

Homework Assignment 3 (11/12/2020)

Niranjana Poudel (A02304550)

Submitted on (12/14/2020)

65 Points — Due Monday 12/14/2020 (via Canvas by 11:59pm)

(i) (25 Points) **Parliaments and Governments SQL Data Base:**

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:

```
# loading the required packages
library(RSQLite)
```

Answer:

```
# Defining the database driver
driver <- dbDriver("SQLite")

# Establishing a connection to SQL database
loc1 <- "C:\\Users\\Niranjan.p\\Box\\Home work and assignments\\"
loc2 <- "Data technologies\\HW3\\HW03_Data\\HW03_Data\\Hw3\\"
file <- "parlgov-development.db"

# Defining a connection to the database file
parlcon <- dbConnect(driver, dbname = paste0(loc1, loc2, file))
```

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

Answer:

```
# Tables in the database
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:

```
# Loading first 5 countries
sql <- "SELECT * FROM country LIMIT 5;"

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:

```
# Loading all country name and abbreviations with sorting
sql <- "SELECT name, name_short FROM country ORDER BY name;"

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:

```
# Earliest and latest time span from election table
sql <- "SELECT MIN(date), MAX(date) FROM election;"

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

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

- (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:

```
# Early election in Sweden, Latvia, Austria and
# United Kingdom
sql <- "SELECT E.country_id, E.date, C.name AS country_name, E.early
FROM country AS C, election AS E
WHERE E.country_id = C.id AND
      E.early = 1 AND
      (C.name = 'Sweden' OR
       C.name = 'Latvia' OR
       C.name = 'Austria' OR
       C.name = 'United Kingdom')
ORDER BY country_name, date;"
```

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

##	country_id	date	country_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:

```
# Getting the most recent early election of United Kingdom
sql <- "SELECT E.country_id, MAX(E.date) AS date, C.name, E.wikipedia
      FROM country AS C, election AS E
      WHERE E.country_id = C.id AND
            E.early = 1 AND
            C.name = 'United Kingdom';"

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

##   country_id      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:

```
# Getting country with cabinets lead by Menzies.
sql <- "SELECT C.name AS country_name, Ca.name AS cabinet_name,
        Ca.start_date
        FROM country AS C, cabinet AS Ca
        WHERE C.id = Ca.country_id AND
              Ca.name LIKE '%Menzies%'
        ORDER BY Ca.start_date;"
```

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

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

- (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:

```
# Getting the cabinet lead by Marin
sql <- "SELECT C.name AS country_name, Ca.start_date
      FROM country AS C, cabinet AS Ca
      WHERE C.id = Ca.country_id AND
            Ca.name LIKE '%Marin%'
      ORDER BY Ca.start_date;"

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

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

- (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:

```
# Getting the cabinet immediately followed by cabinet lead by Martens.
sql <- "SELECT C.name AS country_name, Ca.name AS cabinet_name,
        Min(Ca.start_date) AS start_date
FROM country AS C, cabinet AS Ca
WHERE c.id = Ca.country_id AND
      Ca.start_date > (SELECT MAX(Ca.start_date)
                      FROM country AS C, cabinet as Ca
                      WHERE C.id = Ca.country_id AND
                            Ca.name LIKE '%Martens%' );"

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

##   country_name  cabinet_name start_date
## 1 Switzerland Bundesrat 1991 1991-12-04
```

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

Answer:

```
# Disconnecting the SQL database
dbDisconnect(parlcon)
```

(ii) (20 Points) **OCR from Photo Files:**

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:

```
# Loading the required packages required
library(tesseract)
library(OpenImageR)
library(tidyverse)
```

Answer:

```
# Extracting the text from images
list.files("Photos") %>%
  paste0("Photos/", .) %>%
  lapply(ocr) %>%
  toString() %>%
  strsplit("\n") %>%
  lapply(tolower) %>%
  unlist() %>%
  str_extract(".*(station|university|museum).*") %>%
  .[!is.na(.)] %>%
  str_extract(".*\\b\\w{3,}\\b") %>%
  str_extract("\\b\\w{3,}\\b.*")

## [1] "the russian museum"
## [2] "ubc museum of anthropology §6393nwm"
## [3] "taipel nangang exhibition center station"
## [4] "hsr tainan station"
## [5] "national cheng kung university"
## [6] "taipei zoo station"
## [7] "mrt nangang station"
```

Comments:

We can see the name of the resulting station/university/museum from above answer. Looking at the photographs and the extracted final names i must say the result is pretty decent at least for this homework. The result for this homework

(for my case) can be made better by removing the word having digits in it where we can see part of the address still is in the final answer (2nd one). Again outside of these seven photos sometime there may be digits in the text we do not want to remove, so if we move outside of these seven photos we might have to rethink regexp or coding pattern depending upon what we want to extract, for an example using the similar code we might remove "An" which is two letters and usually comes in beginning which might sometime be a part of a name for something.

(iii) (20 Points) **XML Files of Panopto Classroom Recordings:**

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:

```
# Loading the required packages
library(XML)
library(xml2)
library(anytime)
library(ggplot2)
```

Answer:

```
list.files("XML") %>%
  paste0("XML/", .) %>%
  lapply(xmlParse) %>%
  lapply(xmlRoot) %>%
  lapply(function(x) getChildrenStrings(x)[c("RecordingEndTime",
                                              "RecordingStartTime")]) %>%

  do.call("rbind", .) %>%
  as.data.frame() %>%
  mutate(RecordingEndTime = anytime(RecordingEndTime)) %>%
  mutate(RecordingStartTime = anytime(RecordingStartTime)) %>%
  mutate(TimeDiff = difftime(RecordingEndTime, RecordingStartTime)) %>%
  arrange(RecordingStartTime) ->
  XML

head(XML)

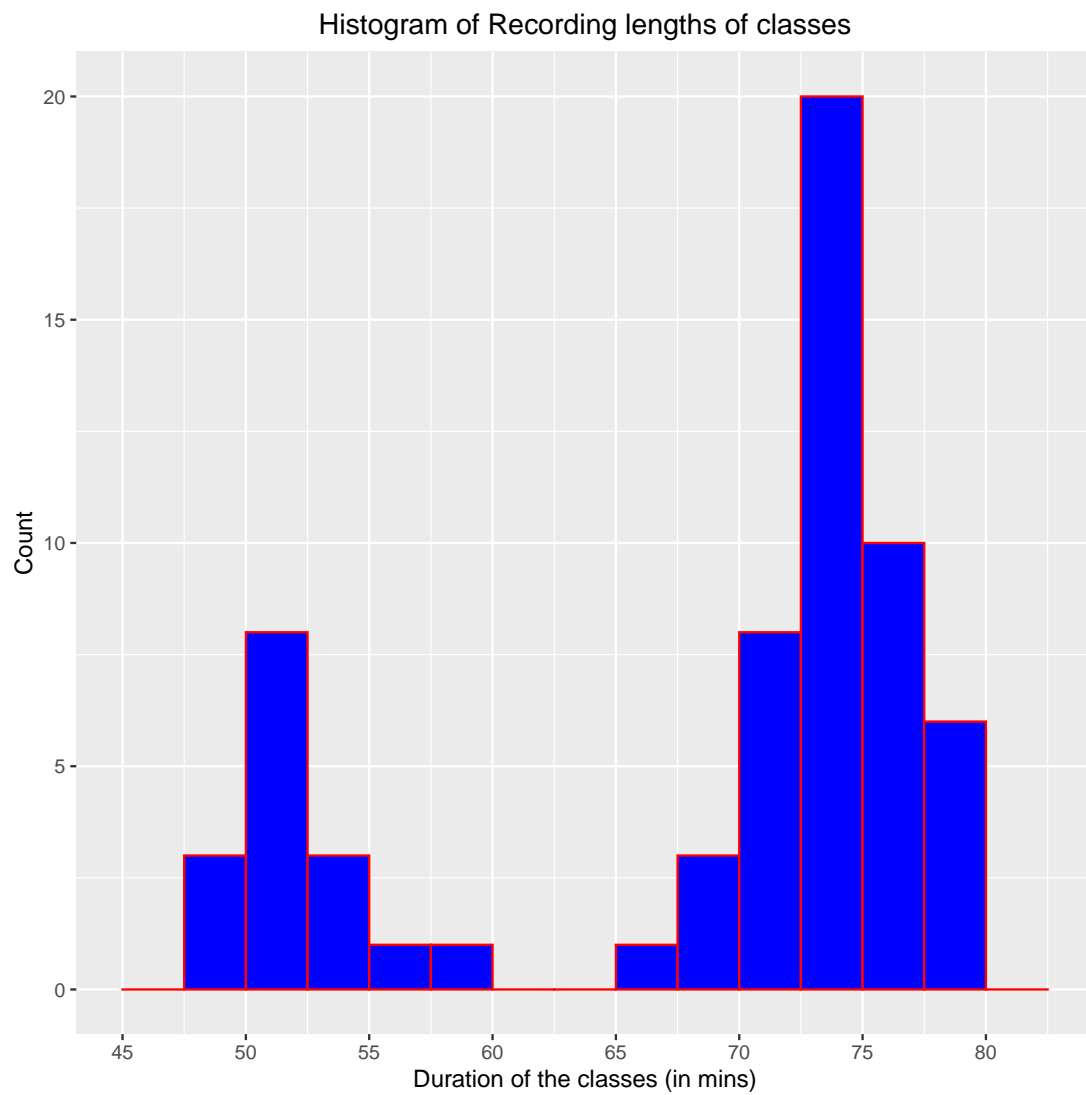
##      RecordingEndTime RecordingStartTime      TimeDiff
## 1 2018-09-13 12:15:17 2018-09-13 11:02:01 73.26647 mins
```

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

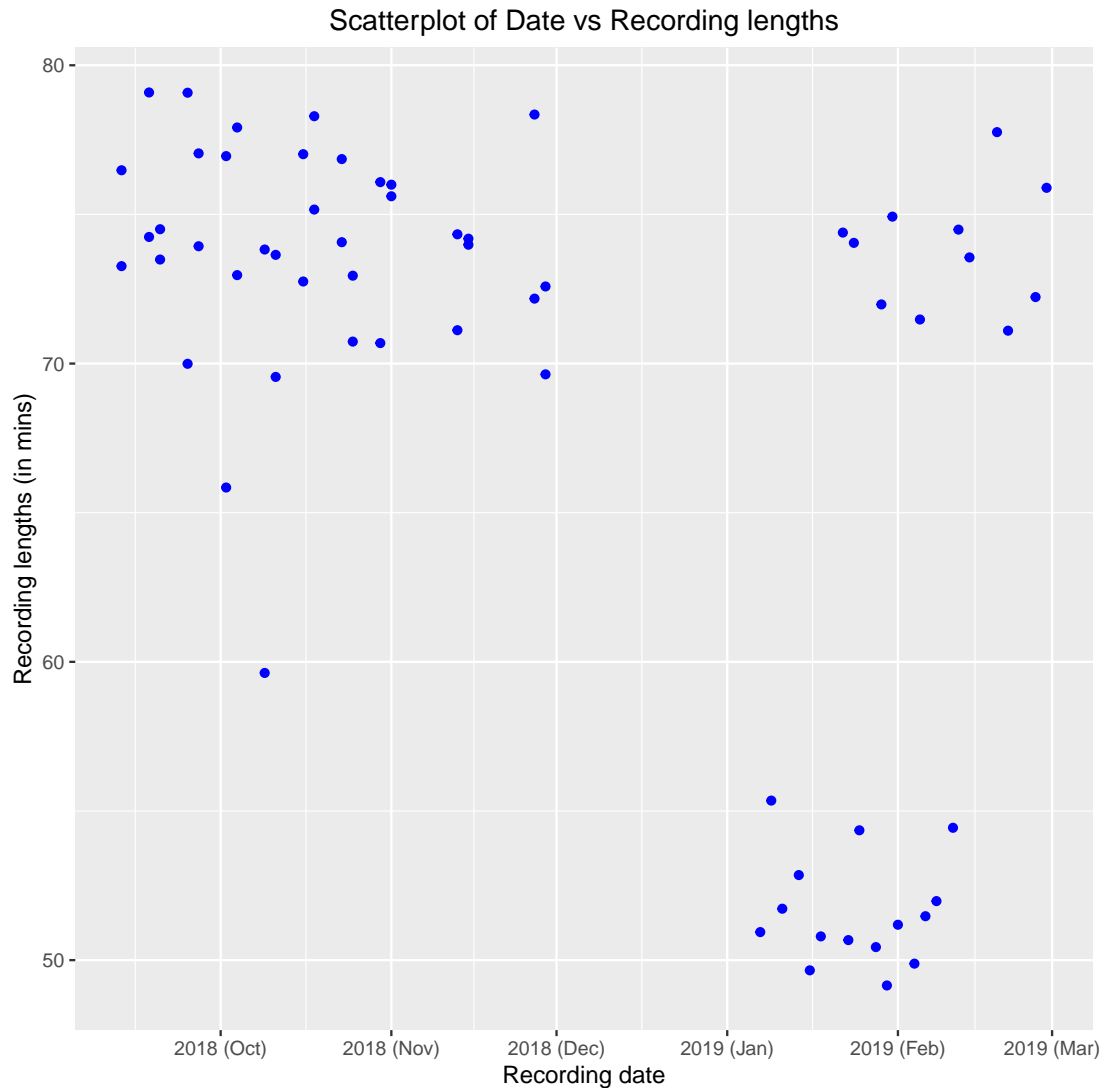
tail(XML)

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

ggplot(data = XML, aes(as.numeric(TimeDiff))) +
  geom_histogram(breaks = seq(45, 83, by = 2.5),
    col = "red", fill = "blue") +
  ggtitle("Histogram of Recording lengths of classes") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Duration of the classes (in mins)") +
  ylab("Count") +
  scale_x_continuous(breaks = seq(45, 83, by = 5))
```



```
ggplot(data = XML, aes(x = as.Date(RecordingStartTime),
                        y = as.numeric(TimeDiff))) +
  geom_point(col = "blue") +
  ggtitle("Scatterplot of Date vs Recording lengths") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Recording date") +
  ylab("Recording lengths (in mins)") +
  scale_x_date(date_labels = "%Y (%b)")
```



Comments:

The two plots of histogram and scatter plot above speaks a lot about the recordings/classes. First of all if we look at the histogram we can see a bi-modal distribution and two separated distribution by some distance which is already explained in the question as well (this histogram was as expected), as one set of class were of length 50 minutes and another set of classes were 75 minutes. One interesting thing we can notice from the histogram is for 50 minutes class the modal duration of class is 50 minutes to 52.5 minutes, and for the longer classes the modal distribution is 72.5 minutes to 75 minutes. So, it might give us the feeling its difficult to wrap of things in shorter class and there might be enough time in longer class. Also for the classes with 50 minutes there are some observations from 47.5

minutes to 50 minutes and no classes have fewer duration than them but but for 75 minutes classes there are some classes between 65 minutes to 67.5 minutes and even one class below 60 minutes which is seen in scatter plot but not in histogram.

Next if we look at the Scatter plot, one outlier is easily visible in the first cluster (the left side of graph) where the recording time is less than 60 which occurred sometime in October of 2018. There can be many reasons for why this class was shorter than 60 minutes (my guess project discussion was not recorded). We can see the school break or no classes at least from December to some part of January. You were teaching two classes in spring of 2019 one 50 minute course and one 75 minute course. Again the claim from histogram that 50 minutes course are generally longer than 50 minutes and 75 minutes course are shorter than 75 minutes can also be seen from the scatter plot. Also some we can see some of the classes starting at the middle of the spring semester after one other course has been already started. The 50 minutes course also seems to be finished somewhere in February. A more detailed time-line plot can be made of everyday if we want to break and understand more but I have not done that for this Homework.

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 L^AT_EX 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/cs1091/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.