens%' LIMIT 1) "
sql9 <- "AND Co.name = (SELECT Co.name "
sql10 <- "FROM cabinet AS Ca "
sql11 <- "LEFT JOIN country AS Co ON Ca.country_id = Co.id "
sql12 <- "WHERE Ca.name LIKE '%Martens%' "
sql13 <- "ORDER BY Ca.start_date DESC LIMIT 1) "
sql14 <- "ORDER BY Ca.start_date ASC LIMIT 1;"
sql <- paste0(sql1, sql2)
sql <- paste0(sql, sql3)
sql <- paste0(sql, sql4)
sql <- paste0(sql, sql5)
sql <- paste0(sql, sql6)
sql <- paste0(sql, sql7)
sql <- paste0(sql, sql8)
sql <- paste0(sql, sql9)
sql <- paste0(sql, sql10)
sql <- paste0(sql, sql11)
sql <- paste0(sql, sql12)
sql <- paste0(sql, sql13)
sql <- paste0(sql, sql14)
```

```
query <- dbSendQuery(parlcon, statement = sql)
qdata <- fetch(query, n = -1)
qdata

##   start_date    name      name
## 1 1992-03-07 Belgium Dehaene I
```

(k) (1 Point) Don't forget to disconnect from the SQL data base.

Answer:

```
dbDisconnect(parlcon)
```

(ii) [redacted]:

In this question, you have to work with some photographs I have taken during visits to Taiwan, Russia, and Canada in the past few years.

Imagine this hypothetical question: Assume you have to explore an extensive photo collection (think of flickr or any other online photo archive). Typically, some photos contain names of places, streets, buildings, airports, museums, train/bus/metro stations, etc. You have to determine which places have been visited by a person (or a group of persons), based on the textual information shown in the photos.

For this question, you have to do the following: Work with the seven photos from the *Photos* directory I have provided in the zip folder called *HW03_Data.zip* in Canvas. First obtain the file names for all photos via the *list.files()* R function. Then, apply OCR to each photo. Break the text into individual lines, and transfer everything to lower case. Then extract the lines that contain *station* or *university* or *museum*. There should be seven such lines. You can ignore all other text in the photos.

Finally, using regular expressions, remove all single letters, double letters, and all other symbols that appear at the start or end of these lines. There may be more than just one letter that needs to be removed, e.g., remove A and E from "A BCD E", remove A from "A BCD", and remove A and B from "A B CDE", etc. You must use general regular expressions that will replace similar occurrences as well. So, you cannot write code to directly to replace "aw" or "zs" or "ah" or anything else you may encounter in these seven lines. Rather you have to replace two letters at the beginning or ending of a line, followed or preceded by a space (or something like that). As an example, you may get something like "to mrt nangang station a ae" from one of the photos. This has to be reported as "mrt nangang station" without explicitly stating to remove the "to" at the beginning and the "a ae" at the end.

Report the resulting station/university/museum names. How good is the OCR result? Can you further remove anything else in an automated way, or could that lead to some other problems (even if not shown in these seven photos)? Comment in a few sentences!

Also show your final R code.

The R code for this question can be very compact: I ended up with only 12 lines of (mostly piped) R code.

Note: You may get several warnings, but you hopefully will still get meaningful results! Warnings in R often can be ignored, but we always should check that the result is still meaningful.

Warning: Results are generally reproducible when run on the same computer. However, results may slightly differ based on the operating system being used (Windows vs. Mac), the version of the R packages being installed, and the local search paths being used on your computer. So, check that your results are meaningful in your final external pdf file and further adjust your R code if necessary and do not rely on results that were correct when the code was run on a different computer or in the RStudio Console earlier on.

Answer:

```
path.one <- "C:/Users/Scott/Downloads/HW03_Data/"
path.two <- "HW03_Data/photos/"
path <- paste0(path.one, path.two)
l.images <- list.files(path, full.names = TRUE)
library(tesseract)
library(magrittr)
ocr.list <- ocr(l.images)
image.one <- strsplit(ocr.list[1], "\n") %>%
             unlist() %>%
             gsub("(\\w)(\\w*)", "\\L\\1\\L\\2", ., perl=TRUE)
text.one <- grep("(station|university|museum)",
                 image.one, perl = TRUE, value = TRUE)

image.two <- strsplit(ocr.list[2], "\n") %>%
             unlist() %>%
             gsub("(\\w)(\\w*)", "\\L\\1\\L\\2", ., perl=TRUE)
text.two <- grep("(station|university|museum)",
                 image.two, perl = TRUE, value = TRUE)

image.three <- strsplit(ocr.list[3], "\n") %>%
               unlist() %>%
               gsub("(\\w)(\\w*)", "\\L\\1\\L\\2", ., perl=TRUE)
text.three <- grep("(station|university|museum)",
                   image.three, perl = TRUE, value = TRUE)

image.four <- strsplit(ocr.list[4], "\n") %>%
              unlist() %>%
              gsub("(\\w)(\\w*)", "\\L\\1\\L\\2", ., perl=TRUE)
text.four <- grep("(station|university|museum)",
                  image.four, perl = TRUE, value = TRUE)

image.five <- strsplit(ocr.list[5], "\n") %>%
              unlist() %>%
              gsub("(\\w)(\\w*)", "\\L\\1\\L\\2", ., perl=TRUE)
text.five <- grep("(station|university|museum)",
```

```
                  image.five, perl = TRUE, value = TRUE)

image.six <- strsplit(ocr.list[6], "\n") %>%
             unlist() %>%
             gsub("(\\w)(\\w*)", "\\L\\1\\L\\2", ., perl=TRUE)
text.six <- grep("(station|university|museum)",
                 image.six, perl = TRUE, value = TRUE)

image.seven <- strsplit(ocr.list[7], "\n") %>%
               unlist() %>%
               gsub("(\\w)(\\w*)", "\\L\\1\\L\\2", ., perl=TRUE)
text.seven <- grep("(station|university|museum)",
                   image.seven, perl = TRUE, value = TRUE)

text.one <- gsub("([a-z]*[^a-z ]+[a-z]*)", "",
                 text.one, perl = TRUE)
text.one <- gsub("  ", " ", text.one, perl = TRUE)
text.one <- gsub("^ ", "", text.one, perl = TRUE)
text.one <- gsub(" $", "", text.one, perl = TRUE)
text.one <- gsub("(^([a-z]{1,2} )*)", "",
                 text.one, perl = TRUE)
text.one <- gsub("(( [a-z]{1,2})*$)", "",
                 text.one, perl = TRUE)
text.one <- gsub("^ ", "", text.one, perl = TRUE)
text.one <- gsub(" $", "", text.one, perl = TRUE)


text.two <- gsub("([a-z]*[^a-z ]+[a-z]*)", "",
                 text.two, perl = TRUE)
text.two <- gsub("  ", " ", text.two, perl = TRUE)
text.two <- gsub("^ ", "", text.two, perl = TRUE)
text.two <- gsub(" $", "", text.two, perl = TRUE)
text.two <- gsub("(^([a-z]{1,2} )*)", "",
                 text.two, perl = TRUE)
text.two <- gsub("(( [a-z]{1,2})*$)", "",
                 text.two, perl = TRUE)


text.three <- gsub("([a-z]*[^a-z ]+[a-z]*)", "",
                   text.three, perl = TRUE)
text.three <- gsub("  ", " ", text.three, perl = TRUE)
text.three <- gsub("^ ", "", text.three, perl = TRUE)
text.three <- gsub(" $", "", text.three, perl = TRUE)
text.three <- gsub("(^([a-z]{1,2} )*)", "",
                   text.three, perl = TRUE)
text.three <- gsub("(( [a-z]{1,2})*$)", "",
                   text.three, perl = TRUE)


text.four <- gsub("([a-z]*[^a-z ]+[a-z]*)", "",
                  text.four, perl = TRUE)
text.four <- gsub("  ", " ", text.four, perl = TRUE)
text.four <- gsub("^ ", "", text.four, perl = TRUE)
text.four <- gsub(" $", "", text.four, perl = TRUE)
```

```
text.four <- gsub("(^([a-z]{1,2} )*)", "",
                  text.four, perl = TRUE)
text.four <- gsub("(( [a-z]{1,2})*$)", "",
                  text.four, perl = TRUE)


text.five <- gsub("([a-z]*[^a-z ]+[a-z]*)", "",
                  text.five, perl = TRUE)
text.five <- gsub("  ", " ", text.five, perl = TRUE)
text.five <- gsub("^ ", "", text.five, perl = TRUE)
text.five <- gsub(" $", "", text.five, perl = TRUE)
text.five <- gsub("(^([a-z]{1,2} )*)", "",
                  text.five, perl = TRUE)
text.five <- gsub("(( [a-z]{1,2})*$)", "",
                  text.five, perl = TRUE)


text.six <- gsub("([a-z]*[^a-z ]+[a-z]*)", "",
                 text.six, perl = TRUE)
text.six <- gsub("  ", " ", text.six, perl = TRUE)
text.six <- gsub("^ ", "", text.six, perl = TRUE)
text.six <- gsub(" $", "", text.six, perl = TRUE)
text.six <- gsub("(^([a-z]{1,2} )*)", "",
                 text.six, perl = TRUE)
text.six <- gsub("(( [a-z]{1,2})*$)", "",
                 text.six, perl = TRUE)


text.seven <- gsub("([a-z]*[^a-z ]+[a-z]*)", "",
                   text.seven, perl = TRUE)
text.seven <- gsub("  ", " ", text.seven, perl = TRUE)
text.seven <- gsub("^ ", "", text.seven, perl = TRUE)
text.seven <- gsub(" $", "", text.seven, perl = TRUE)
text.seven <- gsub("(^([a-z]{1,2} )*)", "",
                   text.seven, perl = TRUE)
text.seven <- gsub("(( [a-z]{1,2})*$)", "",
                   text.seven, perl = TRUE)

text.one
```

```
## [1] "the russian museum"
```

```
text.two
```

```
## [1] "ubc museum of anthropology"
```

```
text.three
```

```
## [1] "taipel nangang exhibition center station"
```

```
text.four
```

```
## [1] "hsr tainan station"
```

```
text.five
```

```
## [1] "national cheng kung university"
```

```
text.six
```

```
## [1] "taipei zoo station"
```

```
text.seven
```

```
## [1] "mrt nangang station"
```

Comments:
Report the resulting station/university/museum names. How good is the OCR result?

The OCR result is pretty accurate after I clean through regular expressions in the prescribed manner. The only mistake is in image three where the ocr functions misspells taipei as "taipel".

Can you further remove anything else in an automated way, or could that lead to some other problems (even if not shown in these seven photos)?

After removing text in the prescribed manner, I feel that I may have removed too much as it leaves out, for example "Department of Statistics" in the fifth image because it is on a separate line. Therefore, I am against removing anything else in an automated way, although one certainly could by like deleting all words that are 3 or less letters long at the start and end of each line (as opposed to doing it for 2 or less letters long words which was prescribed).

(iii) [redacted]

In this question, you have to work with information regarding the Panopto recordings from previous semesters. This information is stored by default in XML format. Panopto describes at `https://www.panopto.com/features/integration/universal-capture-specification/` why they are using such a *Universal Capture Specification* (UCL) format for their audio and video files. There is no need to request the full specifications of their UCL format to answer this HW question.

Work with the 64 XML files I have provided in the zip folder called *HW03_Data.zip* (there is a *XML* subdirectory in the zip file) in Canvas. First obtain the file names for all XML files via the *list.files()* R function. Then extract the RecordingStartTime and RecordingEndTime from each XML file. Calculate the time difference (TimeDiff) as RecordingEndTime − RecordingStartTime (in minutes). Apparently, TimeDiff represents the length (in minutes) of a recording.

Create a data frame that consists of these 3 columns and show the first 6 lines and the last 6 lines of this data frame after **sorting** the data frame according to the RecordingStartTime.

Create two graphs of the time differences: (i) a basic histogram of the recording lengths (use intervals that are 2.5 min wide); and (ii) a scatterplot that shows the date (and time) on the horizontal axis and the recording lengths on the vertical axis.

Summarize the main features of both graphs and describe their characteristics. Are there any outliers? If so, why do you think so? When did they occur? This summary should be between 1/2 and 1 page in length.

Also show your final R code.

Note:

(a) These data were collected over a few semesters. Some courses were taught on a Monday, Wednesday, and Friday. Others were taught on a Tuesday and Thursday. Thus, the time differences should be about 50 min and 75 min. Are they ?!? If you are interested in what I taught those semesters, take a look at my teaching web page at `https://math.usu.edu/~symanzik/teaching/JSteaching.html`.

(b) There are many ways to make mistakes in this question. First, if you do not set the correct XML path, you likely will get NULL, NA, or an empty data frame.

(c) Next, time difference calculations can also go totally wrong. Keep Note (a) from above in mind. If something is totally off, look at the underlying XML file (recall that XML files are human–readable). If a rough manual calculation of a time difference is rather different from the result obtained in R, apparently something is wrong in your R code.

(d) All timestamps and durations in Panopto are in ISO 8601 format. See here for further details on this format: `https://www.iso.org/iso-8601-date-and-time-format.html` and `http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a003169814.htm`.

(e) There exist several R packages such as *anytime* and *timeDate* that provide functions to manipulate different date and time formats, including the ISO 8601 format. Time functions also exist in baseR. You have to figure out yourself which of these functions provide the functionality to transform the recorded XML time format and calculate the time difference in minutes. If something does not make sense, check (and modify) the default settings or use a different time function with different default settings.

Answer:

```
library(XML)
library(anytime)
library(dplyr)


##
## Attaching package:  'dplyr'

## The following objects are masked from 'package:stats':
##
##    filter, lag

## The following objects are masked from 'package:base':
##
##    intersect, setdiff, setequal, union


path.one <- "C:/Users/Scott/Downloads/HW03_Data/"
path.two <- "XML/"
path <- paste0(path.one, path.two)
l.xml <- list.files(path, full.names = TRUE)
R.S.T.l <- c()
R.E.T.l <- c()
R.S.T <- "RecordingStartTime"
R.E.T <- "RecordingEndTime"
for (i in l.xml) {
  doc <- xmlParse(i)
  root <- xmlRoot(doc)
  R.S.T.l <- append(R.S.T.l, xmlValue(root[[R.S.T]]))
```

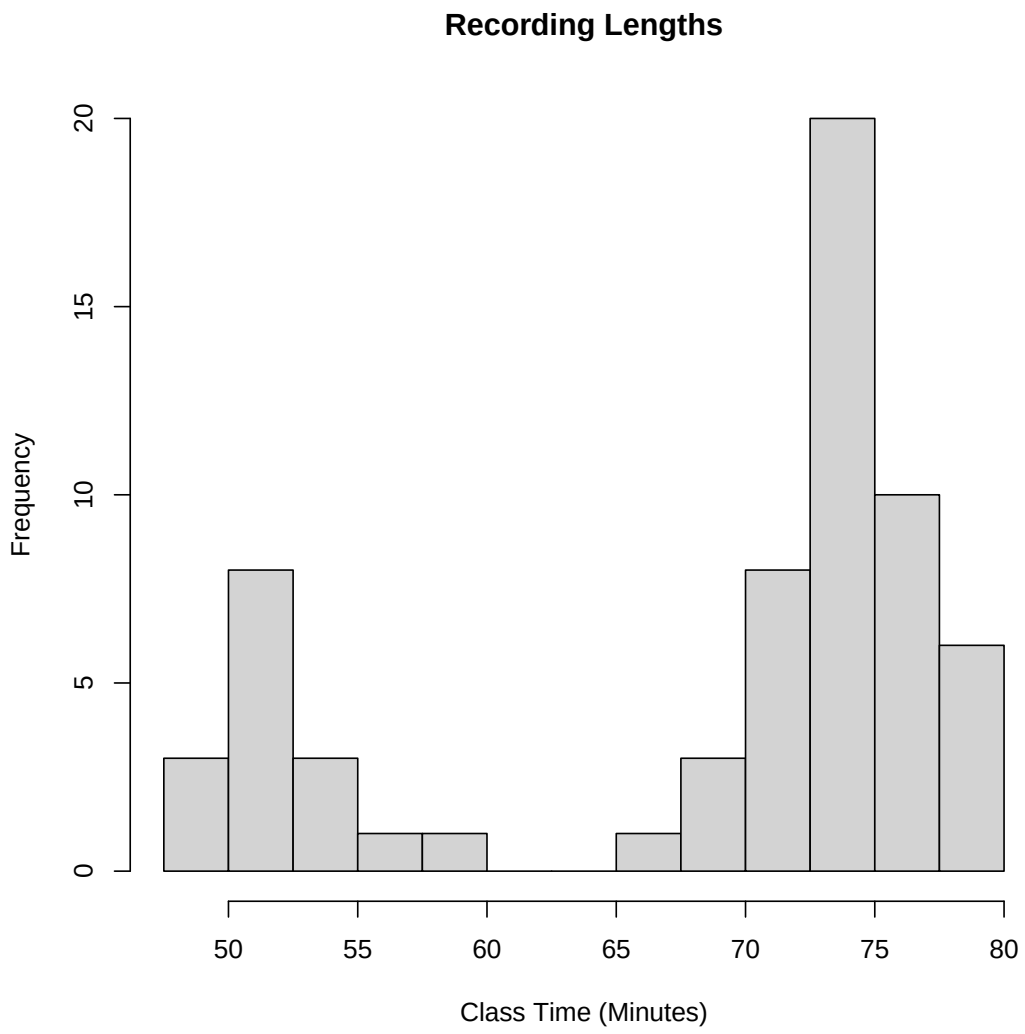
```
  R.E.T.l <- append(R.E.T.l, xmlValue(root[[R.E.T]]))
}
R.S.T.l.a <- anytime(R.S.T.l)
R.E.T.l.a <- anytime(R.E.T.l)

p.times <- as.data.frame(list(R.S.T.l.a,
                              R.E.T.l.a,
                              R.E.T.l.a - R.S.T.l.a),
                         col.names = c("RecordingStartTime",
                                       "RecordingEndTime",
                                       "TimeDiff"))
head(arrange(p.times, RecordingStartTime), 6)
```

```
##    RecordingStartTime    RecordingEndTime      TimeDiff
## 1 2018-09-13 11:02:01 2018-09-13 12:15:17 73.26647 mins
## 2 2018-09-13 13:59:31 2018-09-13 15:16:00 76.48014 mins
## 3 2018-09-18 11:01:56 2018-09-18 12:16:10 74.24463 mins
## 4 2018-09-18 13:58:45 2018-09-18 15:17:51 79.08844 mins
## 5 2018-09-20 11:01:41 2018-09-20 12:15:10 73.48719 mins
## 6 2018-09-20 14:00:11 2018-09-20 15:14:41 74.50409 mins
```

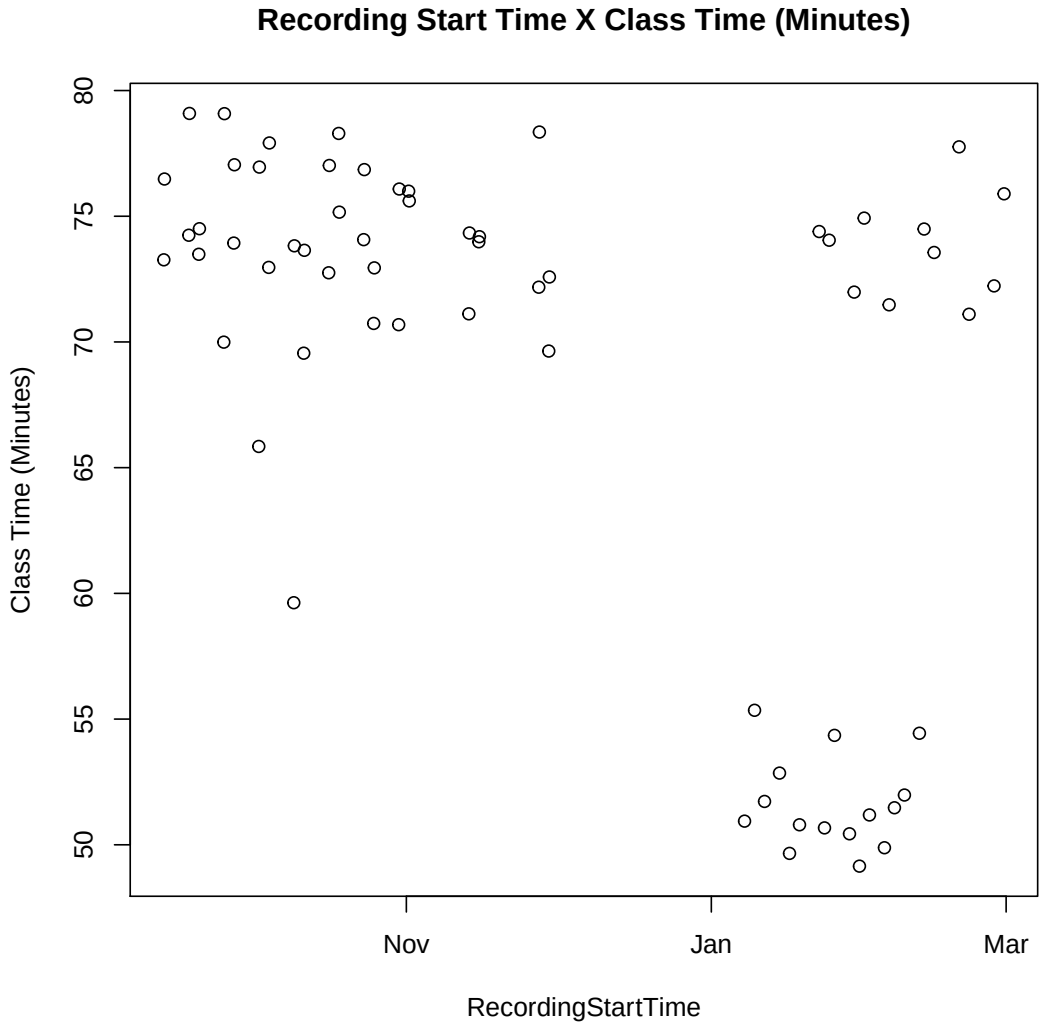
```
tail(arrange(p.times, RecordingStartTime), 6)
```

```
##     RecordingStartTime    RecordingEndTime      TimeDiff
## 59 2019-02-12 14:01:59 2019-02-12 15:16:28 74.49116 mins
## 60 2019-02-14 14:02:41 2019-02-14 15:16:15 73.55843 mins
## 61 2019-02-19 14:01:14 2019-02-19 15:18:59 77.75915 mins
## 62 2019-02-21 14:02:45 2019-02-21 15:13:51 71.10220 mins
## 63 2019-02-26 14:03:22 2019-02-26 15:15:36 72.22920 mins
## 64 2019-02-28 14:02:12 2019-02-28 15:18:06 75.89176 mins
```

```
hist(sapply(p.times["TimeDiff"],
            function(x) as.numeric(x)),
     breaks = 2.5*19:32, main = "Recording Lengths",
     xlab = "Class Time (Minutes)")
```




```
plot(x = p.times[R.S.T],
     y = sapply(p.times["TimeDiff"],
                function(x) as.numeric(x)),
     ylab = "Class Time (Minutes)", xlab = R.S.T,
     main = "Recording Start Time X Class Time (Minutes)")
```



Comments:

Summarize the main features of both graphs and describe their characteristics.

For the histogram, we see that the that the histogram appears to be bimodal, and we see the range of class times (in minutes) goes from just below 50 to just above 80.

For the scatterplot, we see that from October to December we see the range of class times to mainly be just below 70 (minutes) to just above 80 with two main exceptions of about 65 and 60. Then from January to March we see a class time range of about 50 to 55 and a class time range of 70 to 80 (indicating their may be 2 classes from January to March that have different listed class times).

Are there any outliers? If so, why do you think so? When did they occur?

I do not believe there are any outliers when looking at class times as a whole. Maybe if we look at just between October and December we may see a couple of outliers of about 65 (minutes) and 60.

General Instructions

(i) Create a single pdf document, using R Markdown, Sweave, or knitr. When you take this course at the 6000–level, you have to use LaTeX in combination with Sweave or knitr. You only have to submit this one document to Canvas.

(ii) Include a title page that contains your name, your A–number, the number of the assignment, the submission date, and any other relevant information.

(iii) Start your answers to each main question on a new page (continuing with the next part of a question on the same page is fine). Clearly label each question and question part. Your answer to question (i) should start on page 2!

(iv) Show your R code and resulting graph(s) [if any] for each question part!

(v) Before you submit your homework, check that you follow all recommendations from Google's R Style Guide (see `http://web.stanford.edu/class/cs109l/unrestricted/resources/google-style.html`). Moreover, make sure that your R code is consistent, i.e., that you use the same type of assignments and the same type of quotes throughout your entire homework.

(vi) Give credit to external sources, such as stackoverflow or help pages. Be specific and include the full URL where you found the help (or from which help page you got the information). Consider R code from such sources as "legacy code or third–party code" that does not have to be adjusted to Google's R Style (even though it would be nice, in particular if you only used a brief code segment).

(vii) **Not following the general instructions outlined above will result in point deductions!**

(viii) For general questions related to this homework, please use the corresponding discussion board in Canvas! I will try to reply as quickly as possible. Moreover, if one of you knows an answer, please post it. It is fine to refer to web pages and R commands, but do not provide the exact R command with all required arguments or which of the suggestions from a stackoverflow web page eventually worked for you! This will be the task for each individual student!

(ix) Submit your single pdf file via Canvas by the submission deadline. Late submissions will result in point deductions as outlined on the syllabus.

1 Citations and References

- No references