Amy McVicar

6/18/2019

IST 687

Project

DATA SCIENCE PROJECT: COUNTRY POPULATION DENSITY & MOVIE PLOTS

CONTENTS

[INTRODUCTION 2](#_Toc11752223)

[BUSINESS QUESTIONS 2](#_Toc11752224)

[DATA ACQUISITION, CLEANSING, & TRANSFORMING 2](#_Toc11752225)

[Data Acquisition Process 2](#_Toc11752226)

[Movie Plots 2](#_Toc11752227)

[Global Country Demographics 2](#_Toc11752228)

[Data Dictionaries 2](#_Toc11752229)

[Movie Plot Data Dictionary 2](#_Toc11752230)

[Census Data Dictionary 3](#_Toc11752231)

[Cleansing 3](#_Toc11752232)

[Movie Plots 3](#_Toc11752233)

[Country Information 3](#_Toc11752234)

[Transforming 3](#_Toc11752235)

[Movie Plots 3](#_Toc11752236)

[Country Information 3](#_Toc11752237)

[DATA 4](#_Toc11752238)

[Movie Plots 4](#_Toc11752239)

[Country Information 4](#_Toc11752240)

[RESULTS 4](#_Toc11752241)

[Sentiment Analysis 4](#_Toc11752242)

[Movie Plot (+Genre) Word Clouds & Top 10 Genre Summary 5](#_Toc11752243)

[Top 10 Genres per Country 6](#_Toc11752244)

[Word Clouds 7](#_Toc11752245)

[Trends 9](#_Toc11752246)

[Models & Forecasts 10](#_Toc11752247)

[CONCLUSION 11](#_Toc11752248)

[CODE 11](#_Toc11752249)

# INTRODUCTION

This report looks at movie plots and genres from 15 countries compared to the country’s population and density to discover if there are any associations. As part of this investigation, the report will explore if there are any trends over time or by a country that can also be used to predict the tenure or quantity of movies to be produced by the 15 countries included in the report, going forward.

# BUSINESS QUESTIONS

* Is there a movie genre mix per country?
* Sentiment per Country?
* Is there an association between sentiment or genres and country demographics?
* Can you predict sentiment or genre based on country demographics?

# DATA ACQUISITION, CLEANSING, & TRANSFORMING

## Data Acquisition Process

Two data sets are needed for the project: one with movies and needed data points, such as year, country, plot, and genre and another with the associated countries demographics for area and population by year.

#### Movie Plots

Retrieved from <https://www.kaggle.com/jrobischon/wikipedia-movie-plots>

The first dataset contains 34,886 movie records scraped from Wikipedia in January 2019. The dataset was retrieved from the Kaggle website on 5/12/19.

#### Global Country Demographics

Retrieved from <https://www.census.gov/data-tools/demo/idb/informationGateway.php>

The second dataset is curated and provided by the US government. The country information is sourced from the countries to which it relates

## Data Dictionaries

|  |  |  |
| --- | --- | --- |
| Movie Plot Data Dictionary | | |
| Field Name | Description | Data Type |
| Release Year | Year in which the movie was released | Integer |
| Title | Movie title | String |
| Origin/Ethnicity | Origin of the movie (i.e., American, Bollywood, Tamil, etc.) | String |
| Director | Director(s) | String |
| Cast | Main actors/actresses | String |
| Genre | Movie genre(s) | String |
| Wiki Page | URL of Wikipedia page | URL |

|  |  |  |
| --- | --- | --- |
| Census Data Dictionary | | |
| Field Name | Description | Data Type |
| Country | Country | String |
| Year | Year | Integer |
| Population | Population | Integer |
| Area | Area (sq. km.) | Integer |
| Density | Density (persons per sq. km.) | Integer |

## Cleansing

#### Movie Plots

The movie file from Kaggle was very clean. The biggest challenge was cleaning up the genres. Built-in text functions efficiently handled the long plot data. Only three records were lost in the reformatting process of interpreting the comma delimited format. UTF-8 encoding should have been handled at this stage but was neglected to cause complications later in the process. Regardless, considering the range of countries involved, the encoding issue impacted very little data and did not affect results.

#### Country Information

The country file required no cleaning. It was downloadable in a cvs file that was prepared for data use with the removal of a few leading rows.

## Transforming

#### Movie Plots

The movie file did not include the country of origin information. Origin/Ethnicity was used to derive and append country information to match to Country Demographics data set.

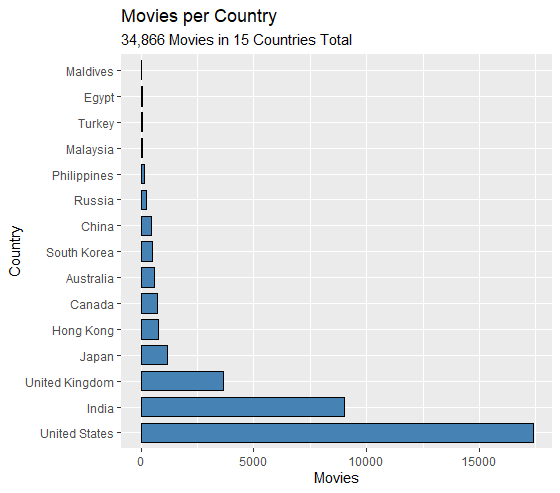
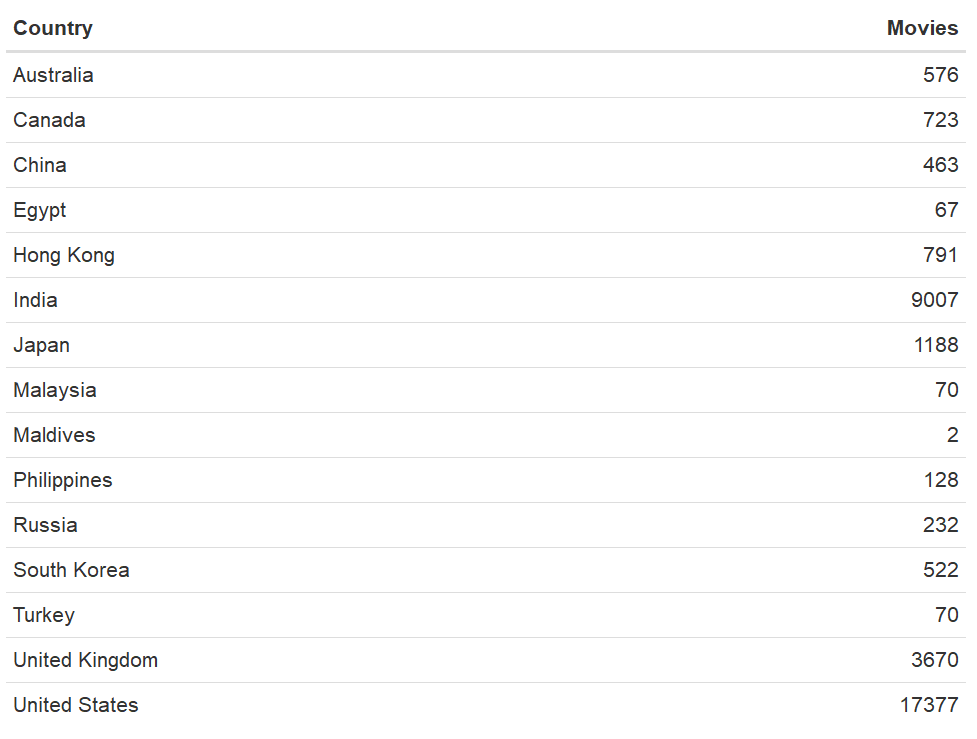
#### Country Information

Due to time constraints of the project, a year to year match wasn’t possible, and the country information was limited after retrieval to the metrics for 2017, which matches the most recent release date in the movie database. The 2017 country population, area, and density data was used in the analysis.

# DATA

## Movie Plots

15 Countries are represented, and movie release dates range from 1901 to 2017.



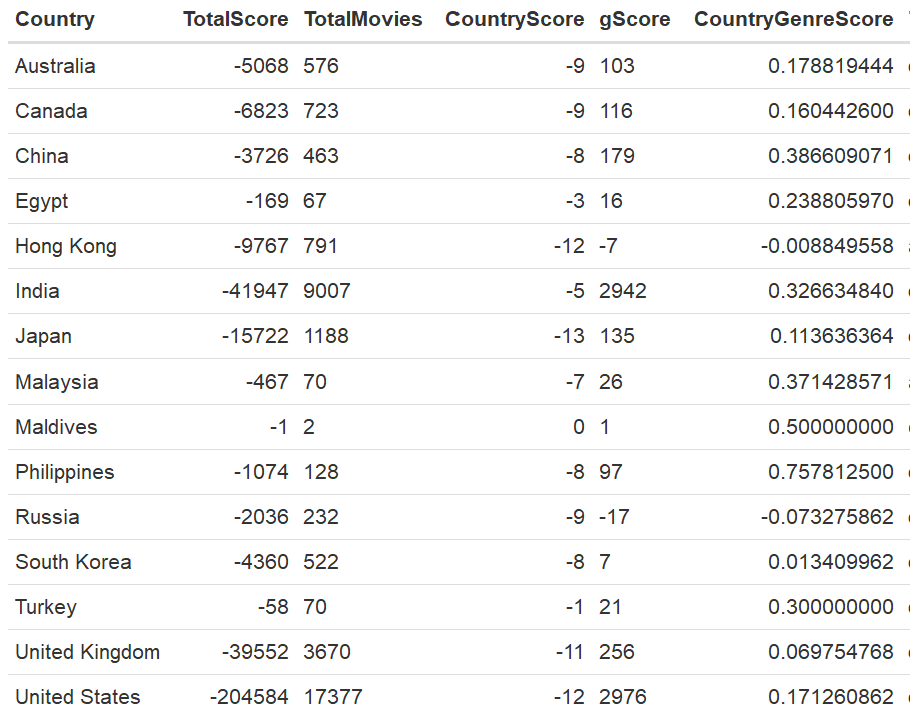
## Country Information

The country demographics span 25 years and provide population, area, and density per year.

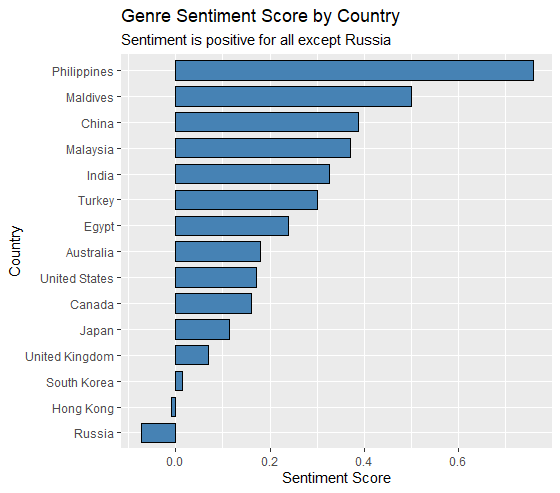
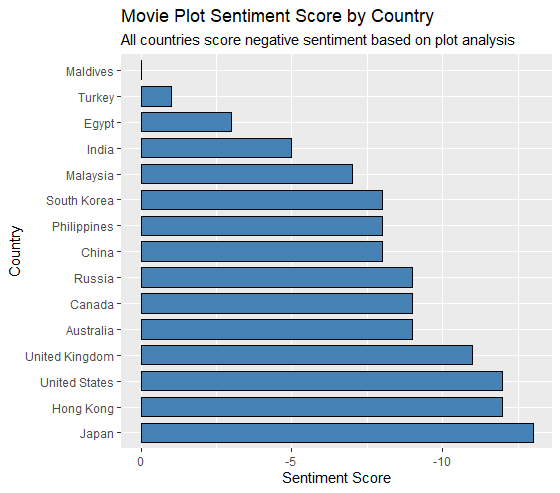
# RESULTS

## Sentiment Analysis

The sentiment is negative for all countries when based on movie plots, but when genre alone is analyzed all but Russia show positive sentiment. Since plots are generally based on a challenge that is overcome during the course of the film, this result is of no surprise.



* **CountryScore** is the total genre and plot sentiment score divided by movies for the country.
* **gScore** is the total sentiment score for the country based on genre only.
* **CountryGenreScore** is the genre sentiment score divided by movies for the country.



### Movie Plot (+Genre) Word Clouds & Top 10 Genre Summary

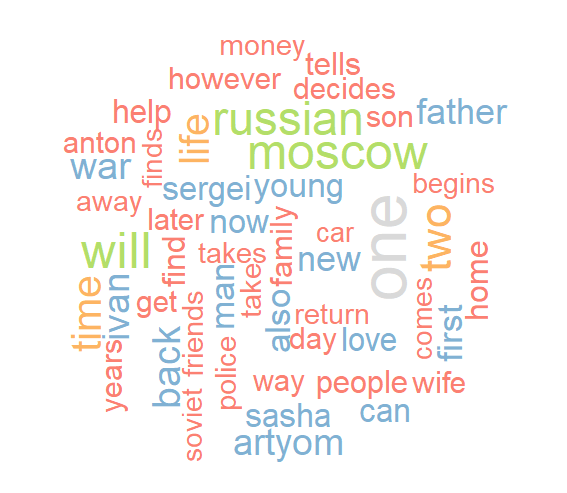
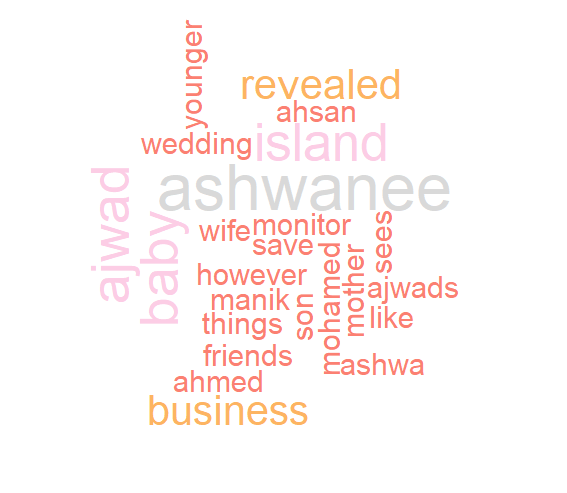
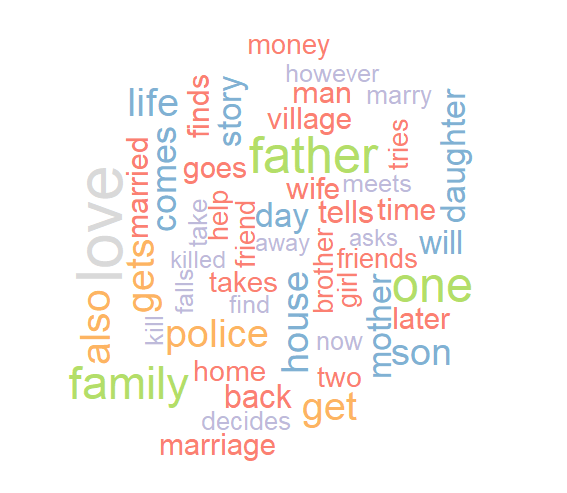
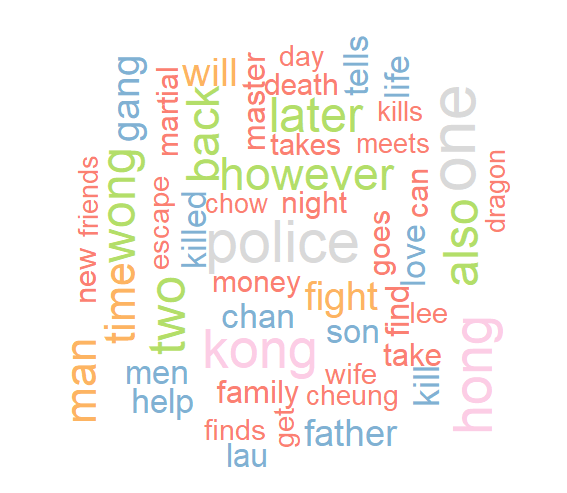
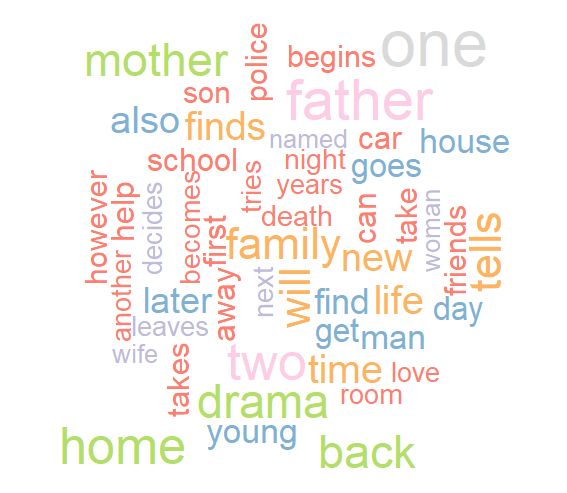
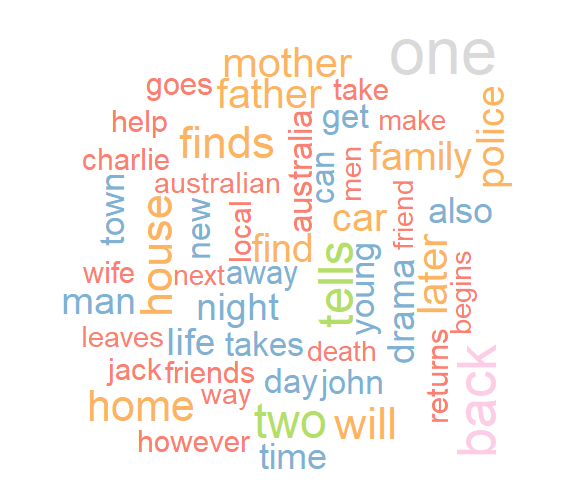
Movie word clouds show a very similar trend amongst all countries to have a high frequency for family terms. Police is also a common plot term amongst countries. There are some country-specific trends visible via the word cloud and seen within the top ten genres for each country, although it seems the similarities outweigh the differences for this level of analysis.

* The United States likes cars, and this word shows up in their plots. Westerns are also a top 10 US genre.
* The UK has a top 10 genre of World War II movies.
* Hong Kong includes martial arts and kung fu in its top 10 genre list.
* Japan made enough Godzilla movies that Godzilla shows up in their word cloud.

#### Top 10 Genres per Country

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Australia** |  | **Canada** |  | **China** |  | **Egypt** |  |  | **Hong Kong** |  |
| drama | 41% | drama | 49% | drama | 32% | drama |  | 57% | action | 24% |
| comedy | 21% | comedy | 16% | action | 16% | romance |  | 13% | comedy | 17% |
| thriller | 11% | horror | 9% | comedy | 12% | comedy |  | 11% | drama | 12% |
| horror | 8% | animated | 6% | romance | 11% | crime |  | 7% | martialarts | 11% |
| action | 5% | thriller | 5% | fantasy | 6% | romcom |  | 4% | crime | 11% |
| adventure | 4% | short | 4% | crime | 5% | musical |  | 2% | horror | 6% |
| crime | 3% | board | 3% | adventure | 5% | political |  | 2% | thriller | 6% |
| animated | 3% | national | 3% | romcom | 5% | biography |  | 1% | romcom | 5% |
| scifi | 2% | crime | 3% | historical | 5% | historical |  | 1% | romance | 4% |
| romance | 2% | scifi | 3% | mystery | 4% |  |  | 0% | kungfu | 3% |
|  |  |  |  |  |  |  |  |  |  |  |
| **India** |  | **Japan** |  | **Malaysia** |  | **Maldives** |  |  | **Philippines** |  |
| drama | 29% | drama | 27% | action | 22% | comedy |  | 33% | drama | 28% |
| romance | 18% | anime | 14% | drama | 22% | suspense |  | 33% | comedy | 14% |
| action | 17% | action | 11% | comedy | 19% | thriller |  | 33% | horror | 13% |
| comedy | 11% | scifi | 11% | horror | 15% |  |  | 0% | romcom | 13% |
| thriller | 10% | fantasy | 9% | romance | 8% |  |  | 0% | romance | 12% |
| family | 4% | horror | 9% | historical | 3% |  |  | 0% | action | 6% |
| romcom | 3% | comedy | 8% | romcom | 3% |  |  | 0% | fantasy | 4% |
| social | 3% | toktsu | 4% | animated | 2% |  |  | 0% | suspense | 4% |
| crime | 3% | romance | 4% | crime | 2% |  |  | 0% | thriller | 3% |
| horror | 2% | thriller | 3% | mystery | 2% |  |  | 0% | adventure | 2% |
|  |  |  |  |  |  |  |  |  |  |  |
| **Russia** |  | **Turkey** |  | **South Korea** |  | **United Kingdom** | |  | **United States** |  |
| drama | 36% | drama | 58% | drama | 49% | drama |  | 32% | drama | 32% |
| comedy | 21% | comedy | 26% | action | 10% | comedy |  | 29% | comedy | 27% |
| war | 9% | action | 3% | ero | 9% | crime |  | 10% | horror | 6% |
| crime | 7% | horror | 3% | melodrama | 8% | thriller |  | 8% | western | 6% |
| thriller | 6% | romance | 3% | comedy | 5% | horror |  | 6% | crime | 6% |
| action | 5% | animated | 2% | horror | 5% | adventure |  | 3% | action | 5% |
| historical | 5% | documentary | 2% | thriller | 5% | musical |  | 3% | scifi | 5% |
| fantasy | 4% | romcom | 2% | romcom | 4% | worldwarii |  | 3% | musical | 4% |
| scifi | 4% | thriller | 2% | animated | 3% | mystery |  | 2% | thriller | 4% |
| adventure | 3% |  | 0% | historical | 3% | romance |  | 2% | animated | 4% |

#### Word Clouds



India

Hong Kong

Australia

China

Egypt

Canada

Japan

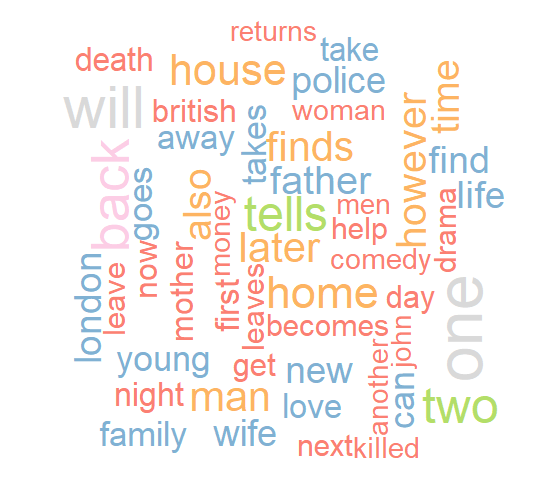
Malaysia

Philippines

Maldives

South Korea

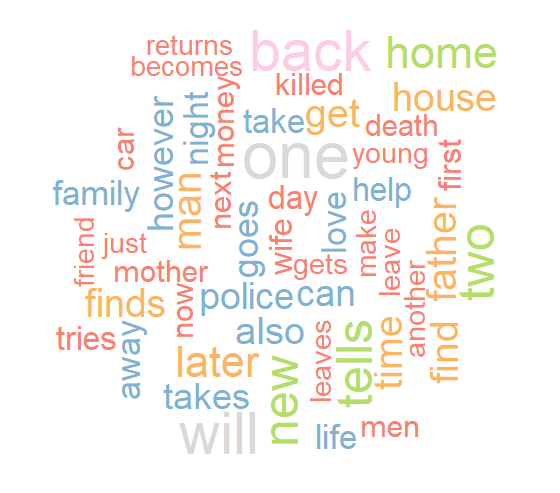
Russia



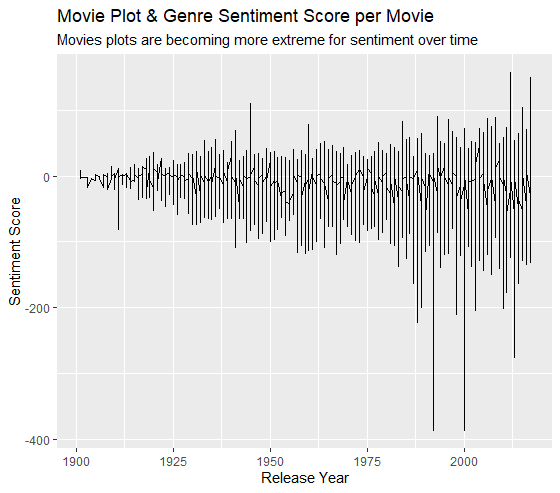
United States

United Kingdom

Turkey



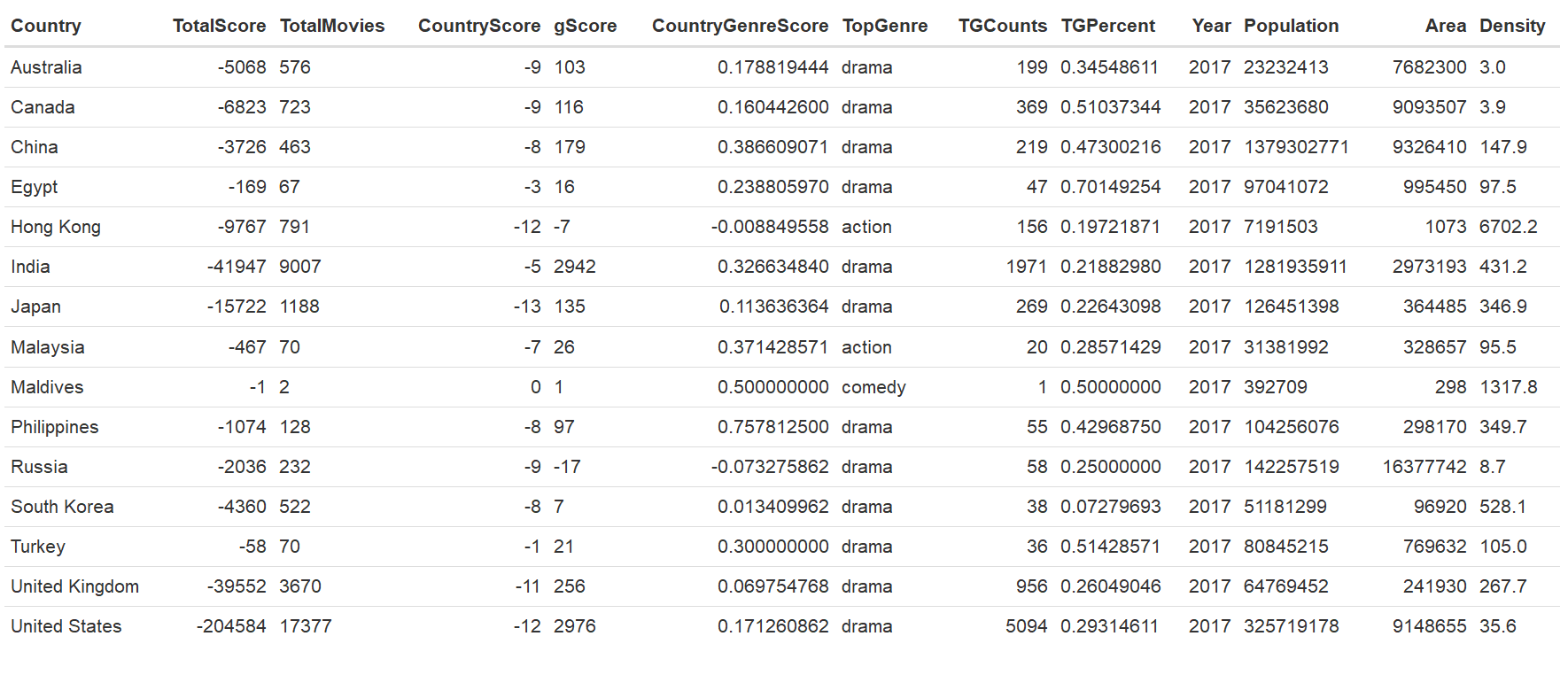
## Trends



## Models & Forecasts

There does seem to be slight difference between countries, but there is not enough variation in the top genre, which is drama for 12 out of 15 countries and a correlation between the country demographics of the population, area, and density to genre or sentiment score.

Table of data summary by Country, followed by statistical evaluation:



TGCounts TGPercent Year Population

Min. : 1.0 Min. :0.0728 Min. :2017 Min. :3.927e+05

1st Qu.: 42.5 1st Qu.:0.2382 1st Qu.:2017 1st Qu.:3.350e+07

Median : 156.0 Median :0.2931 Median :2017 Median :8.085e+07

Mean : 632.5 Mean :0.3519 Mean :2017 Mean :2.501e+08

3rd Qu.: 319.0 3rd Qu.:0.4865 3rd Qu.:2017 3rd Qu.:1.344e+08

Max. :5094.0 Max. :0.7015 Max. :2017 Max. :1.379e+09

Area Density

Min. : 298 Min. : 3.00

1st Qu.: 270050 1st Qu.: 65.55

Median : 769632 Median : 147.90

Mean : 3846561 Mean : 696.05

3rd Qu.: 8387904 3rd Qu.: 390.45

Max. :16377742 Max. :6702.20

Country TotalScore TotalMovies CountryScore

Australia:1 Min. :-204584.0 Min. : 2.0 Min. :-13.000

Canada :1 1st Qu.: -12744.5 1st Qu.: 99.0 1st Qu.:-10.000

China :1 Median : -4360.0 Median : 522.0 Median : -8.000

Egypt :1 Mean : -22356.9 Mean : 2325.7 Mean : -7.667

Hong Kong:1 3rd Qu.: -770.5 3rd Qu.: 989.5 3rd Qu.: -6.000

India :1 Max. : -1.0 Max. :17377.0 Max. : 0.000

(Other) :9

gScore CountryGenreScore TopGenre

Min. : -17.0 Min. :-0.07328 Length:15

1st Qu.: 11.5 1st Qu.: 0.09170 Class :character

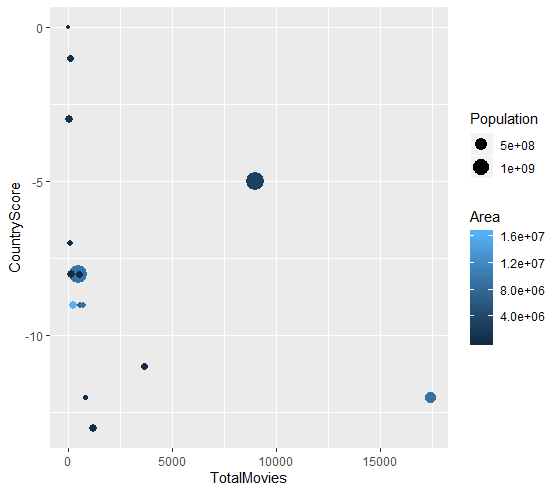
Median : 97.0 Median : 0.17882 Mode :character

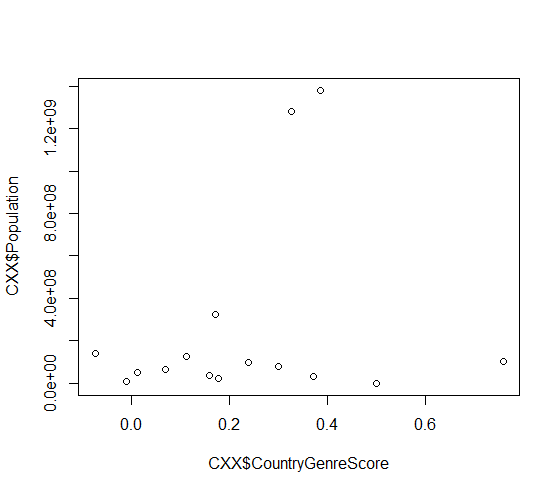
Mean : 456.7 Mean : 0.23377

3rd Qu.: 157.0 3rd Qu.: 0.34903

Max. :2976.0 Max. : 0.75781

Max. :16377742 Max. :6702.20





2 Plots testing for visual confirmation of patterns and variations in data:

# 

Testing for Density predicting Country Sentiment Score:

# 

Testing for Density predicting Country Sentiment Score:

Residuals:

Min 1Q Median 3Q Max

-5.9599 -2.1006 -0.2321 1.4269 7.1026

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -7.103e+00 1.389e+00 -5.112 0.000256 \*\*\*

Population 1.132e-09 2.520e-09 0.449 0.661275

Area -2.201e-07 2.188e-07 -1.006 0.334245

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.062 on 12 degrees of freedom

Multiple R-squared: 0.08047, Adjusted R-squared: -0.07279

F-statistic: 0.5251 on 2 and 12 DF, p-value: 0.6045

# CONCLUSION

Unfortunately, we can see from the data there is not enough variation in the data to predict genre based on the country statistics of population, area, and density. It will take more data and in-depth analysis to predict genre mixes per country or trends over time. There are indicators of variations per country, and it would take far more data to build a prediction model, where it might be driven by culture, tradition, economic resources in the movie industry, the appetite of market, the market served and government censorship.

* Is there a movie genre mix per country?
  + Yes. As seen in the top 10 genres per country. However, the top genre is drama 80% of the time.
* Sentiment per Country?
  + Yes. Each country has a sentiment score in the negative, ranging from 0 to -13.
* Is there an association between sentiment or genres and country demographics?
  + No. There doesn’t seem to be for the country demographics included in the report (population, area, and density). There are differences in countries though that could be driven by factors not explored in this report or available in this data set.
* Can you predict sentiment or genre based on country demographics?
  + No, you can’t predict sentiment or genre based on population, area, or density.

# CODE

########################################

#Amy McVicar

#6/18/2019

#IST 687

#DATA SCIENCE PROJECT: COUNTRY POPULATION DENSITY & MOVIE PLOTS

########################################

#LIBRARIES

install.packages("tidytext")

library(tidytext)

install.packages("NLP")

library(NLP)

install.packages("tm")

library(tm)

install.packages("dplyr")

library(dplyr)

install.packages("textdata")

library(textdata)

install.packages("data.table")

library(data.table)

install.packages("sqldf")

library(sqldf)

install.packages("ggplot2")

library(ggplot2)

install.packages("wordcloud")

library(wordcloud)

install.packages("formattable")

library(formattable)

install.packages("openxlsx")

library(openxlsx)

#install.packages("wordcloud2")

#library(wordcloud2)

#install.packages("gtable")

#library(gtable)#install.packages("tidyverse")

#library(tidyverse)

install.packages("arules")

library(arules)

########################################

#LOAD MOVIE FILE & MAKE COPY

movies\_0 <- read.csv("/Data/wiki\_movie\_plots\_deduped.csv")

#copy movie file

movies <- movies\_0

########################################

#CLEAN MOVIE GENRE

movies$Genre <- trimws(movies$Genre )

movies$Genre <- gsub("/", " \\| ",movies$Genre)

movies$Genre <- gsub(",", " \\|",movies$Genre)

movies$Genre <- gsub(" ", " ",movies$Genre)

movies$Genre <- gsub(" ", " ",movies$Genre)

movies$Genre <- gsub("film", "",movies$Genre)

movies$Genre <- gsub("usa", "",movies$Genre)

movies$Genre <- gsub("usa | can", "",movies$Genre)

movies$Genre <- gsub("-", "",movies$Genre)

movies$Genre <- gsub("[140]", "",movies$Genre)

movies$Genre <- gsub("[144]", "",movies$Genre)

movies$Genre <- gsub("( genre)", "",movies$Genre)

movies$Genre <- gsub("| 7", "",movies$Genre)

movies$Genre <- gsub("\\[\\]", "",movies$Genre)

movies$Genre <- gsub("&", "\\|",movies$Genre)

movies$Genre <- gsub("\\(\\)", "\\|",movies$Genre)

movies$Genre <- gsub("fantasychildren's", "fantasy children's",movies$Genre)

movies$Genre <- gsub("fantasycomedy", "fantasy comedy",movies$Genre)

movies$Genre <- gsub("fantasyperiod", "fantasy period",movies$Genre)

movies$Genre <- gsub("\\|thriller", "\\| thriller",movies$Genre)

movies$Genre <- gsub("action adventure science fiction", "action adventure \\| science fiction",movies$Genre )

movies$Genre <- gsub("horror in 3d\\.", "horror \\| 3d",movies$Genre )

movies$Genre <- gsub("science fiction", "sci-fi",movies$Genre )

movies$Genre <- gsub("science fiction", "sci-fi",movies$Genre )

movies$Genre <- gsub("scifi", "sci-fi",movies$Genre )

movies$Genre <- gsub("\\[\\]", "sci-fi",movies$Genre )

movies$Genre <- gsub("sci-fi horror", "sci-fi \\| horror",movies$Genre )

movies$Genre <- gsub("action???masala", "action \\| masala",movies$Genre )

movies$Genre <- gsub("actionadventure", "action \\| adventure",movies$Genre )

movies$Genre <- gsub("; ", " \\|",movies$Genre )

movies$Genre <- gsub("ancientcostume", " ancient costume",movies$Genre )

movies$Genre <- gsub("6.", "",movies$Genre )

movies$Genre <- gsub(".mm", "",movies$Genre )

movies$Genre <- gsub("adventurecomedy" , "adventure \\| comedy",movies$Genre )

movies$Genre <- gsub("and", "\\|", movies$Genre )

movies$Genre <- gsub("horror comedy \\| horror","horror \\| comedy",movies$Genre )

movies$Genre <- gsub("horror masala", "horror \\| masala",movies$Genre )

movies$Genre <- gsub("horror musical" , "horror \\| musical",movies$Genre )

movies$Genre <- gsub("horror masala", "horror \\| masala",movies$Genre )

movies$Genre <- gsub("horrorthriller", " horror \\| thriller",movies$Genre )

movies$Genre <- gsub("imax" , "",movies$Genre )

movies$Genre <- gsub("imdb", "",movies$Genre )

movies$Genre <- gsub("comedydrama", "comedy drama",movies$Genre )

movies$Genre <- gsub("sciencefiction", "sci-fi",movies$Genre )

movies$Genre[14853] <- "action"

movies$Genre[15508] <- "action"

movies$Genre[31151] <- "action"

movies$Genre[17256] <- "comdedy | drama"

movies$Genre <- gsub("world war i", "world-war-i",movies$Genre )

movies$Genre <- gsub("world war ii", "world-war-ii",movies$Genre )

movies$Genre <- gsub("martial arts", "martial-arts",movies$Genre )

movies$Genre <- gsub("comingofage", "coming-of-age",movies$Genre )

movies$Genre <- gsub("coming of age", "coming-of-age",movies$Genre )

movies$Genre <- gsub("romcom", "rom-com",movies$Genre )

movies$Genre <- gsub("rom com", "rom-com",movies$Genre )

movies$Genre <- gsub("romantic comedy", "rom-com",movies$Genre )

movies$Genre <- gsub("crimethriller", "crime thriller",movies$Genre )

movies$Genre <- gsub("//[not in citation given//]", "",movies$Genre )

movies$Genre <- gsub("comedyhorror", "comedy horror",movies$Genre )

movies$Genre <- gsub("comedythriller", "comedy thriller",movies$Genre )

movies$Genre <- gsub("comedy \\| romance", "rom-com",movies$Genre)

movies$Genre <- gsub("romance \\| comedy", "rom-com",movies$Genre)

movies$Genre <- gsub("comedy romance", "rom-com",movies$Genre)

movies$Genre <- gsub("romance comedy", "rom-com",movies$Genre)

movies$Genre <- gsub("romantic comedy", "rom-com",movies$Genre)

movies$Genre <- gsub("romantic", "romance",movies$Genre)

#movies$Genre <- gsub("unknown", "",movies$Genre )

movies$Genre <- gsub("action\\S", "action ",movies$Genre)

movies$Genre <- gsub("action omedy", "action \\| comedy ",movies$Genre)

movies$Genre <- gsub("animation", "animated",movies$Genre)

movies$Genre <- gsub(" ", " ",movies$Genre)

movies$Genre <- gsub("biographical", "biography",movies$Genre)

movies$Genre <- gsub("biographic", "biography",movies$Genre)

movies$Genre <- gsub("biopic", "biography",movies$Genre)

movies$Genre <- gsub("action hriller", "action \\| thriller",movies$Genre)

movies$Genre <- gsub("action ove", "action",movies$Genre)

movies$Genre <- gsub("action rama", "action",movies$Genre)

movies$Genre <- gsub("kung fu", "kung-fu",movies$Genre)

movies$Genre <- gsub("martialarts", "martial-arts",movies$Genre)

movies$Genre[movies$Genre == ""] <- "unknown"

movies$Genre[movies$Genre == " "] <- "unknown"

#Create Genre word columns for Sentiment & Word Cloud

movies$gSent <- movies$Genre

movies$gSent <- gsub("unknown", "",movies$gSent)

movies$wCloud <- movies$Genre

movies$gSent <- gsub("\\|", " ",movies$gSent)

movies$wCloud <- gsub("\\|", " ",movies$wCloud)

movies$wCloud <- gsub("unknown", "",movies$wCloud)

# update rom-com for Sentiment Analysis

movies$gSent <- gsub("rom-com", "romantic comedy",movies$gSent)

movies$wCloud <- gsub("rom-com", "romantic comedy",movies$gSent)

########################################

# COUNTS

# if can't find count(), reload dplyr package

#install.packages("dplyr")

#library(dplyr)

movieCounts <- count(movies, movies$Genre, name = "Movies")

movieCounts <- count(movies, movies$gSent, name = "Movies")

movieCounts <- count(movies, movies$wCloud, name = "Movies")

movieCounts <- count(movies,movies$Origin.Ethnicity, name = "Movies")

########################################

#ADD COUNTRY INFORMATION

#begin add country & ISO Codes process

movies$Country <- "unknown"

movies$ISO <- "unknown"

# Create CountryCodes Table

Ethnicity <- c("Australian", "Canadian", "Chinese", "Hong Kong", "Egyptian",

"Assamese", "Bangladeshi", "Bengali", "Bollywood", "Kannada",

"Malayalam", "Marathi", "Punjabi", "Tamil", "Telugu",

"Japanese", "Malaysian", "Maldivian", "Filipino", "Russian",

"South\_Korean", "Turkish", "British", "American")

Country <- c("Australia", "Canada", "China", "Hong Kong", "Egypt", "India",

"India", "India", "India", "India", "India", "India", "India",

"India", "India", "Japan", "Malaysia", "Maldives", "Philippines",

"Russia", "South Korea", "Turkey", "United Kingdom",

"United States")

ISO <- c("AU", "CA", "CN", "HK", "IN", "IN", "IN", "IN", "IN", "IN", "IN",

"IN", "IN", "IN", "IN", "JP", "MY", "MV", "PH", "RU", "KR", "TR",

"GB", "US")

CountryCodes <- data.frame(Ethnicity,Country, ISO)

# apply country data to movie table

movies$Country <- CountryCodes$Country[match(movies$Origin.Ethnicity,CountryCodes$Ethnicity)]

movies$ISO <- CountryCodes$ISO[match(movies$Country,CountryCodes$Country)]

# Counts - Country

movieCounts <- count(movies, Country, name = "Movies")

# create table graphic - counts by country

formattable(movieCounts, align = c("l", rep("r", NCOL(movieCounts) - 1)))

#create graph by movie counts

ggplot(movieCounts, aes(x=reorder(Country, -Movies),y=Movies, fill = variable, width = .75))+geom\_bar(colour="black",fill="steel blue", stat="identity")+coord\_flip()+ labs(x = "Country", y = "Movies", title = "Movies per Country", subtitle = "34,866 Movies in 15 Countries Total")

# Counts - ISO

movieCounts <- count(movies, movies$ISO, name = "Movies")

########################################

# SENTIMENT BY MOVIE

# get AFINN

# if can't find get\_sentiments, reload tidytrext package

#install.packages("tidytext")

#library(tidytext)

afinn <- get\_sentiments(lexicon = c("afinn"))

#Prepare for Plot Sentiment Loop

sList <- character()

counter <- 0

#Sentiment Loop

for (i in 1:nrow(movies)) {

counter <- (counter + 1)

Y <- character()

vPlot <- movies$Plot[counter]

words.vec <- VectorSource(vPlot)

words.corpus <- Corpus(words.vec)

words.corpus <- tm\_map(words.corpus, content\_transformer(tolower))

words.corpus <- tm\_map(words.corpus, removePunctuation)

words.corpus <- tm\_map(words.corpus, removeNumbers)

words.corpus <- tm\_map(words.corpus, removeWords, stopwords("english"))

tdm <- TermDocumentMatrix(words.corpus)

m <- as.matrix(tdm)

wordCounts <- rowSums(m)

X <- data.frame(wordCounts)

setDT(X, keep.rownames = TRUE)[]

colnames(X) <- c("Names", "Freq")

join\_string <- "select X.\*, afinn.\* from X join afinn on X.Names=afinn.word"

newX <- sqldf(join\_string,stringsAsFactors = FALSE)

if (nrow(newX)==0)

{Y<-0}

else {

newX$FreqScore <- (newX$Freq\*newX$value)

Y <- sum(newX$FreqScore)

}

sList <- c(sList, Y)

}

#release vPlot memory

vPlot <- character()

#Add Plot Sentiments to movies

movies$Sentiment <- sList

#Prepare for Genre Sentiment Loop

sList <- character()

rowcount <- nrow(movies)

counter <- 0

#Genre Sentiment Loop

for (i in 1:rowcount) {

counter <- (counter + 1)

Y <- character()

vsG <- movies$gSent[counter]

words.vec <- VectorSource(vsG)

words.corpus <- Corpus(words.vec)

words.corpus <- tm\_map(words.corpus, content\_transformer(tolower))

words.corpus <- tm\_map(words.corpus, removePunctuation)

words.corpus <- tm\_map(words.corpus, removeNumbers)

words.corpus <- tm\_map(words.corpus, removeWords, stopwords("english"))

tdm <- TermDocumentMatrix(words.corpus)

m <- as.matrix(tdm)

wordCounts <- rowSums(m)

X <- data.frame(wordCounts)

setDT(X, keep.rownames = TRUE)[]

colnames(X) <- c("Names", "Freq")

join\_string <- "select X.\*, afinn.\* from X join afinn on X.Names=afinn.word"

newX <- sqldf(join\_string,stringsAsFactors = FALSE)

if (nrow(newX)==0)

{Y<-0}

else {

newX$FreqScore <- (newX$Freq\*newX$value)

Y <- sum(newX$FreqScore)

}

sList <- c(sList, Y)

}

#Add Genre sentiments to movies

movies$gScore<- sList

########################################

#Total Sentiment Score

movies$Sentiment <- as.numeric(movies$Sentiment)

movies$gScore <- as.numeric(movies$gScore)

movies$TotalScore <- movies$Sentiment+movies$gScore

########################################

#Graph Sentiment per Country

#CountryStats <- CountryStats[,-1:-6]

CountryStats <- data.frame(tapply(movies$TotalScore,movies$Country, sum))

CountryStats$TotalMovies <- c(tapply(movies$TotalScore,movies$Country, length))

setDT(CountryStats, keep.rownames = TRUE)[]

colnames(CountryStats) <- c("Country", "TotalScore","TotalMovies")

CountryStats$CountryScore <- c(round(CountryStats$TotalScore/CountryStats$TotalMovies))

#GGPLOT Sent by county

ggplot(CountryStats, aes(x=reorder(Country, CountryScore),y=CountryScore, fill = variable, width = .75))+geom\_bar(colour="black",fill="steel blue", stat="identity")+coord\_flip()+scale\_y\_reverse()+ labs(x = "Country", y = "Sentiment Score", title = "Movie Plot Sentiment Score by Country", subtitle = "All countries score negative sentiment based on plot analysis")

########################################

#Graph Sentiment per Country based on Genre

CountryStats$gScore <- c(tapply(movies$gScore,movies$Country,sum))

CountryStats$CountryGenreScore <- c((CountryStats$gScore/CountryStats$TotalMovies))

#Table of Sent

formattable(CountryStats,align = c("l", rep("r", NCOL(movieCounts) - 1)))

ggplot(CountryStats, aes(x=reorder(Country, CountryGenreScore), y=CountryGenreScore, fill = variable, width = .75))+geom\_bar(colour="black",

fill="steel blue", stat="identity")+coord\_flip()+

labs(x = "Country", y= "Sentiment Score",

title = "Genre Sentiment Score by Country", subtitle = "Sentiment is positive for all except Russia")

########################################

#WORD CLOUD

#create country files - loop

for(i in unique(movies$Country)) {

nam <- paste("movies", i, sep = ".")

assign(nam, movies[movies$Country==i,])

}

###################################

#Word Loop & Function

#prepare for word loop

s1 <- character()

sList <- character()

rowcount <- nrow(movies)

counter <- 0

#word loop & function

fWord <- function(countryfile)

{

testtab <- countryfile

counter <- 0

s1 <- character()

for (i in 1:nrow(testtab)) {

counter <- (counter + 1)

s1 <- paste(s1, testtab$wCloud[counter], testtab$Plot[counter], sep = " ")

}

s1 <- gsub("film", "", s1)

s1 <- gsub("movie", "", s1)

words.vec <- VectorSource(s1)

words.corpus <- Corpus(words.vec)

words.corpus <- tm\_map(words.corpus, content\_transformer(tolower))

words.corpus <- tm\_map(words.corpus, removePunctuation)

words.corpus <- tm\_map(words.corpus, removeNumbers)

words.corpus <- tm\_map(words.corpus, removeWords, stopwords("english"))

tdm <- TermDocumentMatrix(words.corpus)

m <- as.matrix(tdm)

wordCounts <- rowSums(m)

wordCounts <- sort(wordCounts, decreasing = TRUE)

X <- data.frame(word=names(wordCounts),freq=wordCounts)

cdf <- testtab$Country[1]

cdf <- as.character(cdf)

return(assign(cdf,X,envir=.GlobalEnv))

}

# Use function to process country files to create word cloud files

fWord(movies.Australia)

fWord(movies.Canada)

fWord(movies.China)

fWord(movies.Egypt)

fWord(`movies.Hong Kong`)

fWord(movies.India)

fWord(movies.Japan)

fWord(movies.Malaysia)

fWord(movies.Maldives)

fWord(movies.Philipines)

fWord(movies.Russia)

fWord(`movies.South Korea`)

fWord(movies.Turkey)

fWord(`movies.United Kingdom`)

fWord(`movies.United States`)

# Word Cloud Creators

wordcloud(Australia$word,Australia$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(Canada$word,Canada$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(China$word,China$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(Egypt$word,Egypt$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(`Hong Kong`$word,`Hong Kong`$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(India$word,India$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(Japan$word,Japan$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(Malaysia$word,Malaysia$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(Maldives$word,Maldives$freq, min.freq = 2, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(Philipines$word,Philipines$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(Russia$word,Russia$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(`South Korea`$word,`South Korea`$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

#Encoding Issue (UTF-8 char mishandled) on Turkey file - delete these bad rows, or run at your peril

#Next time, deal with encoding in the beginning!

Turkey <- Turkey[-23,]

Turkey <- Turkey[1:50,]

wordcloud(Turkey$word,Turkey$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(`United Kingdom`$word,`United Kingdom`$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

wordcloud(`United States`$word,`United States`$freq, min.freq = 3, max.words=50, rot.per=.4, colors = brewer.pal(9,"Set3"))

################################################

# GENRE FREQ LOOP & FUNCTION

#Prepare for Loop

sList <- character()

rowcount <- nrow(movies)

counter <- 0

#Genre Freq Loop & Function

gWord <- function(countryfile)

{

testtab <- countryfile

counter <- 0

s1 <- character()

for (i in 1:nrow(testtab)) {

counter <- (counter + 1)

s1 <- paste(s1, testtab$wCloud[counter], sep = " ")

}

s1 <- gsub("film", "", s1)

s1 <- gsub("movie", "", s1)

words.vec <- VectorSource(s1)

words.corpus <- Corpus(words.vec)

words.corpus <- tm\_map(words.corpus, content\_transformer(tolower))

words.corpus <- tm\_map(words.corpus, removePunctuation)

words.corpus <- tm\_map(words.corpus, removeNumbers)

words.corpus <- tm\_map(words.corpus, removeWords, stopwords("english"))

tdm <- TermDocumentMatrix(words.corpus)

m <- as.matrix(tdm)

wordCounts <- rowSums(m)

wordCounts <- sort(wordCounts, decreasing = TRUE)

X <- data.frame(word=names(wordCounts),freq=wordCounts)

cdf <- paste(testtab$Country[1],"G",sep="")

cdf <- as.character(cdf)

return(assign(cdf,X,envir=.GlobalEnv))

}

#Use Genre Freq Function to process country files

gWord(movies.Australia)

gWord(movies.Canada)

gWord(movies.China)

gWord(movies.Egypt)

gWord(`movies.Hong Kong`)

gWord(movies.India)

gWord(movies.Japan)

gWord(movies.Malaysia)

gWord(movies.Maldives)

gWord(movies.Philipines)

gWord(movies.Russia)

gWord(`movies.South Korea`)

gWord(movies.Turkey)

gWord(`movies.United Kingdom`)

gWord(`movies.United States`)

########################################

#CREATE TOP 10 Genres per Country

TAustralia<- AustraliaG[1:10,]

TCanada <- CanadaG[1:10,]

TChina <- ChinaG[1:10,]

TEgypt <- EgyptG[1:10,]

THK <- `Hong KongG`[1:10,]

TIndia <- IndiaG[1:10,]

TJapan <- JapanG[1:10,]

TMalaysia <- MalaysiaG[1:10,]

TMaldives <- MaldivesG[1:10,]

TPhilipines <- PhilipinesG[1:10,]

TRussia <- RussiaG[1:10,]

TSK <- `South KoreaG`[1:10,]

TTurkey <- TurkeyG[1:10,]

TUK <- `United KingdomG`[1:10,]

TUS <- `United StatesG`[1:10,]

# Export 10 Ten for treatment in Excel, next time handle in R, but out of time on this project!

Z <- list("Australia"=TAustralia, "Canada"=TCanada, "China"=TChina, "Egypt"=TEgypt,"Hong Kong"= THK, "India"=TIndia, "Japan"=TJapan, "Malaysia"=TMalaysia, "Maldives"=TMaldives, "Philippines"=TPhilipines,"Russia"=TRussia,"South Korea"=TSK, "Turkey"=Turkey, "UK"=TUK, "US"=TUS)

write.xlsx(Z, file = "writeXLSX3.xlsx")

########################################

#lLOAD COUNTRY DEMOS

CDemos <- read.csv("/Data/census\_data\_All.csv")

#Make a copy to manipulate

CDX <- CDemos

#2017 year only

CDX <- CDX[CDX$Year == 2017,]

row.names(CDX) <- 1:15

#Adding Metrics, I'm sure there is a better way to loop this,

#but MSWord manipulation made code creation quite easy

CountryStats$TopGenre <- c(as.String(TAustralia[1,1]), as.String(TCanada[1,1]), as.String(TChina[1,1]), as.String(TEgypt[1,1]), as.String(THK[1,1]), as.String(TIndia[1,1]), as.String(TJapan[1,1]), as.String(TMalaysia[1,1]), as.String(TMaldives[1,1]), as.String(TPhilipines[1,1]), as.String(TRussia[1,1]), as.String(TSK[1,1]), as.String(TTurkey[1,1]), as.String(TUK[1,1]), as.String(TUS[1,1]))

CountryStats$TGCounts<- c(as.String(TAustralia[1,2]), as.String(TCanada[1,2]), as.String(TChina[1,2]), as.String(TEgypt[1,2]), as.String(THK[1,2]), as.String(TIndia[1,2]), as.String(TJapan[1,2]), as.String(TMalaysia[1,2]), as.String(TMaldives[1,2]), as.String(TPhilipines[1,2]), as.String(TRussia[1,2]), as.String(TSK[1,2]), as.String(TTurkey[1,2]), as.String(TUK[1,2]), as.String(TUS[1,2]))

#Check dataframe

#str(CountryStats)

#convert

CountryStats$TGCounts <- as.integer(CountryStats$TGCounts)

#add percent metric

CountryStats$TGPercent <- (CountryStats$TGCounts/CountryStats$TotalMovies)

# Create table - Country Stats

formattable(CountryStats)

#merge tables to one mega table with movie & country stats

CXX <- merge(CountryStats,CDX)

# Creat table - Country Stats & Country Stats Mega Table

formattable(CXX, align = c("l", rep("r", NCOL(movieCounts) - 1)))

#Create table statistics

#summary(movies)

summary (CXX)

###########################################################

#MODELS, PLOTS & SCATTER PLOTS

ggplot(movies, aes(x=Release.Year, y=TotalScore, group=1))+geom\_line()+labs(x="Release Year", y="Sentiment Score", title="Movie Plot & Genre Sentiment Score per Movie", subtitle = "Movies plots are becoming more extreme for sentiment over time")

ggplot(CXX, aes(x=TotalMovies, y=CountryScore))+geom\_point(aes(size=Population, color=Area))

plot(CXX$CountryScore,CXX$Population)

plot(CXX$CountryScore,CXX$Density)

#Model

m.m <- lm(formula = CountryScore ~ Density, data = CXX)

plot(CXX$CountryScore,CXX$Density)

summary(m.m)

abline(m.m)