Team Member: Ziling Xu,(andrew id: zilingx) Lin Lyu Lyu,(andrew id: linly)

Part0 Introduction

We would use two datasets. The first one is called IMDB dataset and the second one is called Netflix dataset. The IMDB Movies Dataset contains information about 14,762 movies. Information about these movies was downloaded with wget for the purpose of creating a movie recommendation app. As for Netflix dataset, it consists of tv shows and movies available on Netflix as of 2019. The dataset is collected from Flixable which is a third-party Netflix search engine. In 2018, they released an interesting report which shows that the number of TV shows on Netflix has nearly tripled since 2010. The streaming service number of movies has decreased by more than 2000 titles since 2010, while its number of TV shows has nearly tripled. It will be interesting to explore what all other insights can be obtained from the same dataset. Integrating this dataset with other external datasets such as IMDB ratings, rotten tomatoes can also provide many interesting findings.

In this project, at first we would import these two datasets and then we would do someything about 'data cleaning'. We split IMDB dataset and Netflix dataset into two datasets repectively according to the column called 'type' whcih contain two categories, movie and tv.

Our second part is about Data Visualization. We compare the distribution of released movies and tvs for each of the datasets by creating and we compare the distribution of each of the genres by using both quantile plots and Q-Q plots. To be more detailed, we figure out the distribution of the genres in the IMDB dataset, the rating distribution of each of the genres and the rating count distributionin the IMDB dataset.

The third part is about data Model. We perform a `t.test` to check if the ratings of movies in 1980 and in 2000 could have come from the same distribution. And we also do a `t.test` to check if the ratings of Drama movies and Comedies could have come from the same distribution. For the second section of this part, we do regression and cross validation to fit a natural spline to number of reviews (popularity measure) as a function of rating. We also use ten-fold cross validation (so `K=10`) to fit a natural spline to nrOfUserReviews.

Import Library

```
set.seed(1)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
```

```
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(dplyr)
library(ggplot2)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(boot)
library(splines)
```

Import Datasets

```
imdb = read.csv(file = 'imdb.csv', header = TRUE)
netflix = read.csv(file = 'netflix.csv', header = TRUE)
```

Part1 Data Preparation

1.1 Data Overview

1.1.1. The IMDB dataset (https://www.kaggle.com/orgesleka/imdbmovies)

```
summary(imdb)
```

```
##                   fn               tid               title
##  0                 :  413            :  387            :  421
##  1                 :   16   0        :   41   Der Mann\\:   12
##  titles01/tt0012349:    1   1        :    1   0         :    8
##  titles01/tt0015864:    1   tt0000005:    1   Hilfe\\   :    6
##  titles01/tt0017136:    1   tt0000248:    1   Liebling\\:    4
##  titles01/tt0017925:    1   tt0000306:    1   Arielle\\ :    3
##  (Other)           :14757   (Other)  :14758   (Other)   :14736
##                                            wordsInTitle
##                                                 :  441
##  the tonight show with jay leno episode tv episode:   91
##  jeopardy episode tv episode                     :   31
##  troldspejlet episode tv episode                 :   26
##  the th annual academy awards                    :   23
##  chelsea lately episode tv episode               :   20
##  (Other)                                         :14558
##                                                                      url
##                                                                        :  429
##  der mann der zuviel wu te                                             :    2
##  der tag an dem die erde stillstand                                    :    2
##  the nostalgia chick lord of the rings return of the king part tv episode:    2
##   'n' Eddy (TV Series 1999-2009)                                       :    1
##   1080 Bruxelles (1975)                                                :    1
```

```
##  (Other)                                                                 :14753
##    imdbRating     ratingCount       duration          year
##         : 1608          : 1635          : 1460   2011   :  667
##  7.3    :  584   10     :   62   1800   :  902   2010   :  638
##  7.2    :  561   7      :   59   3600   :  614   2009   :  627
##  7.6    :  556   5      :   55   5400   :  429   2007   :  560
##  7.1    :  552   6      :   51   6000   :  295   2008   :  522
##  7.4    :  539   8      :   46   5820   :  282   2012   :  499
##  (Other):10790   (Other):13282   (Other):11208   (Other):11677
##             type          nrOfWins    nrOfNominations   nrOfPhotos
##  video.movie  :10731   0      :8317   0      :9263    0      :5104
##  video.episode: 1921   1      :1694   1      : 843    2      : 620
##  video.tv     : 1562   2      : 914   2      : 579    4      : 463
##               :  433   3      : 631   3      : 494    3      : 456
##  game         :  118           : 430          : 429           : 429
##  2006         :   18   4      : 413   4      : 417    5      : 355
##  (Other)      :  407   (Other):2791   (Other):3165    (Other):7763
##  nrOfNewsArticles nrOfUserReviews    nrOfGenre            Action
##  0      :6056     Min.   :   0.0   Min.   :   0.000   Min.   :   0.0000
##         : 429     1st Qu.:   3.0   1st Qu.:   2.000   1st Qu.:   0.0000
##  2      : 288     Median :  29.0   Median :   2.000   Median :   0.0000
##  3      : 218     Mean   : 103.6   Mean   :   4.493   Mean   :   0.5219
##  5      : 205     3rd Qu.: 102.0   3rd Qu.:   3.000   3rd Qu.:   0.0000
##  4      : 180     Max.   :4928.0   Max.   :1547.000   Max.   :1930.0000
##  (Other):7814     NA's   :429      NA's   :429        NA's   :429
##      Adult            Adventure       Animation        Biography
##  Min.   :  0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:  0.0000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :  0.0000   Median :0.000   Median :0.0000   Median :0.0000
##  Mean   :  0.0601   Mean   :0.121   Mean   :0.0712   Mean   :0.0431
##  3rd Qu.:  0.0000   3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :488.0000   Max.   :3.000   Max.   :1.0000   Max.   :1.0000
##  NA's   :429        NA's   :429     NA's   :429      NA's   :429
##      Comedy           Crime         Documentary         Drama
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.3512   Mean   :0.1464   Mean   :0.0818   Mean   :0.4079
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :429      NA's   :429      NA's   :429      NA's   :429
##      Family          Fantasy          FilmNoir        GameShow
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.0813   Mean   :0.0593   Mean   :0.0146   Mean   :0.0127
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :429      NA's   :429      NA's   :429      NA's   :429
##     History           Horror           Music           Musical
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.000
##  Mean   :0.0388   Mean   :0.0698   Mean   :0.0381   Mean   :0.026
```

```
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000
##  NA's   :429      NA's   :429      NA's   :429      NA's   :429
##     Mystery            News         RealityTV         Romance
##  Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.000   Median :0.0000   Median :0.0000
##  Mean   :0.0656   Mean   :0.012   Mean   :0.0087   Mean   :0.1226
##  3rd Qu.:0.0000   3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.000   Max.   :1.0000   Max.   :1.0000
##  NA's   :429      NA's   :429     NA's   :429      NA's   :429
##      SciFi            Short            Sport           TalkShow
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.0701   Mean   :0.0382   Mean   :0.0165   Mean   :0.0358
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :429      NA's   :429      NA's   :429      NA's   :429
##     Thriller           War            Western
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.0886   Mean   :0.0325   Mean   :0.0224
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :429      NA's   :429      NA's   :429
```

1.1.2. The Netflix dataset (https://www.kaggle.com/shivamb/netflix-shows)

```
summary(netflix)
```

```
##     show_id              type              title
##  Min.   :  247747   Movie  :4265   Limitless  :   3
##  1st Qu.:80035802   TV Show:1969   Love       :   3
##  Median :80163367                  Oh My Ghost:   3
##  Mean   :76703679                  The Silence:   3
##  3rd Qu.:80244889                  Tunnel     :   3
##  Max.   :81235729                  Aquarius   :   2
##                                    (Other)    :6217
##                    director                     cast
##                        :1969                      : 570
##  Raúl Campos, Jan Suter:  18   David Attenborough:  18
##  Marcus Raboy          :  14   Samuel West       :  10
##  Jay Karas             :  13   Jeff Dunham       :   7
##  Jay Chapman           :  12   Craig Sechler     :   6
##  Martin Scorsese       :   9   Bill Burr         :   5
##  (Other)               :4199   (Other)           :5618
##            country                date_added    release_year
##  United States :2032   January 1, 2020  : 122   Min.   :1925
##  India         : 777   November 1, 2019 :  94   1st Qu.:2013
##                : 476   March 1, 2018    :  78   Median :2016
##  United Kingdom: 348   December 31, 2019:  74   Mean   :2013
##  Japan         : 176   October 1, 2018  :  72   3rd Qu.:2018
##  Canada        : 141   October 1, 2019  :  71   Max.   :2020
```

```
##  (Other)       :2284   (Other)          :5723
##      rating          duration
##  TV-MA  :2027   1 Season :1321
##  TV-14  :1698   2 Seasons: 304
##  TV-PG  : 701   3 Seasons: 158
##  R      : 508   90 min   : 111
##  PG-13  : 286   91 min   : 104
##  NR     : 218   92 min   : 101
##  (Other): 796   (Other)  :4135
##                                                 listed_in
##  Documentaries                                       : 299
##  Stand-Up Comedy                                     : 273
##  Dramas, International Movies                        : 248
##  Dramas, Independent Movies, International Movies: 186
##  Comedies, Dramas, International Movies              : 174
##  Kids' TV                                            : 159
##  (Other)                                             :4895
##
##  A surly septuagenarian gets another chance at her 20s after having her photo snapped at a studio tha
##  A ruthless businessman's mission to expose electoral fraud brings him into a heated and dangerous po
##  A young Han Solo tries to settle an old score with the help of his new buddy Chewbacca, a crew of sp
##  An affable, newly appointed college warden proves to be no ordinary man when an old enemy resurfaces
##  An aspiring musician battles age-old caste divides to be able to learn the art of a classical instru
##  As a series of murders hit close to home, a video game designer with post-traumatic stress must con
##  (Other)
```

1.In the IMDB dataset,

1.2 Data Preprocess

1.2.1.data cleaning

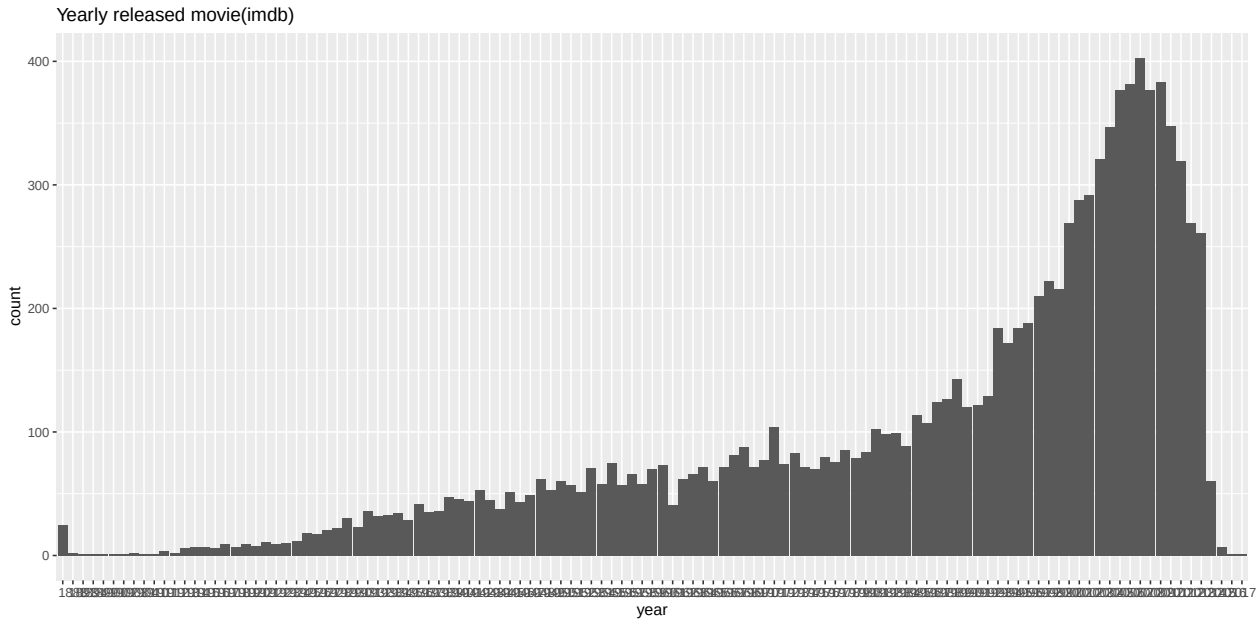
```
imdb <- filter(imdb,
       type =="video.movie")
imdb <- transform(imdb,
                  type = mapvalues(type, c("video.movie"), c("movie")))
imdb <- transform(imdb,
                  imdbRating = as.numeric(imdbRating)/10.0,
                  ratingCount = as.integer(ratingCount),
                  duration = as.integer(duration),
                  nrOfWins = as.integer(nrOfWins),
                  nrOfNominations = as.integer(nrOfNominations),
                  nrOfPhotos = as.integer(nrOfPhotos),
                  nrOfNewsArticles = as.integer(nrOfNewsArticles),
                  nrOfUserReviews = as.integer(nrOfUserReviews),
                  nrOfGenre = as.integer(nrOfGenre))
imdb <-filter(imdb,imdbRating!="")
netflix <- filter(netflix,
       type =="Movie")
netflix <- transform(netflix,
                     type = mapvalues(type, c("Movie"), c("movie")))
```

Part2 Data Visualization

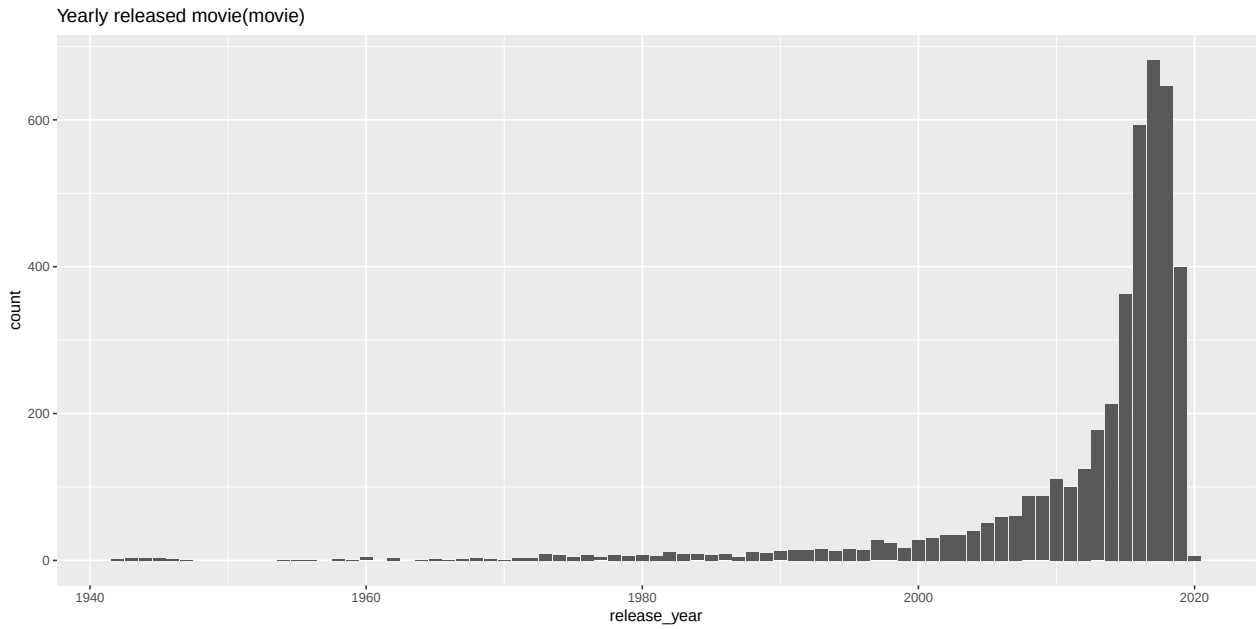
2.1 Visualising distributions

Compare the distribution of released movies and tvs for each of the datasets

```
ggplot(data = imdb) +
  geom_bar(mapping = aes(x = year))+
  labs(title = 'Yearly released movie(imdb)')
```




```
ggplot(data = netflix) +
  geom_bar(mapping = aes(x = release_year))+
  labs(title = 'Yearly released movie(movie)')
```



Compare the distribution of each of the genres

IMDB Dataset

```
colnames(imdb)
```

```
##  [1] "fn"               "tid"              "title"
##  [4] "wordsInTitle"     "url"              "imdbRating"
##  [7] "ratingCount"      "duration"         "year"
## [10] "type"             "nrOfWins"         "nrOfNominations"
## [13] "nrOfPhotos"       "nrOfNewsArticles" "nrOfUserReviews"
## [16] "nrOfGenre"        "Action"           "Adult"
## [19] "Adventure"        "Animation"        "Biography"
## [22] "Comedy"           "Crime"            "Documentary"
## [25] "Drama"            "Family"           "Fantasy"
## [28] "FilmNoir"         "GameShow"         "History"
## [31] "Horror"           "Music"            "Musical"
## [34] "Mystery"          "News"             "RealityTV"
## [37] "Romance"          "SciFi"            "Short"
## [40] "Sport"            "TalkShow"         "Thriller"
## [43] "War"              "Western"
```

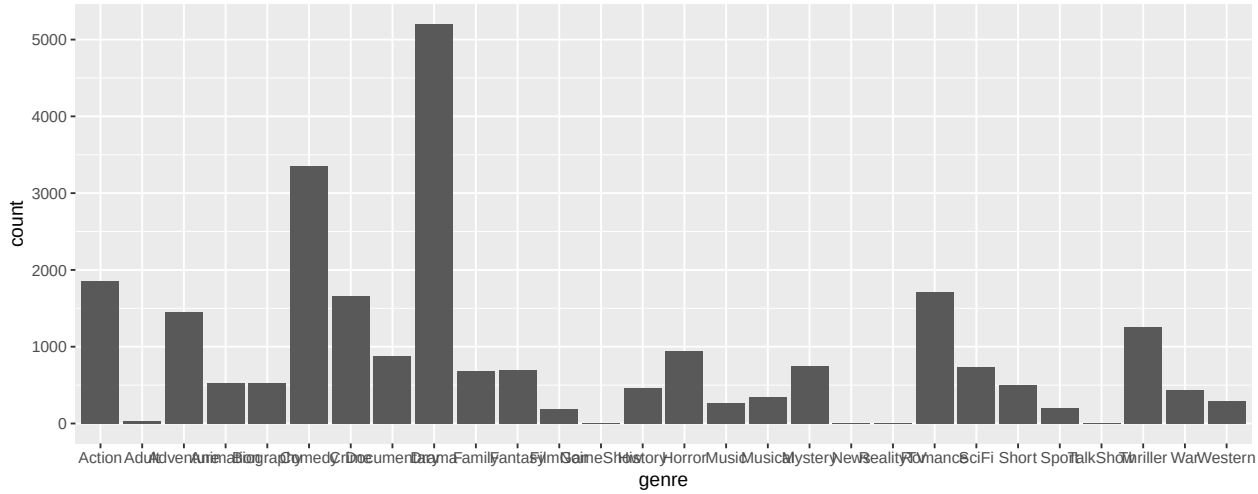
There are several genres int the IMDB Dataset: Action,Adventure,Animation,Biography,Comedy,Crime,Documentary,Drama,Fa
And it is possible that a movie belongs to more than one genre,so we need to include it in both distributions.

```
Action.imdb = imdb %>% filter(imdb[,"Action"]==1)%>% mutate(genre="Action")
Adult.imdb = imdb %>% filter(imdb[,"Adult"]==1)%>% mutate(genre="Adult")
Adventure.imdb = imdb %>% filter(imdb[,"Adventure"]==1)%>% mutate(genre="Adventure")
Animation.imdb = imdb %>% filter(imdb[,"Animation"]==1)%>% mutate(genre="Animation")
Biography.imdb = imdb %>% filter(imdb[,"Biography"]==1)%>% mutate(genre="Biography")
Comedy.imdb = imdb %>% filter(imdb[,"Comedy"]==1)%>% mutate(genre="Comedy")
Crime.imdb = imdb %>% filter(imdb[,"Crime"]==1)%>% mutate(genre="Crime")
Documentary.imdb = imdb %>% filter(imdb[,"Documentary"]==1)%>% mutate(genre="Documentary")
Drama.imdb = imdb %>% filter(imdb[,"Drama"]==1)%>% mutate(genre="Drama")
Family.imdb = imdb %>% filter(imdb[,"Family"]==1)%>% mutate(genre="Family")
Fantasy.imdb = imdb %>% filter(imdb[,"Fantasy"]==1)%>% mutate(genre="Fantasy")
FilmNoir.imdb = imdb %>% filter(imdb[,"FilmNoir"]==1)%>% mutate(genre="FilmNoir")
GameShow.imdb = imdb %>% filter(imdb[,"GameShow"]==1)%>% mutate(genre="GameShow")
History.imdb= imdb %>% filter(imdb[,"History"]==1)%>% mutate(genre="History")
Horror.imdb = imdb %>% filter(imdb[,"Horror"]==1)%>% mutate(genre="Horror")
Music.imdb = imdb %>% filter(imdb[,"Music"]==1)%>% mutate(genre="Music")
Musical.imdb = imdb %>% filter(imdb[,"Musical"]==1)%>% mutate(genre="Musical")
Mystery.imdb = imdb %>% filter(imdb[,"Mystery"]==1)%>% mutate(genre="Mystery")
News.imdb = imdb %>% filter(imdb[,"News"]==1)%>% mutate(genre="News")
RealityTV.imdb =imdb %>% filter(imdb[,"RealityTV"]==1)%>% mutate(genre="RealityTV")
Romance.imdb = imdb %>% filter(imdb[,"Romance"]==1)%>% mutate(genre="Romance")
SciFi.imdb = imdb %>% filter(imdb[,"SciFi"]==1)%>% mutate(genre="SciFi")
Short.imdb = imdb %>% filter(imdb[,"Short"]==1)%>% mutate(genre="Short")
Sport.imdb = imdb %>% filter(imdb[,"Sport"]==1)%>% mutate(genre="Sport")
TalkShow.imdb = imdb %>% filter(imdb[,"TalkShow"]==1)%>% mutate(genre="TalkShow")
Thriller.imdb = imdb %>% filter(imdb[,"Thriller"]==1)%>% mutate(genre="Thriller")
War.imdb = imdb %>% filter(imdb[,"War"]==1)%>% mutate(genre="War")
Western.imdb = imdb %>% filter(imdb[,"Western"]==1)%>% mutate(genre="Western")
imdb.genres <- rbind(Action.imdb,Adult.imdb,Adventure.imdb,Animation.imdb,
                     Biography.imdb,Comedy.imdb,Crime.imdb,Documentary.imdb,
                     Drama.imdb,Family.imdb,Fantasy.imdb,FilmNoir.imdb,
                     GameShow.imdb,History.imdb,Horror.imdb,Music.imdb,
                     Musical.imdb,Mystery.imdb,News.imdb,RealityTV.imdb,
```

```
                      Romance.imdb,SciFi.imdb,Short.imdb,Sport.imdb,
                      TalkShow.imdb,Thriller.imdb,War.imdb,Western.imdb)
imdb.genres <- select(imdb.genres,genre,imdbRating,ratingCount,duration,year,nrOfWins,nrOfNominations,n
nrOfNewsArticles,nrOfUserReviews)
```

Distribution of the genres in the IMDB dataset

```
ggplot(data = imdb.genres) +
  geom_bar(mapping = aes(x = genre))
```




```
ddply(imdb.genres, "genre", summarize, n.count = length(genre))
```

```
##          genre n.count
## 1       Action    1856
## 2        Adult      28
## 3    Adventure    1452
## 4    Animation     531
## 5    Biography     533
## 6       Comedy    3359
## 7        Crime    1655
## 8  Documentary     876
## 9        Drama    5201
## 10      Family     683
## 11     Fantasy     694
## 12    FilmNoir     189
## 13    GameShow       3
## 14     History     456
## 15      Horror     939
## 16       Music     263
## 17     Musical     348
## 18     Mystery     743
## 19        News       2
## 20   RealityTV       3
## 21     Romance    1713
## 22       SciFi     733
## 23       Short     506
## 24       Sport     205
## 25    TalkShow       8
## 26    Thriller    1252
```

```
## 27          War     434
## 28      Western     291
```

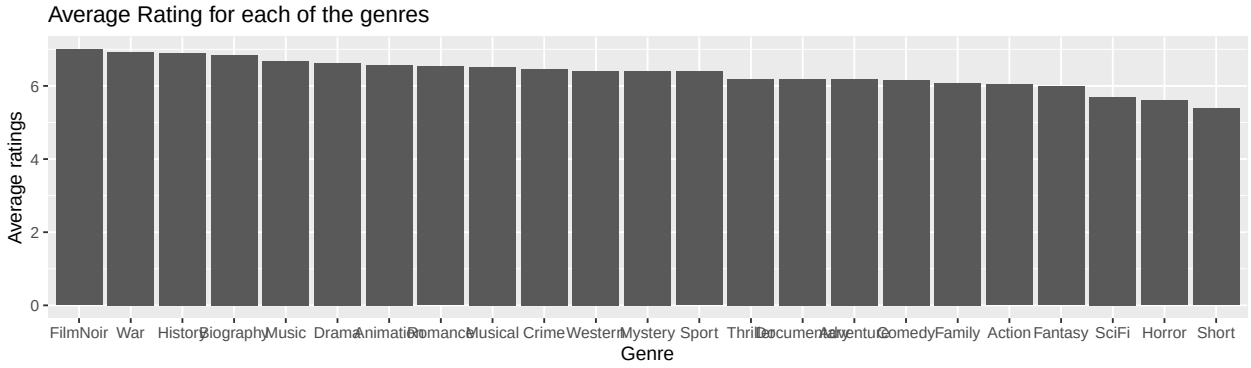
From the plot above, in the IMDB datatset, we can see 'Drama' genre appears the most times, 'Gameshow', 'News' and 'Reality TV' has the lowest count. Noticing that movies that belong to "Adult","GameShow","News","RealityTV", and "TalkShow" genre is extremely small, and it is hard to get any meaninful result from such small amount of data points. We can simply exclude them from the analysis below.

```
imdb.genres <- filter(imdb.genres,genre!="Adult",genre!="News",genre!="GameShow",genre!="TalkShow",genre
imdb<-imdb %>%
  filter(Adult!=1|(Adult==1&nrOfGenre>1)) %>%
  filter(News!=1|(News==1&nrOfGenre>1)) %>%
  filter(GameShow!=1|(GameShow==1&nrOfGenre>1)) %>%
  filter(TalkShow!=1|(TalkShow==1&nrOfGenre>1)) %>%
  filter(RealityTV!=1|(RealityTV==1&nrOfGenre>1))
```

```
imdb.statistics<-ddply(imdb.genres, "genre", summarize,
      n.count = length(genre),
      avg.rating = mean(imdbRating),
      avg.ratingCount = mean(ratingCount),
      avg.reviews = mean(nrOfUserReviews),
      avg.nominations = mean(nrOfNominations))
```

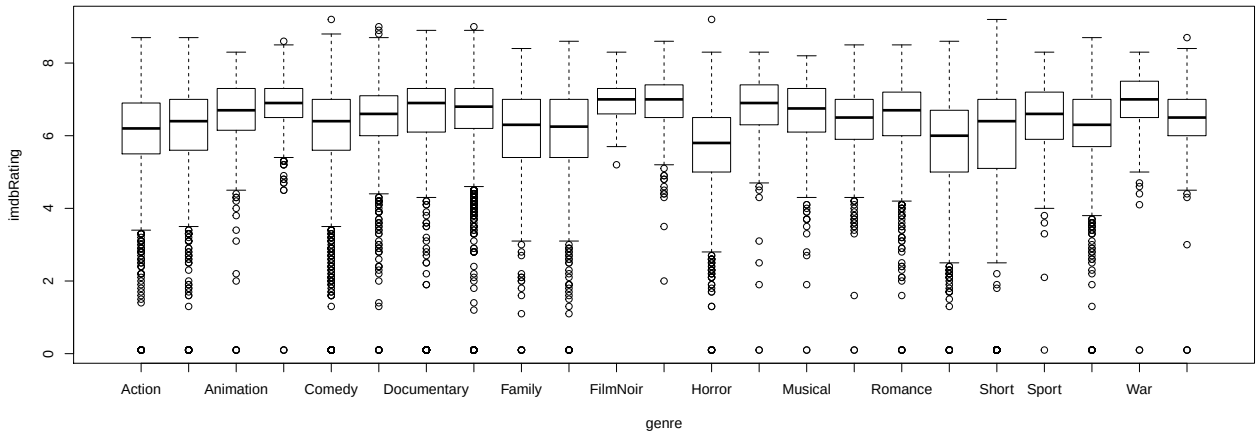
Rating distribution of each of the genres in the IMDB dataset

```
ggplot(data = imdb.statistics, mapping=aes(x = reorder(genre, desc(avg.rating)), y = avg.rating)) +
  geom_bar(stat='identity')+
  labs(title='Average Rating for each of the genres',
       x = 'Genre', y = 'Average ratings')
```



From the plot above, we can see movies under 'FilmNoir' genre have the highest average rating, movies under 'Short' genre have the highest average rating.

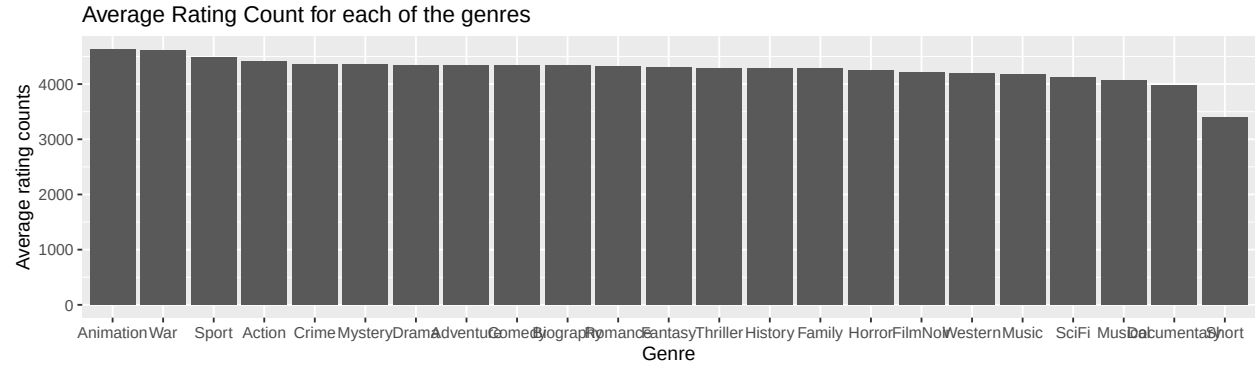
```
boxplot(imdbRating~genre,data = imdb.genres)
```



From the plot above, we can see that almost each genre has nearly the same maximum value and almost the same minimum value and their medium is almost the same

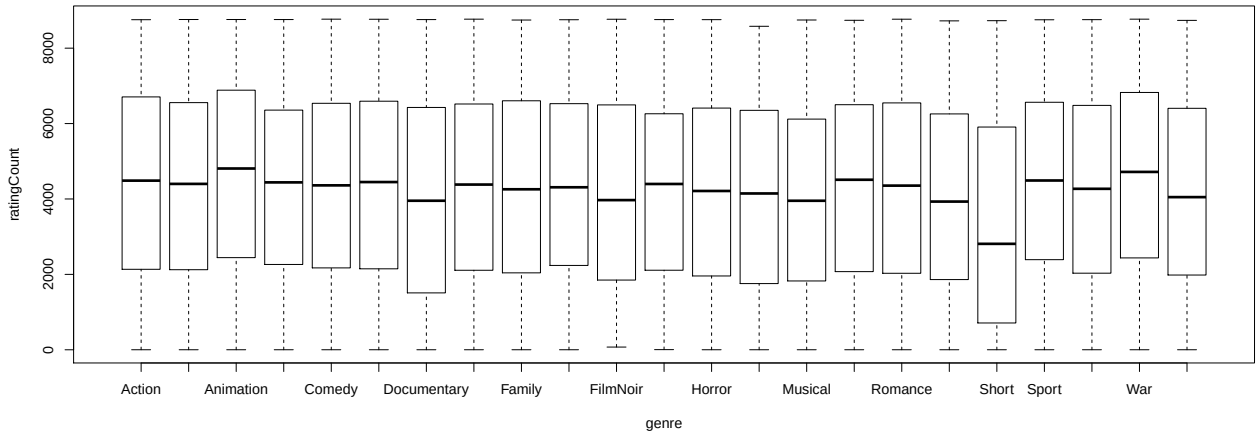
Rating count distribution of each of the genres in the IMDB dataset

```
ggplot(data = imdb.statistics, mapping=aes(x = reorder(genre, desc(avg.ratingCount)), y = avg.ratingCou
  geom_bar(stat='identity')+
  labs(title='Average Rating Count for each of the genres',
       x = 'Genre', y = 'Average rating counts')
```



From the plot above, we can see movies under 'Animation' genre have the highest average rating count, movies under 'Short' genre have the highest average rating count.

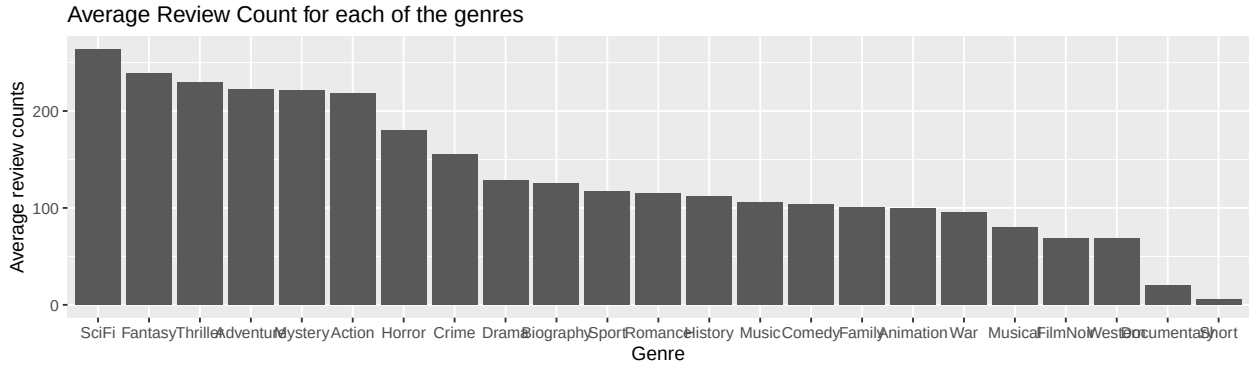
```
boxplot(ratingCount~genre,data = imdb.genres)
```



From the plot above, we can see that almost each genre has nearly the same maximum value and almost the same minimum value and their medium is almost the same

Review count distribution of each of the genres in the IMDB dataset

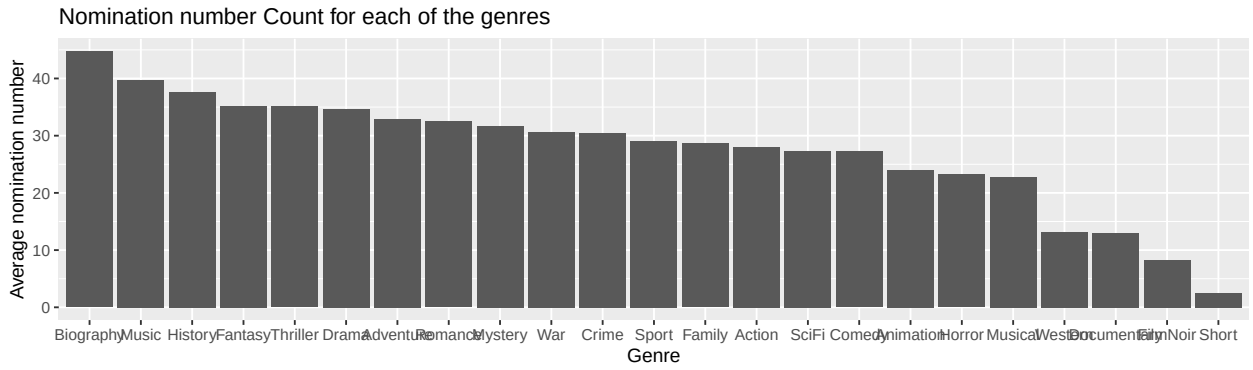
```
ggplot(data = imdb.statistics, mapping=aes(x = reorder(genre, desc(avg.reviews)), y = avg.reviews)) +
  geom_bar(stat='identity')+
  labs(title='Average Review Count for each of the genres',
       x = 'Genre', y = 'Average review counts')
```



From the plot above, we can see movies under 'SciFi' genre have the highest average review count, movies under 'Short' genre have the highest average review count.

Nomination number distribution of each of the genres in the IMDB dataset

```
ggplot(data = imdb.statistics, mapping=aes(x = reorder(genre, desc(avg.nominations)), y = avg.nominatio
  geom_bar(stat='identity')+
  labs(title='Nomination number Count for each of the genres',
       x = 'Genre', y = 'Average nomination number')
```



From the plot above, we can see movies under 'Biology' genre have the highest average nomination number, movies under 'Short' genre have the highest average nomination number.

Netflix Dataset

```
genres <- vector()
for (i in unique(netflix$listed_in)){
  if (length(str_split(i,",")[[1]])<2){
    genres <- c(genres , i)
  } else {
    for (j in str_split(i,",")[[1]]){
    genres  <- c(genres, str_trim(j))
```

```
    }
  }
}
genres <- unique(genres)
genres
```

```
##  [1] "Children & Family Movies" "Comedies"
##  [3] "Stand-Up Comedy"          "International Movies"
##  [5] "Sci-Fi & Fantasy"         "Thrillers"
##  [7] "Action & Adventure"       "Dramas"
##  [9] "Cult Movies"              "Independent Movies"
## [11] "Romantic Movies"          "Documentaries"
## [13] "Horror Movies"            "Music & Musicals"
## [15] "Anime Features"           "Faith & Spirituality"
## [17] "LGBTQ Movies"             "Movies"
## [19] "Classic Movies"           "Sports Movies"
```

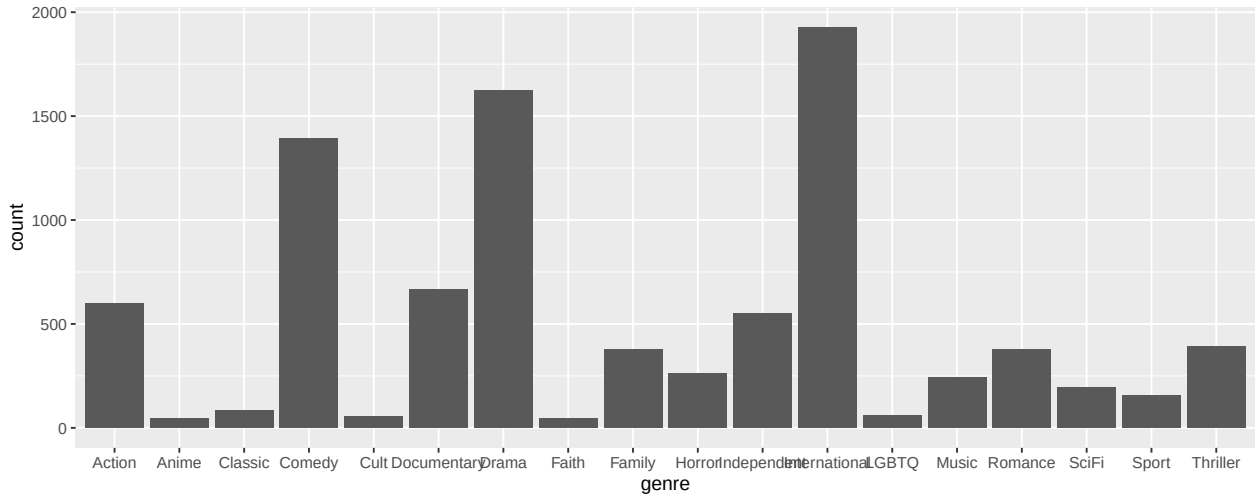
There are several genres int the Netflix Dataset: Family,Comedy,International,SciFi,Thriller,Action,Drama,Cult,Independent,Ro
And it is possible that a movie belongs to more than one genre, so we need to include it in both distributions.

```
Family.netflix<- netflix %>%
  filter(str_detect(listed_in,"Children & Family Movies"))%>%
  mutate(genre="Family")
Comedy.netflix<- netflix %>%
  filter(str_detect(listed_in,"Comedies")|str_detect(listed_in,"Stand-Up Comedy"))%>%
  mutate(genre="Comedy")
International.netflix<-netflix %>%
  filter(str_detect(listed_in,"International Movies"))%>%
  mutate(genre="International")
SciFi.netflix<-netflix %>%
  filter(str_detect(listed_in,"Sci-Fi & Fantasy"))%>%
  mutate(genre="SciFi")
Thriller.netflix<-netflix %>%
  filter(str_detect(listed_in,"Thrillers"))%>%
  mutate(genre="Thriller")
Action.netflix <-netflix %>%
  filter(str_detect(listed_in,"Action & Adventure" ))%>%
  mutate(genre="Action")
Drama.netflix <-netflix %>%
  filter(str_detect(listed_in, "Dramas"))%>%
  mutate(genre="Drama")
Cult.netflix <-netflix %>%
  filter(str_detect(listed_in, "Cult Movies"))%>%
  mutate(genre="Cult")
Independent.netflix <-netflix %>%
  filter(str_detect(listed_in, "Independent Movies"))%>%
  mutate(genre="Independent")
Romance.netflix <-netflix %>%
  filter(str_detect(listed_in, "Romantic Movies"))%>%
  mutate(genre="Romance")
Documentary.netflix <-netflix %>%
  filter(str_detect(listed_in, "Documentaries"))%>%
  mutate(genre="Documentary")
Horror.netflix <-netflix %>%
  filter(str_detect(listed_in, "Horror Movies"))%>%
```

```
  mutate(genre="Horror")
Music.netflix <-netflix %>%
  filter(str_detect(listed_in, "Music & Musicals"))%>%
  mutate(genre="Music")
Anime.netflix <-netflix %>%
  filter(str_detect(listed_in, "Anime Features"))%>%
  mutate(genre="Anime")
Faith.netflix <-netflix %>%
  filter(str_detect(listed_in, "Faith & Spirituality"))%>%
  mutate(genre="Faith")
LGBTQ.netflix <-netflix %>%
  filter(str_detect(listed_in, "LGBTQ Movies"))%>%
  mutate(genre="LGBTQ")
Classic.netflix <-netflix %>%
  filter(str_detect(listed_in, "Classic Movies"))%>%
  mutate(genre="Classic")
Sport.netflix <-netflix %>%
  filter(str_detect(listed_in, "Sports Movies"))%>%
  mutate(genre="Sport")

netflix.genres <- rbind(Family.netflix,Comedy.netflix,International.netflix,SciFi.netflix,
                        Thriller.netflix,Action.netflix,Drama.netflix,Cult.netflix,
                        Independent.netflix,Romance.netflix,Documentary.netflix,
                        Horror.netflix,Music.netflix,Anime.netflix,Faith.netflix,
                        LGBTQ.netflix,Classic.netflix,Sport.netflix)
```

```
ggplot(data = netflix.genres) +
  geom_bar(mapping = aes(x = genre))
```




```
ddply(netflix.genres, "genre", summarize, n.count = length(genre))
```

```
##            genre n.count
## 1         Action     597
## 2          Anime      45
## 3        Classic      84
## 4         Comedy    1394
## 5           Cult      55
## 6    Documentary     668
```

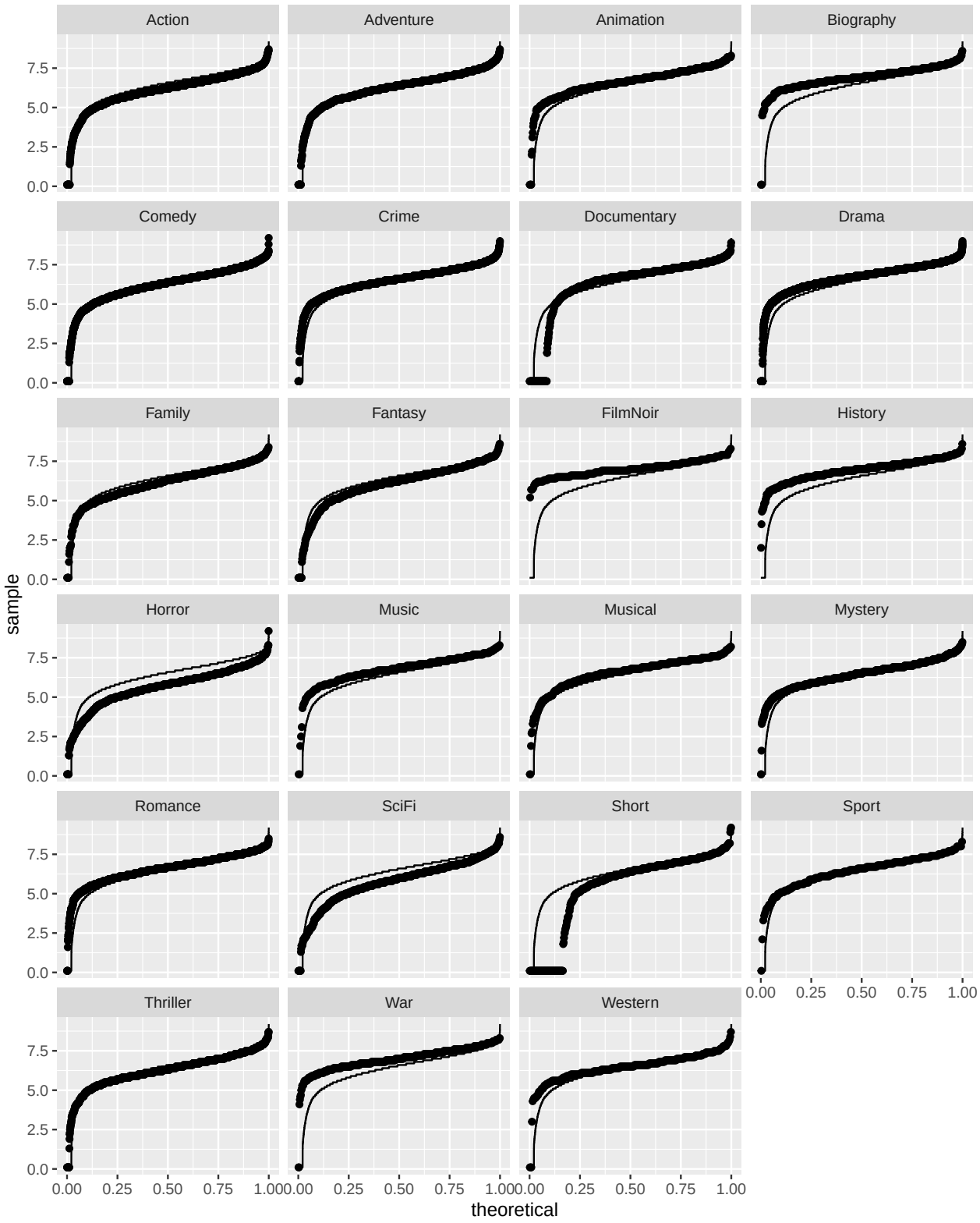
```
## 7          Drama    1623
## 8          Faith      47
## 9         Family     378
## 10        Horror     262
## 11   Independent     552
## 12 International    1927
## 13         LGBTQ      60
## 14         Music     243
## 15       Romance     376
## 16         SciFi     193
## 17         Sport     157
## 18      Thriller     392
```

From the plot above,in the Netflix dataset, despite 'International' genre, 'Drama' genre appears the most times, 'Anime', 'Cult' and 'Faith' has the lowest count.

Compare the distribution of ratings for each of the genres

Use both quantile plots and Q-Q plots to compare the distribution of ratings for each of the genres. For the quantile plot, we want to use facets to divide the data into groups, and show a common reference line (specifically, the distribution of the pooled data) in each facet to make visual comparisons easy. For the QQ plot we will similarly compare the quantiles of each group against the quantiles of the pooled data.

```
ggplot(data = imdb.genres, mapping = aes(sample = imdbRating)) +
  stat_qq(distribution='qunif') +
  facet_wrap('genre', nrow = 6)  +
  stat_qq(data = imdb,mapping = aes(sample = imdbRating), distribution =  qunif, geom='line')
```

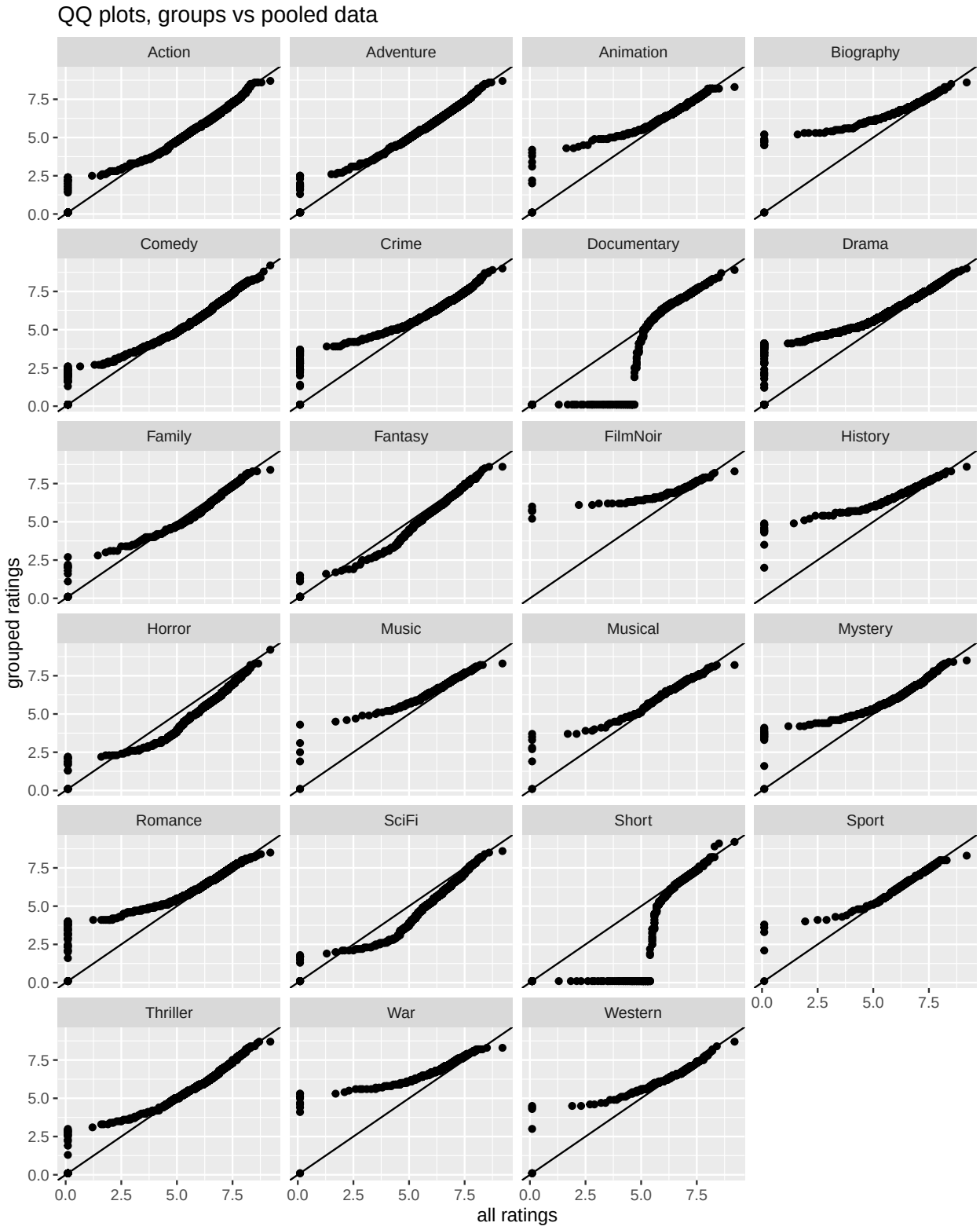



```
Find.QQ = function(genre.data,pooled.data,type) {
  probs = seq(from = 0, to = 1, length.out = nrow(genre.data))
  q1 = quantile(genre.data$imdbRating, probs= probs)
  q2 = quantile(pooled.data$imdbRating, probs=probs )
```

```
  return( data.frame(genre = type, group.data = q1, pooled.data = q2, quantile = probs))
}

imdb_qq.genres = rbind(Find.QQ(Action.imdb,imdb,"Action"),
                       Find.QQ(Adventure.imdb,imdb,"Adventure"),
                       Find.QQ(Animation.imdb,imdb,"Animation"),
                       Find.QQ(Biography.imdb,imdb,"Biography"),
                       Find.QQ(Comedy.imdb,imdb,"Comedy"),
                       Find.QQ(Crime.imdb,imdb,"Crime"),
                       Find.QQ(Documentary.imdb,imdb,"Documentary"),
                       Find.QQ(Drama.imdb,imdb,"Drama"),
                       Find.QQ(Family.imdb,imdb,"Family"),
                       Find.QQ(Fantasy.imdb,imdb,"Fantasy"),
                       Find.QQ(FilmNoir.imdb,imdb,"FilmNoir"),
                       Find.QQ(History.imdb,imdb,"History"),
                       Find.QQ(Horror.imdb,imdb,"Horror"),
                       Find.QQ(Music.imdb,imdb,"Music"),
                       Find.QQ(Musical.imdb,imdb,"Musical"),
                       Find.QQ(Mystery.imdb,imdb,"Mystery"),
                       Find.QQ(Romance.imdb,imdb,"Romance"),
                       Find.QQ(SciFi.imdb,imdb,"SciFi"),
                       Find.QQ(Short.imdb,imdb,"Short"),
                       Find.QQ(Sport.imdb,imdb,"Sport"),
                       Find.QQ(Thriller.imdb,imdb,"Thriller"),
                       Find.QQ(War.imdb,imdb,"War"),
                       Find.QQ(Western.imdb,imdb,"Western"))
```

```
ggplot(data = imdb_qq.genres, mapping=aes(x=pooled.data, y=group.data)) +
  geom_point() +
  facet_wrap('genre', nrow=6) +
  labs(title='QQ plots, groups vs pooled data',
       x = 'all ratings', y = 'grouped ratings') +
  geom_abline(slope=1)
```



The distribution of ratings for action movies have better ratings than the overall distribution at the lower quantiles, and also better than the overall distribution at the highest quantiles The distribution of ratings for adventure movies have better ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles The distribution of ratings for animation first have better

ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles The distribution of ratings for biology movies generally have better ratings than the overall distribution The distribution of ratings for comedy movies have better ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles The distribution of ratings for crime movies have better ratings than the overall distribution at the lower quantiles, and also better than the overall distribution at the highest quantiles The distribution of ratings for documentary movies have worse ratings than the overall distribution at the lower quantiles, and similar ratings as the overall distribution at the highest quantiles The distribution of ratings for drama movies generally have better ratings than the overall distribution The distribution of ratings for family movies have better ratings than the overall distribution at the lower quantiles, and worse ratings than the overall distribution at the middle quantiles The distribution of ratings for fantasy movies generally have worse ratings than the overall distribution The distribution of ratings for history movies have better ratings than the overall distribution at the lower quantiles, and similar ratings as the overall distribution at the highest quantiles The distribution of ratings for horror movies generally have worse ratings than the overall distribution The distribution of ratings for music movies generally have better ratings than the overall distribution The distribution of ratings for musical movies generally have better ratings than the overall distribution The distribution of ratings for mystery movies generally have better ratings than the overall distribution The distribution of ratings for romance movies have better ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles The distribution of ratings for scifi movies generally have worse ratings than the overall distribution The distribution of ratings for short movies generally have worse ratings than the overall distribution The distribution of ratings for sport movies have better ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles The distribution of ratings for thriller movies have better ratings than the overall distribution at the lower quantiles, and similar ratings as the overall distribution at the highest quantiles The distribution of ratings for war movies generally have better ratings than the overall distribution The distribution of ratings for western movies have better ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles

Part3 Data Model

3.1 T test

Do a `t.test` to check if the ratings of movies in 1980 and in 2000 could have come from the same distribution.

```
with(imdb,
     t.test(x = imdbRating[ year == 1980], y = imdbRating[ year == 2000]))
```

```
##
##  Welch Two Sample t-test
##
## data:  imdbRating[year == 1980] and imdbRating[year == 2000]
## t = 3.1737, df = 207.34, p-value = 0.001734
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1611940 0.6898504
## sample estimates:
## mean of x mean of y
##  6.527381  6.101859
```

Mean rating of movies in year 1980 is 6.527381 and mean rating of movies in year 2000 is 6.101859. The conclusion of the t-test: difference in ratings of movies in 1980 and in 2000 is statistically significant.

Do a `t.test` to check if the ratings of Drama movies and Comedies could have come from the same distribution.

```
with(imdb.genres,
     t.test(x = imdbRating[genre=="Drama"], y = imdbRating[genre=="Comedy"]))
```

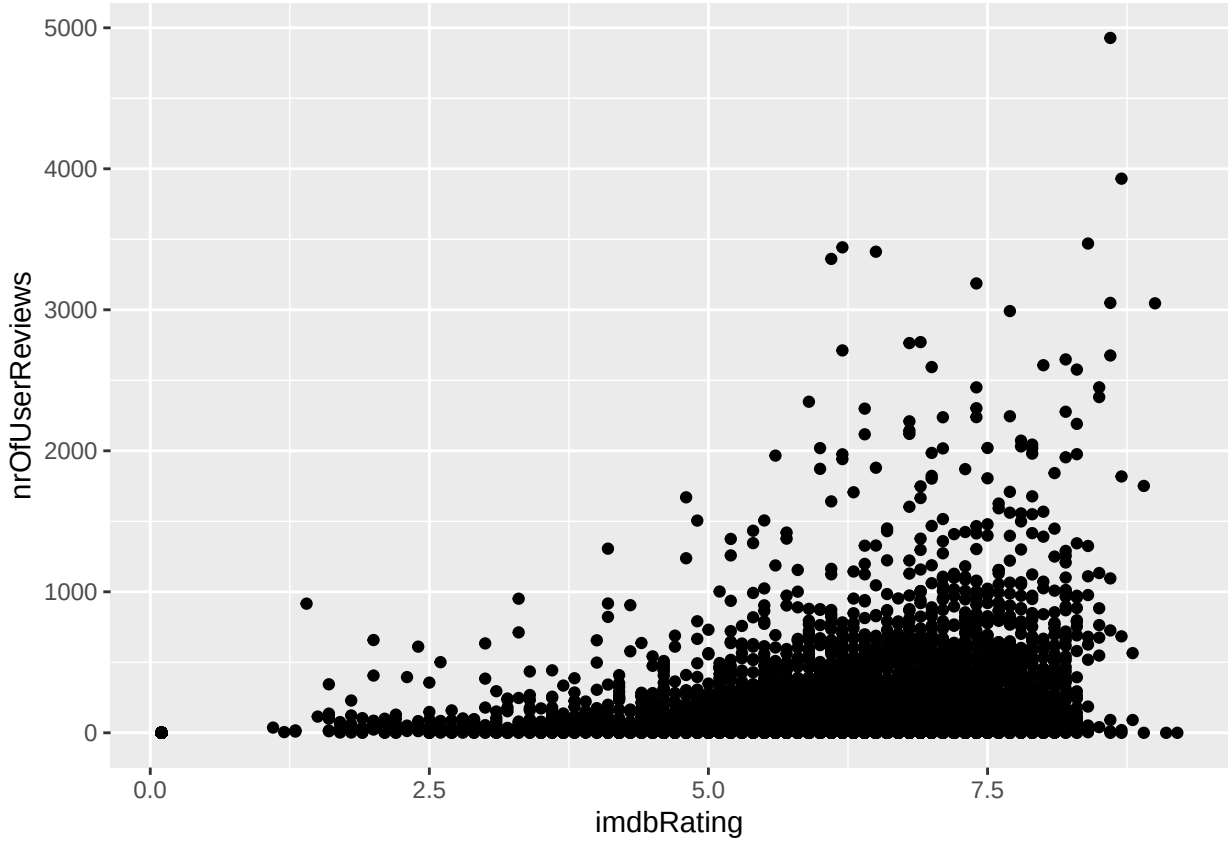
```
##
##  Welch Two Sample t-test
##
## data:  imdbRating[genre == "Drama"] and imdbRating[genre == "Comedy"]
## t = 17.597, df = 6326.8, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.4146939 0.5186708
## sample estimates:
## mean of x mean of y
##  6.633667  6.166984
```

Mean rating of Drama movies is 6.633667 and mean rating of Comedies is 6.166984. The conclusion of the t-test: difference in ratings of Drama movies and Comedies is statistically significant.

3.2 Regression and Cross Validation

fit a natural spline to number of reviews (popularity measure) as a function of rating 1. Make a scatterplot of `nrOfUserReviews` as a function of `imdbRating`.

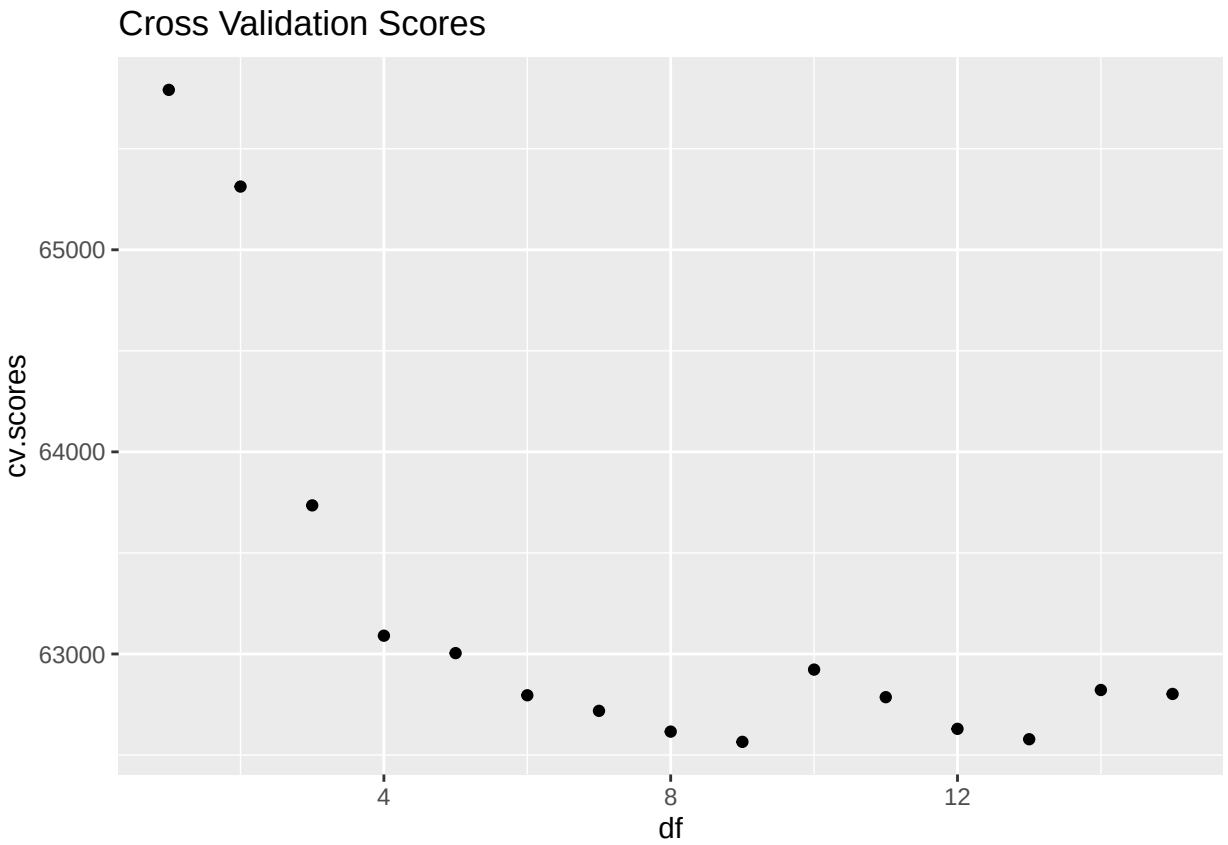
```
ggplot(data=imdb, mapping=aes(x=imdbRating, y=nrOfUserReviews)) + geom_point()
```



2.use ten-fold cross validation (so `K=10`) to fit a natural spline to nrOfUserReviews

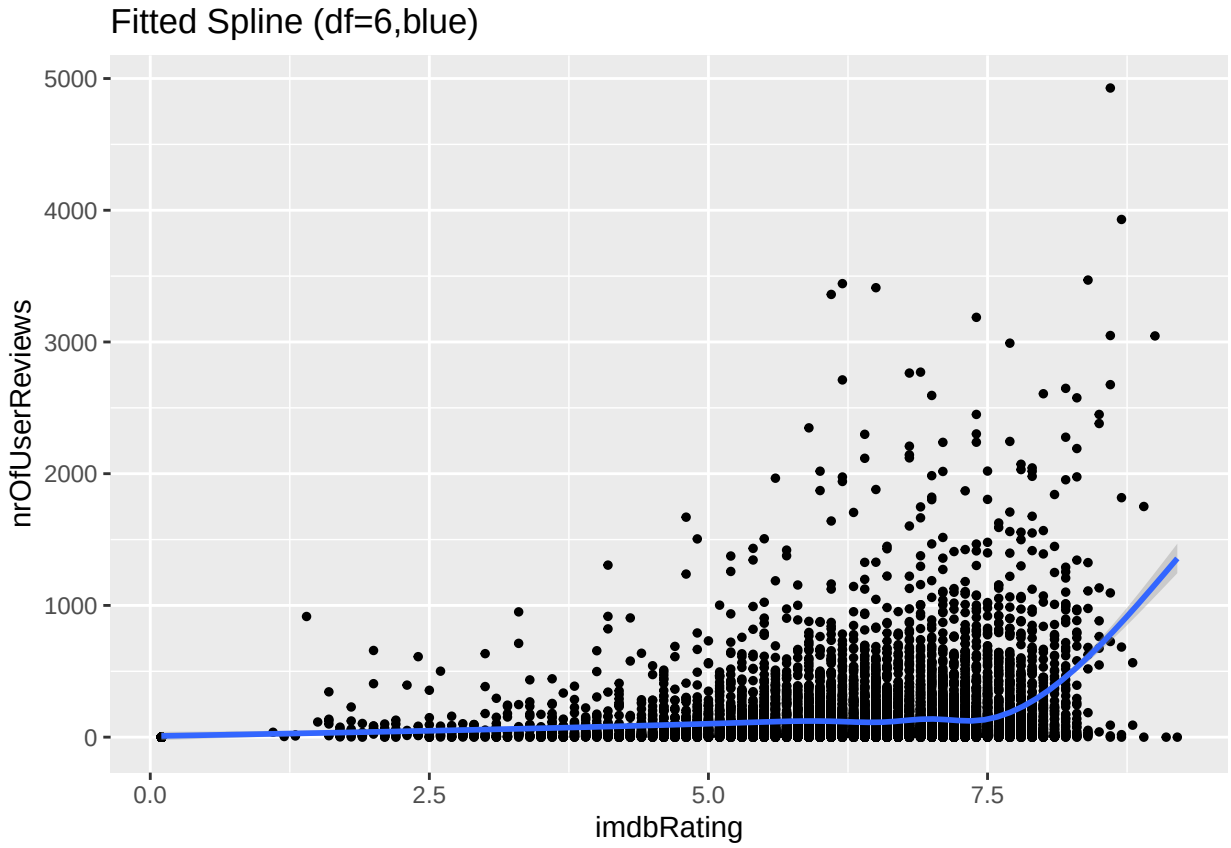
```
#  1. a plot of the cross-validation scores as a function of `df`
# cv.score will store all of our scores. We will initialize it to zero.
```

```
cv.scores = rep(0, times=15)
# vary DF from 1 to 10
for (DF in 1:15) {
  # fit the spline fit with df=DF, using glm
  spline.model = glm(nrOfUserReviews~ns(imdbRating, df=DF), data=imdb)
  # run fourfold cross validation
  cv = cv.glm(data=imdb, glmfit=spline.model, K=10)
  # extract the cross-validation score
  cv.scores[DF] = cv$delta[1]
}
# plot the cross validation score vs DF:
ggplot(mapping=aes(x=1:15, y=cv.scores)) + geom_point() +
  labs(x='df', title='Cross Validation Scores')
```



Since the cv score first reach the valley when df = 6, we will use df =6 in the below

```
# 2. a scatterplot showing `nrOfUserReviews` as a function of `imdbRating`, along with the fitted trend
ggplot(data=imdb, mapping=aes(x=imdbRating, y=nrOfUserReviews)) + geom_point(size=1) +
  geom_smooth(method='lm', formula = y ~ ns(x, df=6))  +
  labs(title = 'Fitted Spline (df=6,blue)')
```



Part4 Conclusion

Data Visualization

The distribution of the genres in the IMDB dataset

'Drama' genre appears the most times, 'Gameshow', 'News' and 'Reality TV' has the lowest count. Movies under 'FilmNoir' genre have the highest average rating, movies under 'Short' genre have the highest average rating. Movies under 'Animation' genre have the highest average rating count, movies under 'Short' genre have the highest average rating count. Movies under 'SciFi' genre have the highest average review count, movies under 'Short' genre have the highest average review count. Movies under 'Biology' genre have the highest average nomination number, movies under 'Short' genre have the highest average nomination number.

The distribution of the genres in the Netflix dataset

Despite 'International' genre, 'Drama' and 'Comedy 'genre appears the most times, 'Anime', 'Cult' and 'Faith' has the lowest count.

The distribution of ratings for each of the genres

The distribution of ratings for action movies have better ratings than the overall distribution at the lower quantiles, and also better than the overall distribution at the highest quantiles The distribution of ratings for adventure movies have better ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles The distribution of ratings for animation first have better ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles The distribution of ratings for biology movies generally have better ratings than the overall distribution The distribution of ratings for comedy movies have better ratings than the overall distribution

at the lower quantiles, but worse than the overall distribution at the highest quantiles The distribution of ratings for crime movies have better ratings than the overall distribution at the lower quantiles, and also better than the overall distribution at the highest quantiles The distribution of ratings for documentary movies have worse ratings than the overall distribution at the lower quantiles, and similar ratings as the overall distribution at the highest quantiles The distribution of ratings for drama movies generally have better ratings than the overall distribution The distribution of ratings for family movies have better ratings than the overall distribution at the lower quantiles, and worse ratings than the overall distribution at the middle quantiles The distribution of ratings for fantasy movies generally have worse ratings than the overall distribution The distribution of ratings for history movies have better ratings than the overall distribution at the lower quantiles, and similar ratings as the overall distribution at the highest quantiles The distribution of ratings for horror movies generally have worse ratings than the overall distribution The distribution of ratings for music movies generally have better ratings than the overall distribution The distribution of ratings for musical movies generally have better ratings than the overall distribution The distribution of ratings for mystery movies generally have better ratings than the overall distribution The distribution of ratings for romance movies have better ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles The distribution of ratings for scifi movies generally have worse ratings than the overall distribution The distribution of ratings for short movies generally have worse ratings than the overall distribution The distribution of ratings for sport movies have better ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles The distribution of ratings for thriller movies have better ratings than the overall distribution at the lower quantiles, and similar ratings as the overall distribution at the highest quantiles The distribution of ratings for war movies generally have better ratings than the overall distribution The distribution of ratings for western movies have better ratings than the overall distribution at the lower quantiles, but worse than the overall distribution at the highest quantiles

T test

- Mean rating of movies in year 1980 is 6.527381 and mean rating of movies in year 2000 is 6.101859. - The conclusion of the t-test: difference in ratings of movies in 1980 and in 2000 is statistically significant. - Mean rating of Drama movies is 6.633667 and mean rating of Comedies is 6.166984. - The conclusion of the t-test: difference in ratings of Drama movies and Comedies is statistically significant.