# STAT 4410/8416 Homework 5

*Danuwar Ramesh*

*Due on Nov 28, 2020*

1. **Working with databases:** Please follow the instruction below before answering the questions:

- Install the package `sqldf` using `install.packages('sqldf')`
- Import the library using `library('sqldf')`
- Read the file `https://raw.githubusercontent.com/dsindy/kaggle-titanic/master/data/train.csv` and store it in `titanic`

We can now start writing SQL Script using `SQLDF library` right inside R. See example below:

```
library(sqldf)

sqldf("SELECT passengerid, name, sex
      FROM titanic
      limit 5", drv="SQLite")
```

```
##   PassengerId                                                Name    Sex
## 1           1                             Braund, Mr. Owen Harris   male
## 2           2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female
## 3           3                              Heikkinen, Miss. Laina female
## 4           4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female
## 5           5                            Allen, Mr. William Henry   male
```

Answer the following questions. Write SQL Script where applicable.

a. What is a Database Management System? Name few types of Database Management System. Answer: Database management system is the system that allows to store, modify, and extract information from a database.The different types of databasae management system are: mysql, oracle, ms sql, db2, ms access etc

b. Provide 3 verbs of SQL? Please write what they do Answer: Select:The SELECT statement is used to select data from a database. Update:The SQL UPDATE Query is used to modify the existing records in a table Delete:To remove one or more rows from a table completely, you use the DELETE statement INSERT:used to add new rows of data to a table in the database

c. What does the following command do in MySQL?

   i. ‘show databases; Answer: To display the available databases the command is show.
   ii. `show tables;` Answer:It helps to list the tables for the current/specified database or schema or accross the entire account.

d. Write `SQL` script to answer the following questions based on titanic data. Display the results of your script.

   iii. What is the average age of passengers who survived? Group the data by Sex. Display only the column `Sex`, `AverageAge`

```
kable(sqldf("SELECT Sex,
                    avg(Age) as AverageAge
      FROM titanic
      WHERE Survived = 1
      group by Sex", drv="SQLite"))
```

| Sex | AverageAge |
|---|---:|
| female | 28.84772 |
| male | 27.27602 |

```
iv. What is the percentage of passengers who survived in each Passenger Class or `Pclass`? Group the da
```

```
kable(sqldf("SELECT count(Survived) as total_rows_survived
      FROM titanic",
            drv="SQLite"))
```

| total_rows_survived |
|---:|
| 891 |

```
sqldf("SELECT pclass,
      sex, cast(sum(survived) as real) / count(survived) * 100 as avg_survived_percent
      FROM titanic
      GROUP BY pclass, sex;"
      , drv="SQLite")
```

```
##   Pclass    Sex avg_survived_percent
## 1      1 female             96.80851
## 2      1   male             36.88525
## 3      2 female             92.10526
## 4      2   male             15.74074
## 5      3 female             50.00000
## 6      3   male             13.54467
```

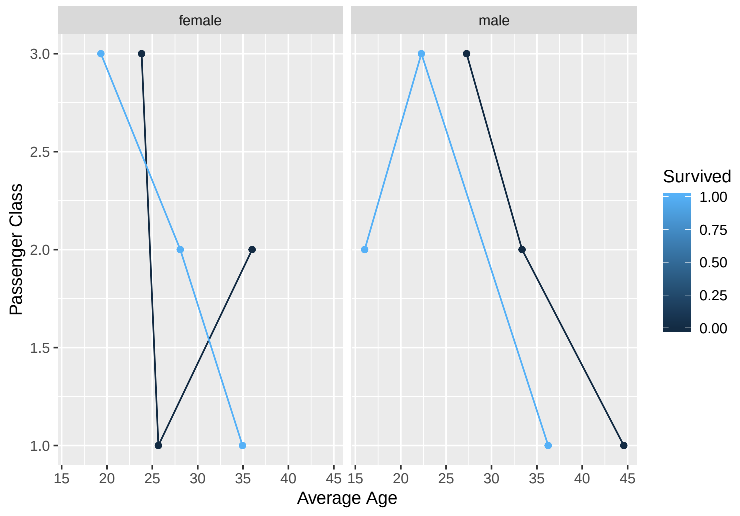
```
v. What is the average age of all the passenger (survived and not survived)? Group the data by `Pclass`
```

```
q1_iii <- sqldf("SELECT Pclass,
                        Sex,
                        Survived,
                        avg(Age) as AverageAge
      FROM titanic
      group by Pclass, Sex, Survived", drv="SQLite")

kable(q1_iii)
```

| Pclass | Sex | Survived | AverageAge |
|---:|---|---:|---:|
| 1 | female | 0 | 25.66667 |
| 1 | female | 1 | 34.93902 |

| Pclass | Sex | Survived | AverageAge |
|---|---|---|---|
| 1 | male | 0 | 44.58197 |
| 1 | male | 1 | 36.24800 |
| 2 | female | 0 | 36.00000 |
| 2 | female | 1 | 28.08088 |
| 2 | male | 0 | 33.36905 |
| 2 | male | 1 | 16.02200 |
| 3 | female | 0 | 23.81818 |
| 3 | female | 1 | 19.32979 |
| 3 | male | 0 | 27.25581 |
| 3 | male | 1 | 22.27421 |

```r
library(ggplot2)
ggplot(data=q1_iii, aes(x=AverageAge, y=Pclass, color=Survived, group=Survived))+
  geom_line()+
  geom_point()+
  facet_grid(~Sex)+
  xlab("Average Age")+
  ylab("Passenger Class")+
  labs(color='Survived')
```



vi. What is the name, age, sex and pclass of the 5 oldest and 5 youngest persons who died?

```r
# 5 Youngest person who died
kable(sqldf(" SELECT Name, Age, Sex, Pclass
        FROM titanic
        WHERE Age IS NOT NULL OR age <= 0
        AND Survived = 0
        ORDER BY Age ASC
        LIMIT 5", drv="SQLite"))
```

| Name | Age | Sex | Pclass |
|---|---|---|---|
| Thomas, Master. Assad Alexander | 0.42 | male | 3 |
| Hamalainen, Master. Viljo | 0.67 | male | 2 |
| Baclini, Miss. Helene Barbara | 0.75 | female | 3 |
| Baclini, Miss. Eugenie | 0.75 | female | 3 |
| Caldwell, Master. Alden Gates | 0.83 | male | 2 |

```
# 5 Oldest person who died
kable(sqldf(" SELECT name, age, sex, pclass
        FROM titanic
        WHERE age IS NOT NULL OR age <= 0
        AND Survived = 0
        ORDER BY age DESC
        LIMIT 5", drv="SQLite"))
```

| Name | Age | Sex | Pclass |
|---|---|---|---|
| Barkworth, Mr. Algernon Henry Wilson | 80.0 | male | 1 |
| Svensson, Mr. Johan | 74.0 | male | 3 |
| Goldschmidt, Mr. George B | 71.0 | male | 1 |
| Artagaveytia, Mr. Ramon | 71.0 | male | 1 |
| Connors, Mr. Patrick | 70.5 | male | 3 |

```
vii. On average which Passenger Class is more expensive?
```

```
kable(sqldf(" SELECT Pclass, avg(Fare) as AverageFare
        FROM titanic
        GROUP BY Pclass", drv="SQLite"))
```

| Pclass | AverageFare |
|---|---|
| 1 | 84.15469 |
| 2 | 20.66218 |
| 3 | 13.67555 |

2. **Extracting twitter data;** In this problem we would like to extract data from twitter. For this refer to the documentation in the following link.

`https://github.com/geoffjentry/twitteR/`

a. **Twitter API set up** Set up twitter API using any of the following methods. Make sure you installed all the packages as mentioned in the class.

**Method 1:** Read Getting Started section of the above link and create a twitter application by going to the link `https://apps.twitter.com/`. Once you created your application connect twitter from `R` using the secrets and keys obtained from your twitter application.

```
library(twitteR)
api_key <- "your api key"
```

```r
api_secret <- "your api secret"
access_token <- "your access tocken"
access_token_secret <- "your access tocken secret"

setup_twitter_oauth(api_key,api_secret,access_token,
                    access_token_secret)
```

**Method 2:** If you don't like creating an account with twitter and going through all the trouble, you can use my keys (ssh, don't tell anyone). For this download the `hw5-twitter-auth` file from blackboard and load it as follows.

```r
load("hw5-twitter-auth")
library(base64enc)
library(httr)
library(twitteR)
setup_twitter_oauth(api_key,api_secret,access_token,
                    access_token_secret)
```

```
## [1] "Using direct authentication"
```

b. Now search twitter messages for "data science job". Display few job informations.

```r
datascience_jobs_tweets <- twitteR::searchTwitter("data science job")
head(datascience_jobs_tweets)
```

```
## [[1]]
## [1] "DJKatie888: RT @TennConserv: ...this was never about science or data; it's about control and pla
##
## [[2]]
## [1] "epuujee: RT @tmj_inh_it: Nervous to apply for a job like \"HCaaS- Data Science- Research and Se
##
## [[3]]
## [1] "tmj_inh_it: Nervous to apply for a job like \"HCaaS- Data Science- Research and Sensing- Hydera
##
## [[4]]
## [1] "Sad_Little_King: RT @TennConserv: ...this was never about science or data; it's about control a
##
## [[5]]
## [1] "hbbtruth: RT @TennConserv: ...this was never about science or data; it's about control and plac
##
## [[6]]
## [1] "mrjuoss: RT @freeCodeCamp: Spreadsheet software like Microsoft Excel is used in office work, da
```

c. Search 300 tweets using the hash tag `#chess` and save them in an object called `rTweets`. Show the top 7 sources of tweets (such as android or iphone) in a ordered bar plot.

```r
# Search 300 tweets with (#chees)
rTweets <- twitteR::searchTwitter("#chess", n=300)

# Get the source from where the tweet came from
tweet_sources_raw <- sapply(rTweets, function(x) x$getStatusSource())
```

```r
# Remove html tags that is on the sources and only get the value between the tags
# This is the source
library(stringr)
regex_pattern <- '[^">]+\\</a>'
tweet_sources_text <- gsub("</a>","", unlist(str_extract(tweet_sources_raw, regex_pattern)))

# Display few sources
head(tweet_sources_text)
```

```
## [1] "Cheap Bots, Done Quick!" "Instagram"
## [3] "Merchant Media Bot"      "Twitter Web App"
## [5] "TweetDeck"               "The Tweeted Times"
```

```r
# We now have the source from which the 300 tweets were being sent
# Now lets get the frequency count of the sources of distict 300 tweets
tweet_sources_text_FreqDist <- table(tweet_sources_text)

# Display few tweets with Frequenct Distribution
head(tweet_sources_text_FreqDist)
```

```
## tweet_sources_text
##     @100DaysOfCode_ 1 2 Chess! Tweetbot         3dRenderBot
##                   1                   10                   1
##      AdeptLibrarium         BGA Curator            BirdSite
##                   1                    1                   1
```

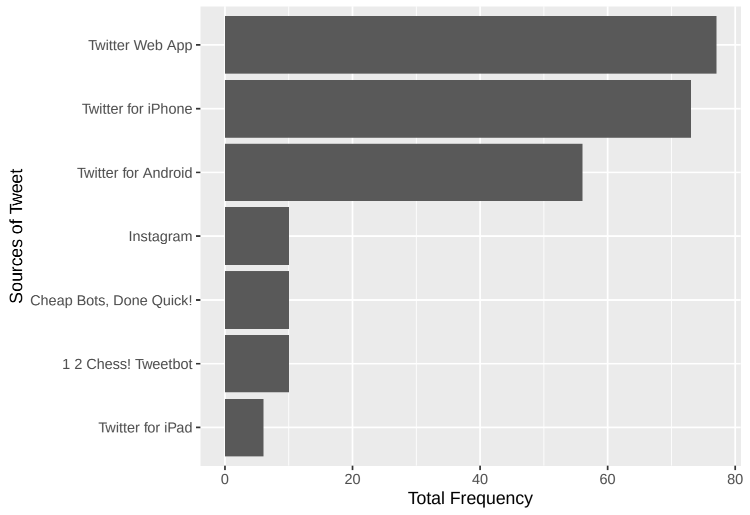
```r
library(ggplot2)
# Convert the Frequency Distirbution data into ta nice Data Frame
# Also sort the frequency data in decreasing order at the same time
library(dplyr)
rTweetsDF_sources <- data.frame(tweet_sources_text_FreqDist) %>% arrange(desc(Freq))

# Display few data
kable(head(rTweetsDF_sources))
```

| tweet_sources_text | Freq |
|---|---|
| Twitter Web App | 77 |
| Twitter for iPhone | 73 |
| Twitter for Android | 56 |
| 1 2 Chess! Tweetbot | 10 |
| Cheap Bots, Done Quick! | 10 |
| Instagram | 10 |

```r
# We only want the top 7 data
# Now lets subset the data
rTweetsDF_sources_top7 <- head(rTweetsDF_sources, 7)

# Display top 7 data
rTweetsDF_sources_top7
```

```
##        tweet_sources_text Freq
## 1          Twitter Web App   77
## 2       Twitter for iPhone   73
## 3      Twitter for Android   56
## 4      1 2 Chess! Tweetbot   10
## 5 Cheap Bots, Done Quick!   10
## 6                Instagram   10
## 7          Twitter for iPad    6
```

```
# Use ggplot to plot the ordered bar plot
ggplot(rTweetsDF_sources_top7, aes(x=reorder(tweet_sources_text, Freq), y =Freq ))+
  geom_bar(stat="identity")+
  xlab("Sources of Tweet")+
  ylab("Total Frequency")+
  coord_flip()
```



d. Notice that the object `rTweets` is a list. Convert it into a data frame using function `twListToDF` and store it in an object called `dTweets`. Display some data from `dTweets`.

```
dTweets <- twListToDF(rTweets)
head(dTweets)
```

```
##
## 1 Chessbot  Results: Black to win in 4 turns\n\u2b1b\u2b1c\u2b1b\u2b1c   \u2b1c\u2b1b   \n   \u2b1b\u2
## 2
## 3
## 4
## 5
## 6
##   favorited favoriteCount replyToSN             created truncated
## 1     FALSE             0      <NA> 2020-11-29 04:29:01      TRUE
## 2     FALSE             1      <NA> 2020-11-29 04:21:23      TRUE
## 3     FALSE             0      <NA> 2020-11-29 04:17:18     FALSE
## 4     FALSE             0      <NA> 2020-11-29 04:15:04     FALSE
## 5     FALSE             0      <NA> 2020-11-29 04:12:39     FALSE
## 6     FALSE             0      <NA> 2020-11-29 04:07:02     FALSE
##   replyToSID                  id replyToUID
## 1       <NA> 1332904375736471554       <NA>
## 2       <NA> 1332902453126238211       <NA>
## 3       <NA> 1332901424594903041       <NA>
## 4       <NA> 1332900862428147712       <NA>
## 5       <NA> 1332900252882386952       <NA>
## 6       <NA> 1332898841532788741       <NA>
##                                                                          statusSource
## 1 <a href="https://cheapbotsdonequick.com" rel="nofollow">Cheap Bots, Done Quick!</a>
## 2                         <a href="http://instagram.com" rel="nofollow">Instagram</a>
## 3        <a href="https://merchant.media/home" rel="nofollow">Merchant Media Bot</a>
## 4            <a href="https://mobile.twitter.com" rel="nofollow">Twitter Web App</a>
## 5 <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
## 6        <a href="https://www.tweetedtimes.com" rel="nofollow">The Tweeted Times</a>
##        screenName retweetCount isRetweet retweeted longitude latitude
## 1  ChessScenarios            0     FALSE     FALSE      <NA>     <NA>
## 2      GashPhotos            0     FALSE     FALSE    -79.65     43.6
## 3 MerchantMediaCo            3      TRUE     FALSE      <NA>     <NA>
## 4  CodyReedTerry1            0     FALSE     FALSE      <NA>     <NA>
## 5 NewIndianXpress            1      TRUE     FALSE      <NA>     <NA>
## 6   ChessIndiaNet            0     FALSE     FALSE      <NA>     <NA>
```

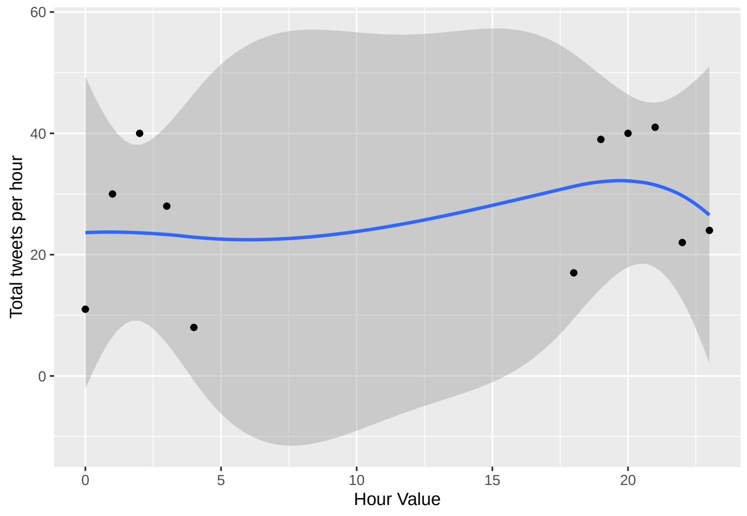
e. `dTweets` has a column showing the time the tweet was created. Generate a plot showing number of tweets on each of the hours. Add a smooth line overlaid on your plot.

```r
library(lubridate)

# convert the time into the format ymd_hms()
hour <- data.frame(hour(ymd_hms(dTweets$created)))

# add a new column in the dTweet with hour value
dTweets <- data.frame(hour, dTweets)

# rename the newly added hour column
colnames(dTweets)[1] <-"hourValue"

# filter NA values if any
dTweetsHourCount <- dTweets %>%
  filter(!is.na(hourValue)) %>%
  group_by(hourValue) %>%
  tally()

# Display Tweets per hour table
kable(dTweetsHourCount)
```

| hourValue | n |
|---:|---:|
| 0 | 11 |
| 1 | 30 |
| 2 | 40 |
| 3 | 28 |
| 4 | 8 |
| 18 | 17 |

| hourValue | n |
|---:|---:|
| 19 | 39 |
| 20 | 40 |
| 21 | 41 |
| 22 | 22 |
| 23 | 24 |

```r
# Generate a plot showing number of tweets on each of the hours
# Add a smooth line overlaid on your plot
ggplot(dTweetsHourCount, aes(x=hourValue, y=n))+
  geom_smooth(method = 'loess')+
  geom_point()+
  xlab("Hour Value")+
  ylab("Total tweets per hour")
```



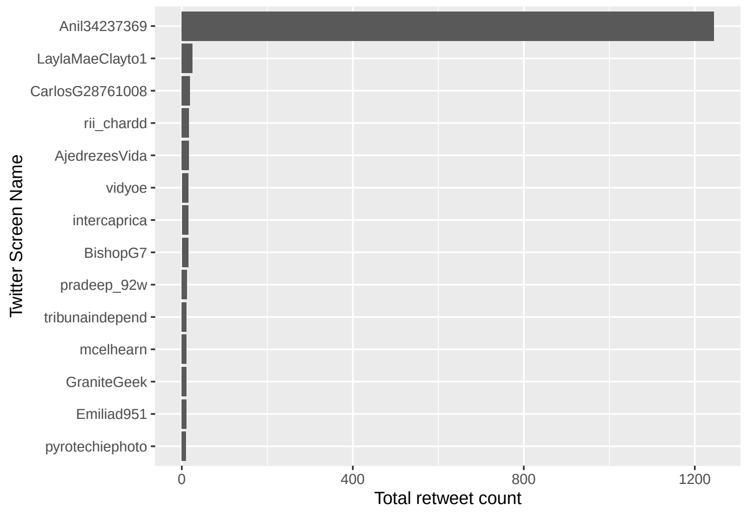
f. Arrange the dataframe `dTweets` based on the `retweetCount`. While doing this select only columns `text, screenName, retweetCount`. Store the data in a object called `mostTweets`. Display five texts that are most retweeted.

```r
#Arrange the dataframe `dTweets` based on the `retweetCount`
# While doing this select only columns `text, screenName, retweetCount`
mostTweets <- dTweets %>%
  select(text, screenName, retweetCount) %>%
  arrange(desc(retweetCount))

head(mostTweets$text,5)
```

```
## [1] "RT @KhushnumaKashm1: #Chess is the gymnasium of the mind.\nA Chess and carrom competition was o
## [2] "RT @KhushnumaKashm1: Breaking the monotonous rountine in view of COVID-19 restrictions, #Indian
## [3] "RT @MikeMavo: unprecedented levels of IQ on display #chess https://t.co/fBalNcdaNf"
## [4] "RT @dgriffinchess: NEWS: My translation of Levenfish's memoir, 'Selected Games &amp; Reminiscen
## [5] "RT @Amb_Salukvadze: The real-life Queen's Gambit: how \nNona Gaprindashvili, world's first femal
```

g. Generate a bar chart showing top 15 screen names and count of retweets from `mostTweets`. Order the bars based on the retweet counts.

```r
top15_mostTweets <- head(mostTweets, 15)

ggplot(top15_mostTweets, aes(x = reorder(screenName, retweetCount), y = retweetCount)) +
  geom_bar(stat= "Identity") +
  xlab("Twitter Screen Name") +
  ylab("Total retweet count") +
  coord_flip() +
  theme(plot.title = element_text(hjust = 0.5))
```


Anil34237369
LaylaMaeClayto1
CarlosG28761008
rii_chardd
AjedrezesVida
vidyoe
intercaprica
BishopG7
pradeep_92w
tribunaindepend
mcelhearn
GraniteGeek
Emiliad951
pyrotechiephoto
Twitter Screen Name
Total retweet count