Assignment2 - DATA 607

John Mazon

9/5/2020

Assignment – SQL and R

Objective:

Choose six recent popular movies. Ask at least five people that you know (friends, family, classmates, imaginary friends if necessary) to rate each of these movies that they have seen on a scale of 1 to 5. Take the results (observations) and store them in a SQL database of your choosing.

Load the information from the SQL database into an R dataframe

```
#Commands to load library RMYSQL and GGPLOT2
library(RMySQL)
```

```
## Loading required package: DBI
```

```
library(ggplot2)

#I used DBConnect function to be able to gain access to my localhost MYSQL database "movies"
my.database = dbConnect(MySQL(), user='root', password = 'Password1', dbname='movies', host='localhost'
dbListTables(my.database)
```

```
## [1] "movieinfo"
```

From the Movieinfo table located in the Movies database we pass the informaiton into reviews using a fetch function

```
reviews <- fetch(dbSendQuery(my.database, "SELECT * FROM movieinfo ORDER BY movieid"))
```

To make sure we have all the necessary information now and checkout dimensions, we use a DIM function. This shows us presently we have 30 entries of data with 5 Columns

```
dim(reviews)
```

```
## [1] 30  5
```

We checkout the column names of our 'reviews' by executing the colnames function. We witness that the newly obtained information currently has column names of "reviewerid" "firstname" "moviename" "movieid" "rating"

```
colnames(reviews)
```

```
## [1] "reviewerid" "firstname"  "moviename"  "movieid"    "rating"
```

Verifying all the information from our movies database has transferred in the data frame

Using str function we could observe a condensing of our data frame info, in which we can see the different SQL data types present. We could also observer a preview of our reviewers ID's, firstname, movie title, movie identification number and most importantly score.

```
str(reviews)
```

```
## 'data.frame':    30 obs. of  5 variables:
##  $ reviewerid: int  1 2 3 4 5 1 2 3 4 5 ...
##  $ firstname : chr  "Bart" "Lisa" "Homer" "Marge" ...
##  $ moviename : chr  "Mulan" "Mulan" "Mulan" "Mulan" ...
##  $ movieid   : int  1 1 1 1 1 2 2 2 2 2 ...
##  $ rating    : int  5 4 1 2 3 4 3 5 1 3 ...
```

As we've done in the past we slightly change up our columns names within our vector to more meaninful or understable terms. In this case we've used 'ID', 'ReviewerName', 'Movie', 'MovieID', 'Score'

```
colnames(reviews) <- c('ID', 'ReviewerName', 'Movie', 'MovieID', 'Score')
```

We use the colnames function once again to observe what are our current column names

```
colnames(reviews)
```

```
## [1] "ID"           "ReviewerName" "Movie"        "MovieID"      "Score"
```

Handling missing data is a foundational skill when working with SQL or R

Once again we use str function to preview our data frame information and make sure everything is formatted to our liking. Additonally, below we draw out our review/scores information for each Movie. We witness not all of the reviewers have seen every movie. Those movies without scores have transfered in with corresponding null value

```
str(reviews)
```

```
## 'data.frame':    30 obs. of  5 variables:
##  $ ID          : int  1 2 3 4 5 1 2 3 4 5 ...
##  $ ReviewerName: chr  "Bart" "Lisa" "Homer" "Marge" ...
##  $ Movie       : chr  "Mulan" "Mulan" "Mulan" "Mulan" ...
##  $ MovieID     : int  1 1 1 1 1 2 2 2 2 2 ...
##  $ Score       : int  5 4 1 2 3 4 3 5 1 3 ...
```

reviews

```
reviews$Score[which(reviews$Movie=="Mulan")]
```

```
## [1] 5 4 1 2 3
```

```
reviews$Score[which(reviews$Movie=="Invasion")]
```

```
## [1] 4 3 5 1 3
```

```
reviews$Score[which(reviews$Movie=="Artemis")]
```

```
## [1] NA  2  2  5  3
```

```
reviews$Score[which(reviews$Movie=="Capone")]
```

```
## [1]  2  5  2  2 NA
```

```
reviews$Score[which(reviews$Movie=="Tesla")]
```

```
## [1]  4  1 NA NA  2
```

```
reviews$Score[which(reviews$Movie=="Tigertail")]
```

```
## [1]  1 NA  3  5  2
```

As an example we take the movie 'Tesla' and notice that there are two null values

```
reviews$Score[which(reviews$Movie=="Tesla")]
```

```
## [1]  4  1 NA NA  2
```

Excluding Missing Values from Analyses using na.rm = True

We notice without na.rm = true the mean value returned is NA

```
meantesla = mean(reviews$Score[which(reviews$Movie=="Tesla")])
meantesla
```

```
## [1] NA
```

```
#We notice with na.rm = true the mean value actually returns a number[2.3333]
teslamean = mean(reviews$Score[which(reviews$Movie=="Tesla")], na.rm = TRUE)
teslamean
```

```
## [1] 2.333333
```

Test na.rm = True with movie that received all 5 participant scores

```
mulanmean = mean(reviews$Score[which(reviews$Movie=="Mulan")], na.rm = TRUE)
mulanmean
```

```
## [1] 3
```

```
meanmulan = mean(reviews$Score[which(reviews$Movie=="Mulan")])
meanmulan
```

```
## [1] 3
```

Below we run the following lines of code to return a mean value using only all present review scores for the 6 movies

```
mulanmean = mean(reviews$Score[which(reviews$Movie=="Mulan")], na.rm = TRUE)
mulanmean
```

```
## [1] 3
```

```
invasionmean = mean(reviews$Score[which(reviews$Movie=="Invasion")], na.rm = TRUE)
invasionmean
```

```
## [1] 3.2
```

```
artemismean = mean(reviews$Score[which(reviews$Movie=="Artemis")], na.rm = TRUE)
artemismean
```

```
## [1] 3
```

```
caponemean = mean(reviews$Score[which(reviews$Movie=="Capone")], na.rm = TRUE)
caponemean
```

```
## [1] 2.75
```

```
teslamean = mean(reviews$Score[which(reviews$Movie=="Tesla")], na.rm = TRUE)
teslamean
```

```
## [1] 2.333333
```

```
tigertailmean = mean(reviews$Score[which(reviews$Movie=="Tigertail")], na.rm = TRUE)
tigertailmean
```

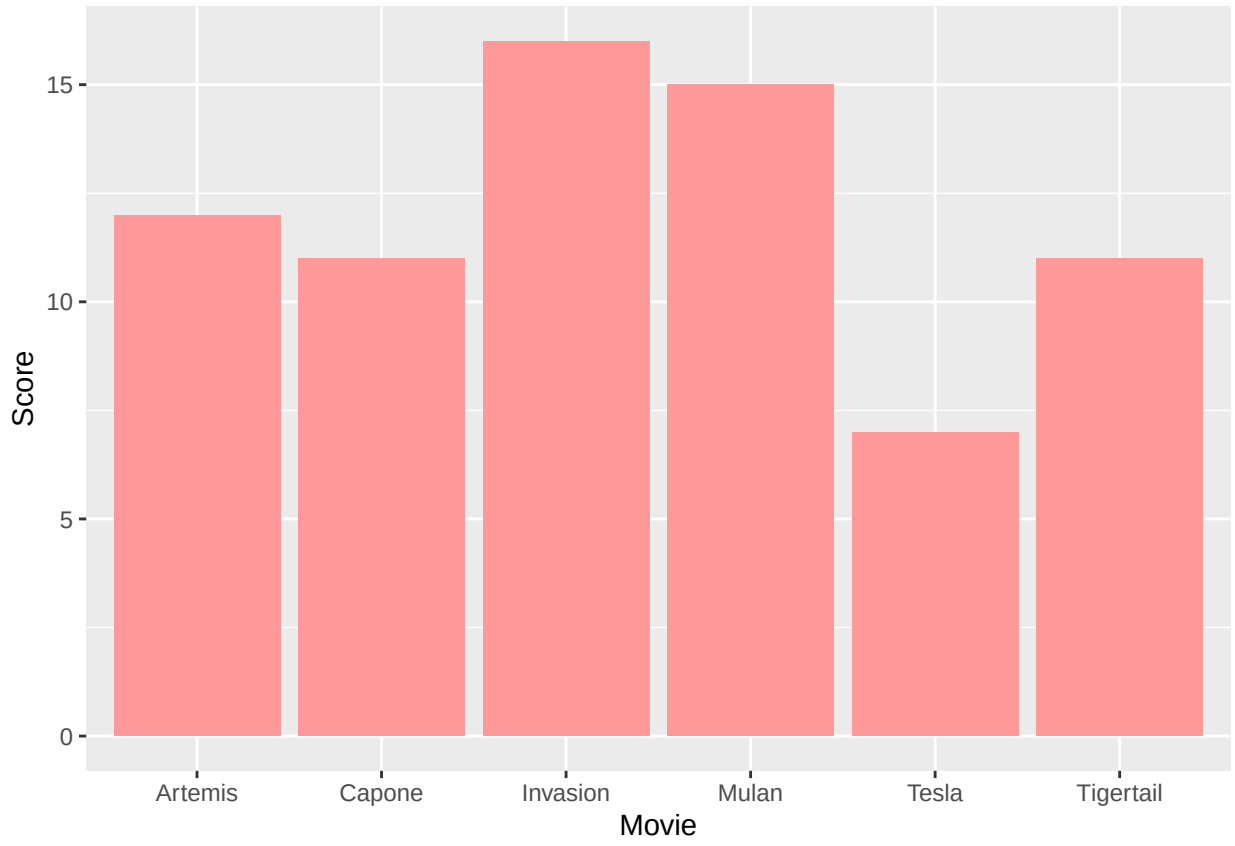
```
## [1] 2.75
```

Graphical Analysis of total movie ratings

Using the ggplot2 library and geombar we create a nice visual of the 'count up' of all the scores given by the reviewers for each movie. We have labeled the x-axis referencing the movie title and the y-axis labeled referencing the score. Notice that conveniently we have our Graphic depict our information with 5 rows removed containing missing values

```r
library(ggplot2)
# Basic barplot
p<-ggplot(data=reviews, aes(x=Movie, y=Score)) +
  geom_bar(stat="identity",fill="#FF9999")
p
```

```
## Warning: Removed 5 rows containing missing values (position_stack).
```



Conclusion

In conclusion, we witness that Invasion was the best rated and overall most watched movie. A few reviewers didn't watch hence didn't rate the movie Tesla, which is why it ended in last place. Capone and Tigertail are nicely depicted with a tie in score.