Final Project

#Executive Summary

Each row represents a movie available on FilmTV.it, with the original title, year, genre, duration, country, director, actors, average vote and votes. The fie in the English version contains 37,711 movies and 19 attributes.

(we should add what each of the different variables mean)

EDA

Data Cleaning

The first step in our data cleaning is to deal with missing values. First, there werre 88 movies that had no genre. So we used google to fill in the missing genres to ensure we could use those data points for our analysis. We also were able to find one movie that had no title, however, from the description we were able to determine the movie name as well as the genre.

Another thing we noticed when analyzing the data was that some of the genres overlapped. For instance, there was a level "Romantic" and "Romance". We decided that these are essentially the same thing and thus should be marked as the same.

Here we are making film into a factor as it contains categorical variables that would be better analyzed as factors.

```
##  [1] "Action"      "Adventure"   "Animation"   "Biblical"    "Biography"
##  [6] "Comedy"      "Crime"       "Documentary" "Drama"       "Fantasy"
## [11] "Gangster"    "Grotesque"   "History"     "Horror"      "Mélo"
## [16] "Musical"     "Mythology"   "Noir"        "Romance"     "Short Movie"
## [21] "Sperimental" "Sport"       "Spy"         "Super-hero"  "Thriller"
## [26] "War"         "Western"
```

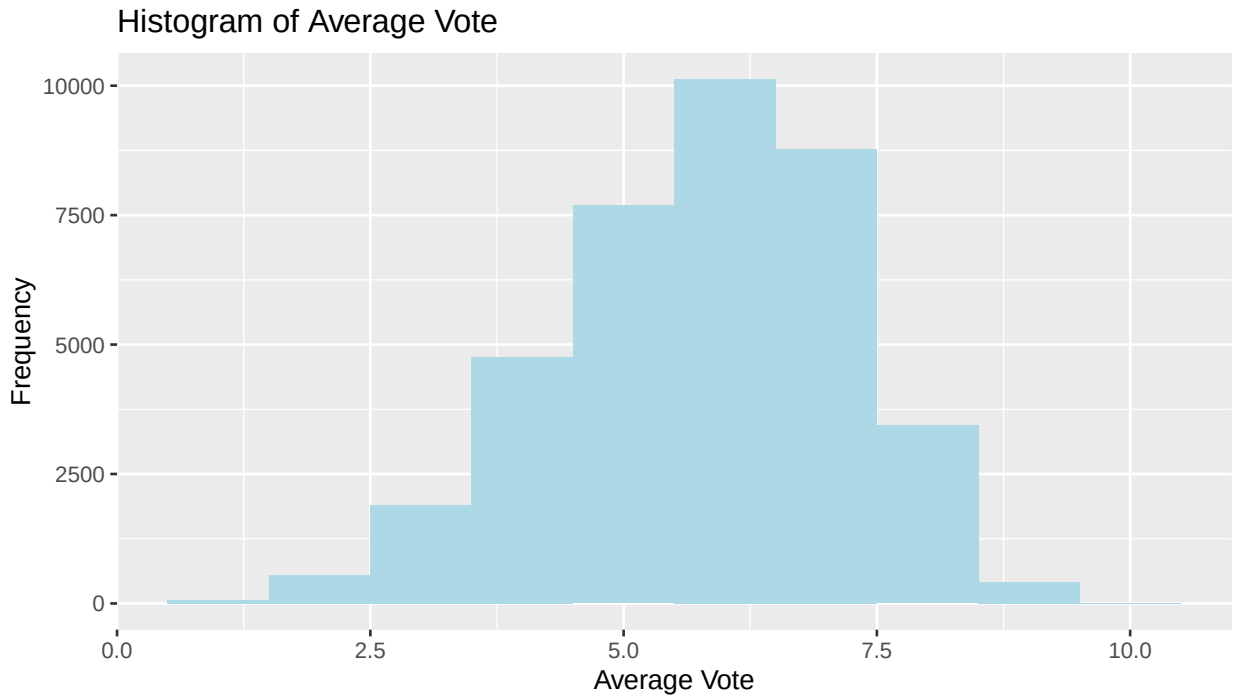
We will also make the year variable into different subsections including different years that will be used as a factor. We are going to make the years in sections of 10 years from 1897 to 2021, that will give us 14 different levels of year modulos to work with

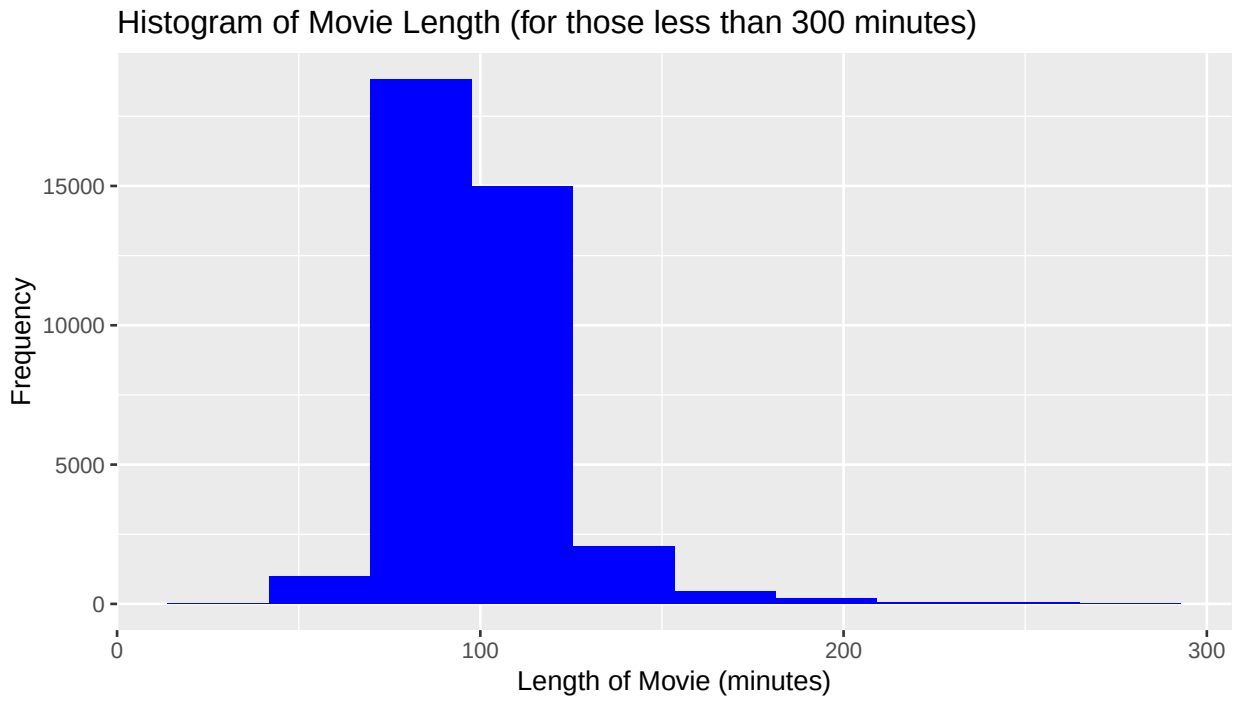
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1897    1975    2000    1992    2012    2021
```

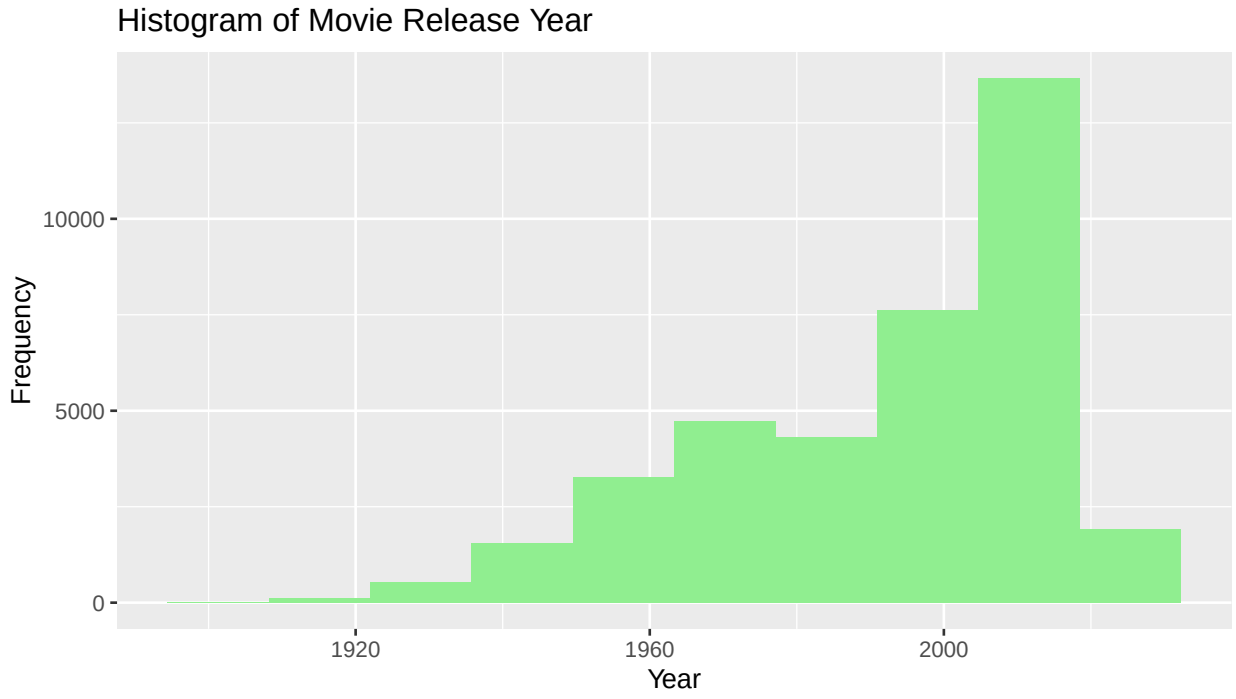
```
## [1890,1900) [1900,1910) [1910,1920) [1920,1930) [1930,1940) [1940,1950)
##           1           1          65         256         689        1190
## [1950,1960) [1960,1970) [1970,1980) [1980,1990) [1990,2000) [2000,2010)
##        2265        2980        3323        3363        4722        7358
## [2010,2020) [2020,2030)
##       10614         884
```

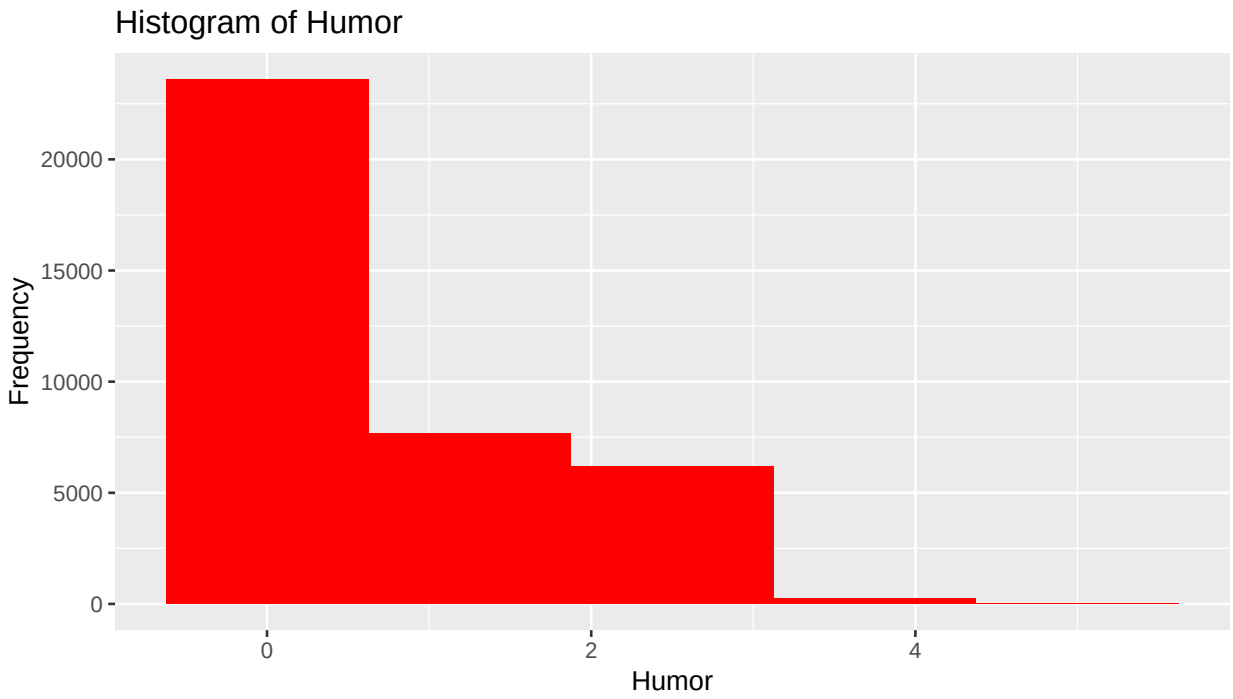
Plots and Charts

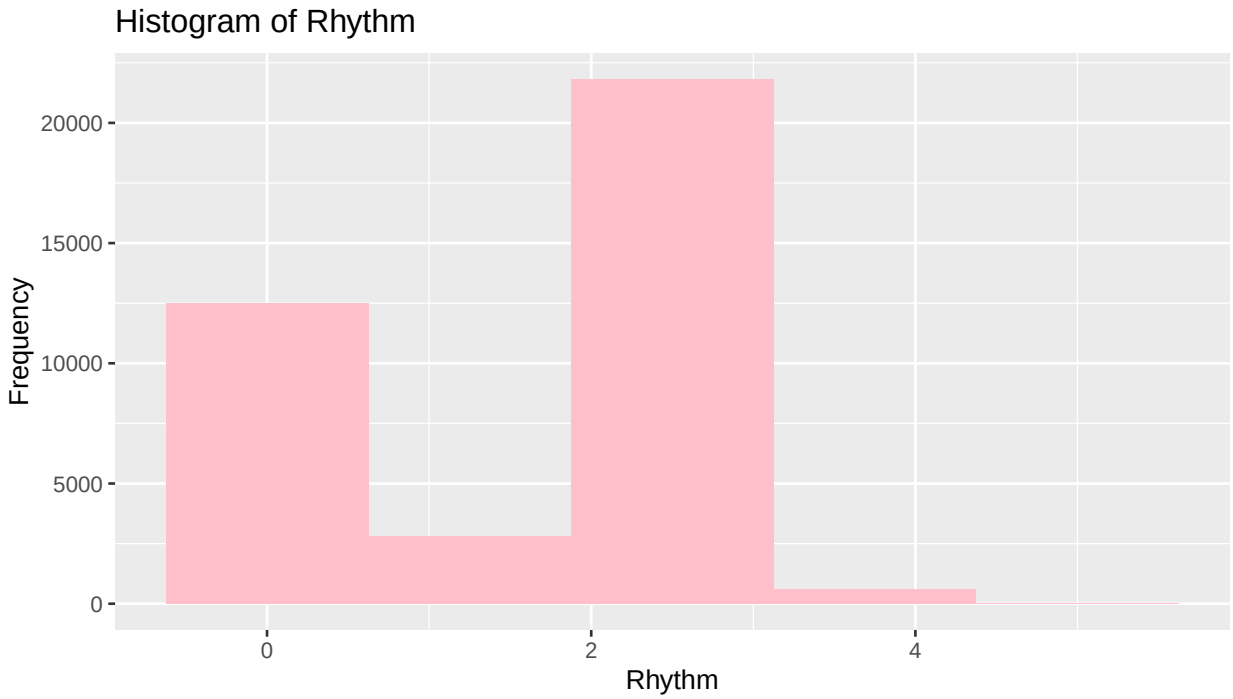
```
##
##      Action    Adventure    Animation    Biblical   Biography       Comedy
##        2047         1429          952          37         613         8694
##       Crime Documentary        Drama     Fantasy    Gangster   Grotesque
##         451         1802        11002        1307          54         237
##     History      Horror         Mélo     Musical   Mythology        Noir
##         116         2141          103         404          65         232
##     Romance Short Movie Sperimental       Sport         Spy  Super-hero
##         744           5          146           7         231          77
##    Thriller         War      Western
##        3316         382         1117
```

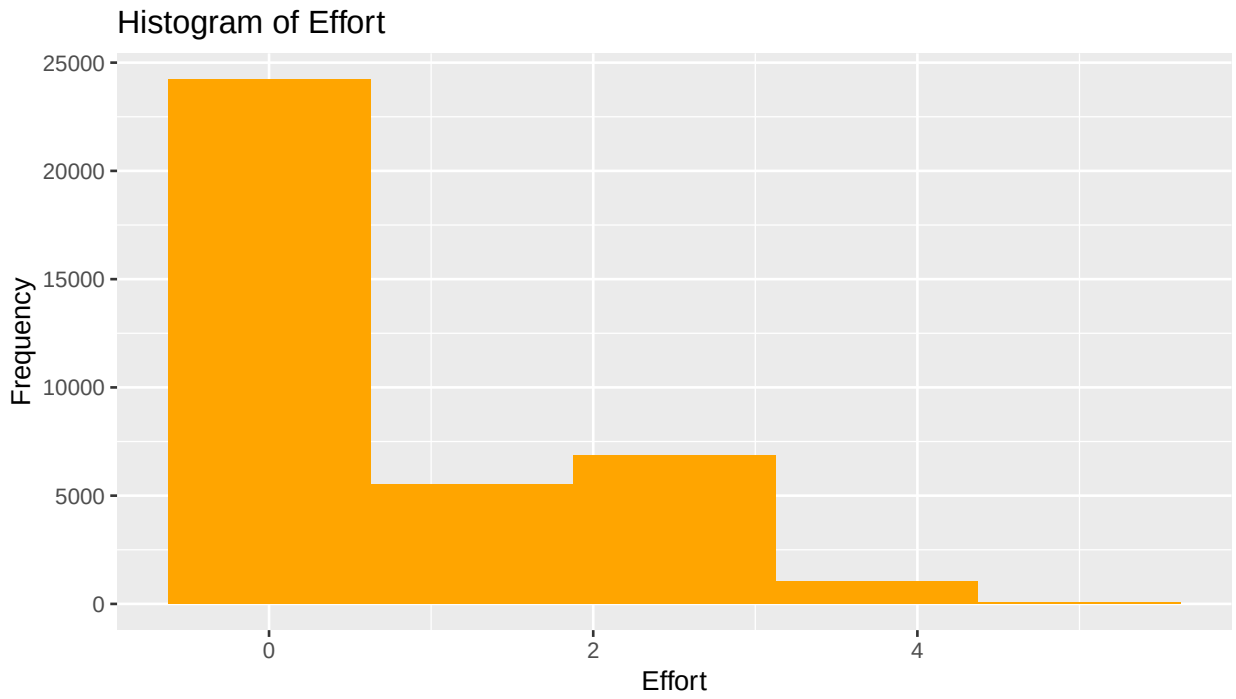


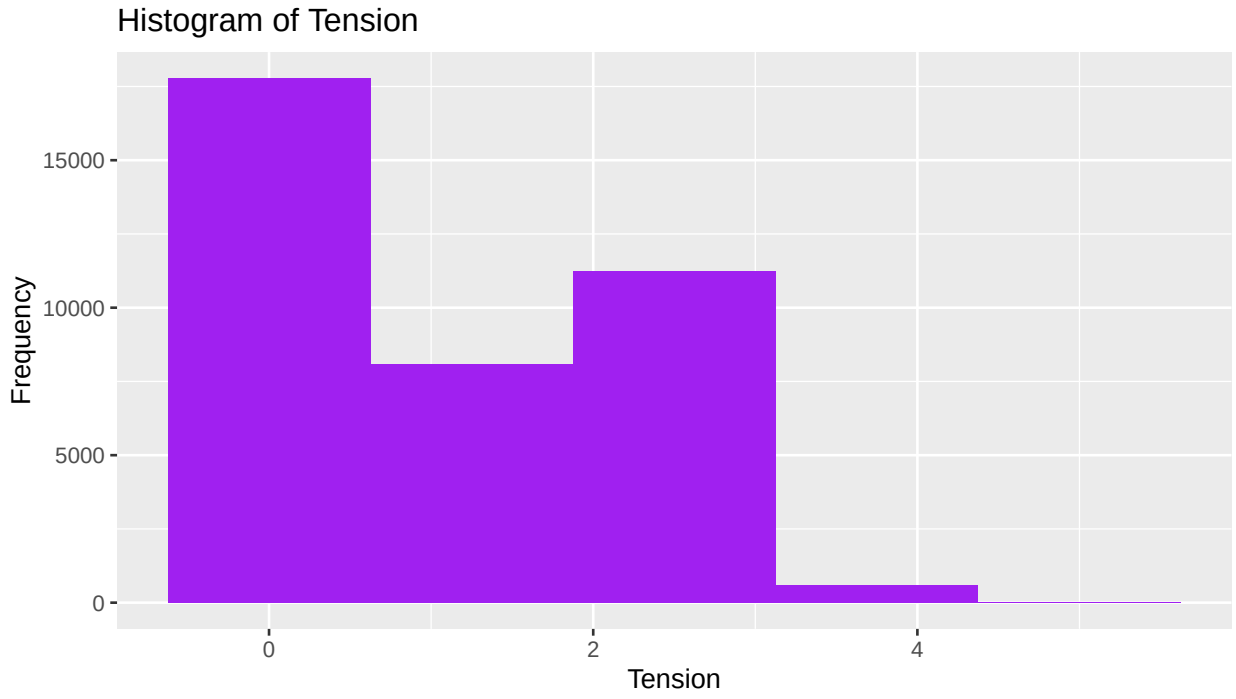

Histogram of Movie Length (for those less than 300 minutes)
Frequency
15000
10000
5000
0
0
100
200
300
Length of Movie (minutes)

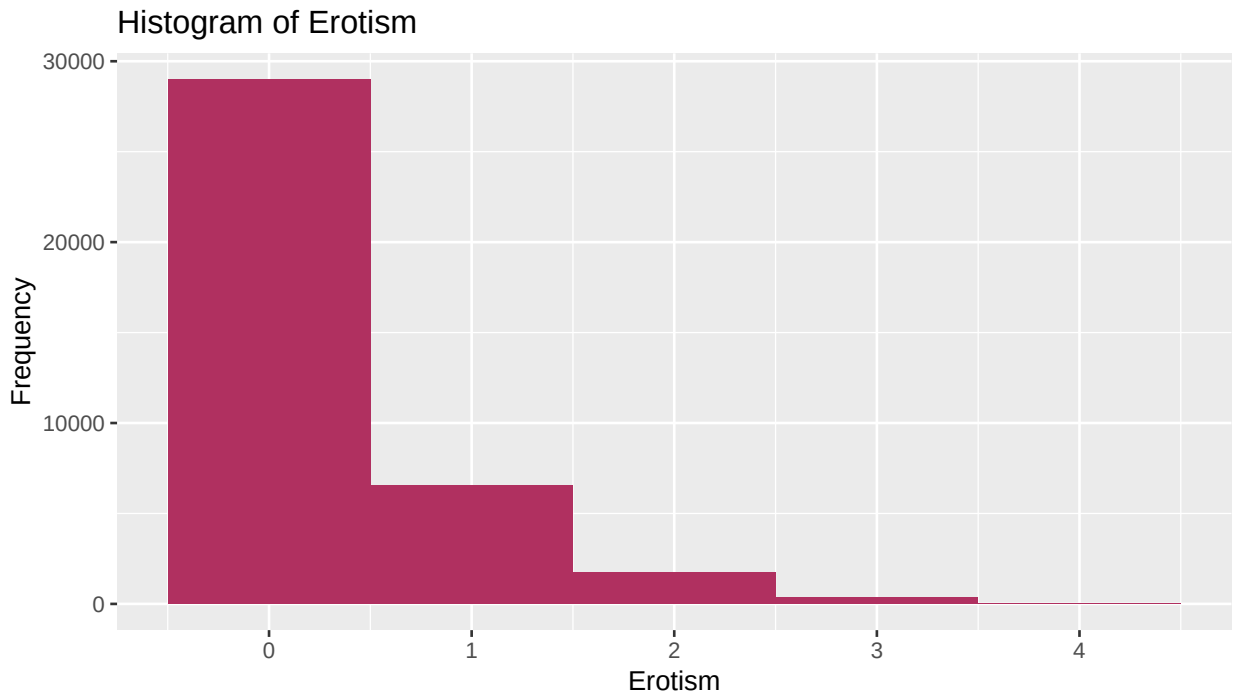

Histogram of Movie Release Year
Frequency
10000
5000
0
1920
1960
2000
Year


Histogram of Humor
Frequency
20000
15000
10000
5000
0
0
2
4
Humor


Histogram of Rhythm
Frequency
20000
15000
10000
5000
0
0
2
4
Rhythm


Histogram of Effort
Frequency
25000
20000
15000
10000
5000
0
0
2
4
Effort

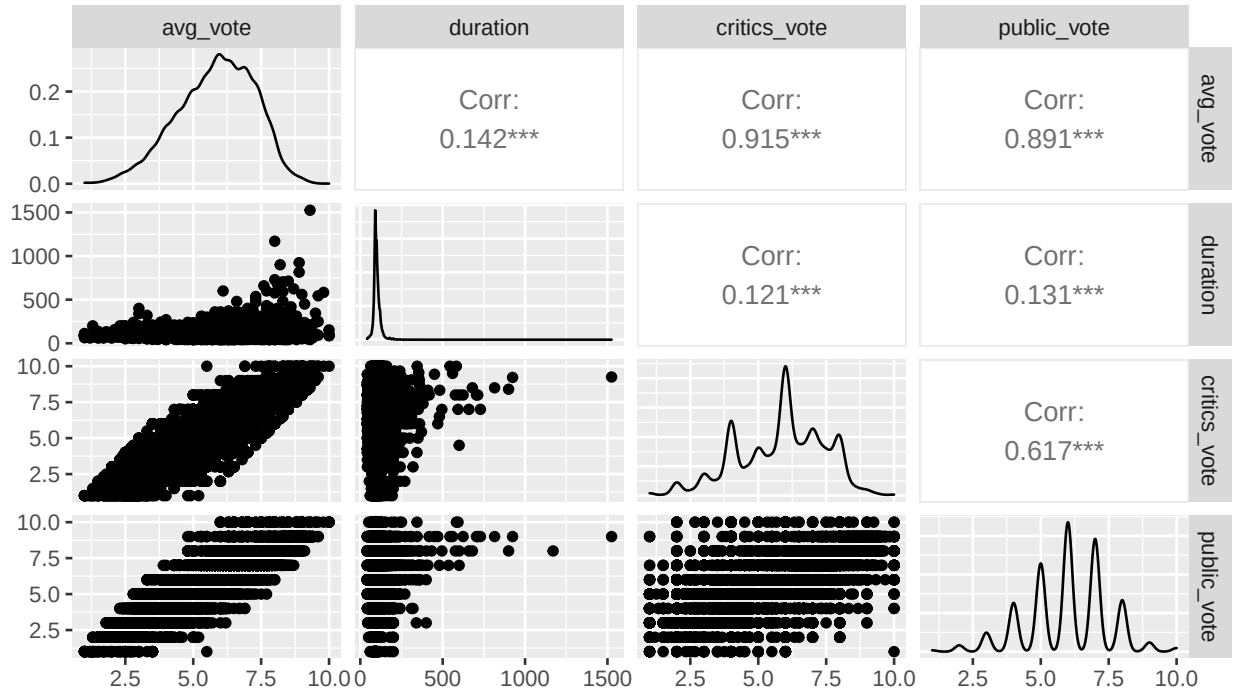

Histogram of Tension
Frequency
15000
10000
5000
0
0
2
4
Tension



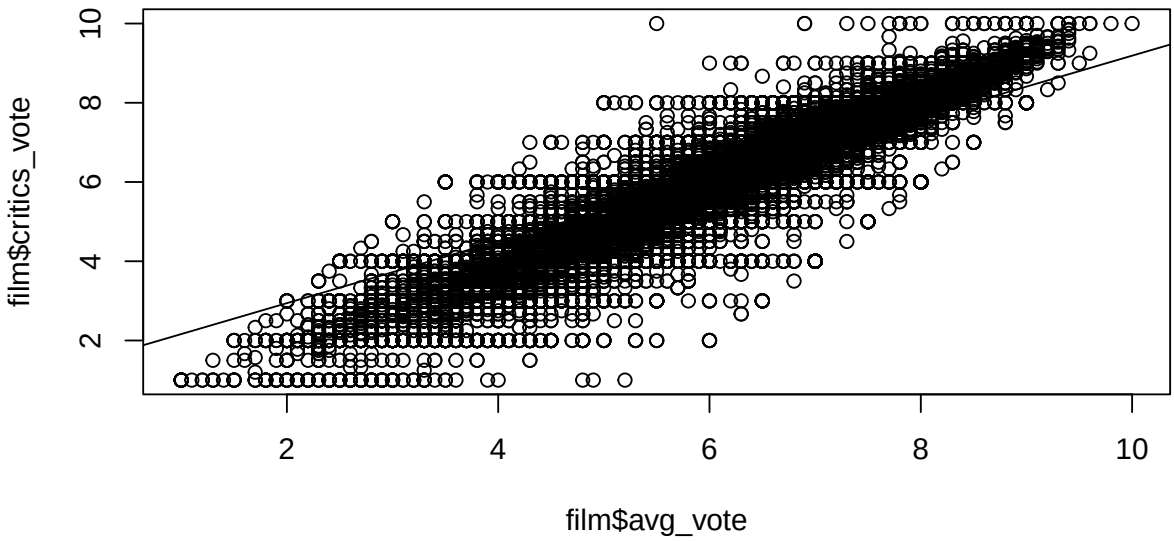
Let's look at the correlation between some of the numerical variables and average vote.

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 4085 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 219 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 4085 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 219 rows containing missing values

## Warning: Removed 4085 rows containing missing values (geom_point).

## Warning: Removed 4085 rows containing missing values (geom_point).

## Warning: Removed 4085 rows containing non-finite values (stat_density).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 4304 rows containing missing values

## Warning: Removed 219 rows containing missing values (geom_point).

## Warning: Removed 219 rows containing missing values (geom_point).

## Warning: Removed 4304 rows containing missing values (geom_point).
```

```
## Warning: Removed 219 rows containing non-finite values (stat_density).
```

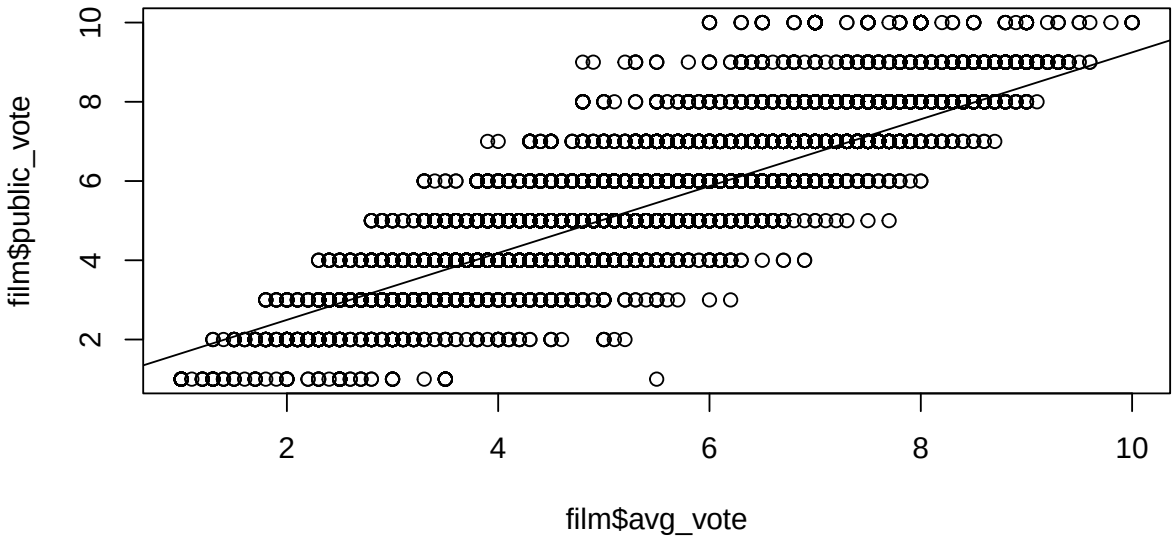


In our data we noticed that there are three different voting variables. Our ultimate goal is to predict the average vote, so we want to determine if the public vote and critics vote have a linear relationship or high correlation to the average vote. If that is the case, we will remove those variables from the regression as they have a somewhat linear dependency on one another.



We can see that the scatter plot shows a high correlation between the average vote and the critics vote with a correlation of 0.915. Thus there is almost a linear relationship between the variables so we will remove

critics vote for the sake of prediction and regression.



We can see that this scatter plot also shows a high correlation between the average vote and the public vote with a correlation of 0.891. Thus there is almost a linear relationship between the variables so we will remove public vote for the sake of prediction and regression as well.

PCA

Let's create PCA scores for our data set and see if they tell us anything important.

Linear Regression

Let's try to predict the average vote using some of the other variables in the data set. First, we will drop some of the variables in the data set that are not useful for the regression, such as `filmtv_id`,`year`, `title`,`country`, `directors`, `actors`, `description`, and `notes`. We will also remove `critics_vote` and `public_vote` as indicated previously.

Simple Regression

First let's start with a linear regression to determine what is the best model fro predicting the average vote. Specifically, let's determine if we can use `genre` to predict what the average vote is. We also can note that `genre` is a categorical/factor variable. We decided to start with `genre` as many people give different reviews based on the genres they like, thus we decided it could be significant in predicting the average vote the movie received.

```
##
## Call:
## lm(formula = avg_vote ~ genre, data = film2)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5959 -0.8715  0.1074  0.9825  4.4927
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.157792   0.029445 175.166  < 2e-16 ***
## genreAdventure    0.415616   0.045924   9.050  < 2e-16 ***
## genreAnimation    1.027082   0.052262  19.653  < 2e-16 ***
## genreBiblical    -0.003738   0.220984  -0.017 0.986505
## genreBiography    0.797673   0.061337  13.005  < 2e-16 ***
## genreComedy       0.349489   0.032728  10.678  < 2e-16 ***
## genreCrime        0.977241   0.069298  14.102  < 2e-16 ***
## genreDocumentary  1.438102   0.043034  33.418  < 2e-16 ***
## genreDrama        1.159705   0.032068  36.164  < 2e-16 ***
## genreFantasy      0.371129   0.047169   7.868 3.70e-15 ***
## genreGangster     1.534801   0.183666   8.356  < 2e-16 ***
## genreGrotesque    1.317314   0.091409  14.411  < 2e-16 ***
## genreHistory      0.775829   0.127149   6.102 1.06e-09 ***
## genreHorror       0.141788   0.041182   3.443 0.000576 ***
## genreMélo         1.549004   0.134528  11.514  < 2e-16 ***
## genreMusical      0.876862   0.072526  12.090  < 2e-16 ***
## genreMythology   -0.560869   0.167843  -3.342 0.000834 ***
## genreNoir         1.854277   0.092287  20.092  < 2e-16 ***
## genreRomance     -0.374727   0.057030  -6.571 5.07e-11 ***
## genreShort Movie  1.782208   0.596509   2.988 0.002812 **
## genreSperimental  1.433304   0.114119  12.560  < 2e-16 ***
## genreSport        0.156494   0.504388   0.310 0.756361
## genreSpy          0.636148   0.092466   6.880 6.09e-12 ***
## genreSuper-hero   0.761689   0.154648   4.925 8.46e-07 ***
## genreThriller     0.413740   0.037446  11.049  < 2e-16 ***
## genreWar          1.190114   0.074250  16.029  < 2e-16 ***
## genreWestern      0.628511   0.049557  12.683  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.332 on 37684 degrees of freedom
## Multiple R-squared:  0.1115, Adjusted R-squared:  0.1109
## F-statistic: 181.9 on 26 and 37684 DF,  p-value: < 2.2e-16

## Anova Table (Type II tests)
##
## Response: avg_vote
##           Sum Sq    Df F value    Pr(>F)
## genre       8396    26  181.94 < 2.2e-16 ***
## Residuals  66881 37684
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the summary, we can first not that the genre "Action" is the base for the factor and is represented by the intercept. We can also see that the majority of the different genres are significant in predicting the average vote. We can see that most of the genres increase the average vote given we are only looking at the affect that genre has on the average score. There were however a few genres that decreased the average score,

those being biblical, mythology, and romance. Looking at the ANOVA, however, we can see that `genre` as a whole is significant in predicting the average vote, as it's p-value is 2.2e-16.

##Multiple Regression Let's continue with genre, and add another variable to see if anything with the model changes. First, let's add on the variable `duration`. We decided to add duration as the next variable since the lenght of different movies also could be important to people reviewing the movies. For instance, a longer movie may loose the attention of some of the viewers and thus receive a lower score.

```
##
## Call:
## lm(formula = avg_vote ~ genre + duration, data = film2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1816 -0.8673  0.1115  0.9489  4.5502
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.501951   0.039241 114.726  < 2e-16 ***
## genreAdventure     0.405879   0.045549   8.911  < 2e-16 ***
## genreAnimation     1.111253   0.051943  21.394  < 2e-16 ***
## genreBiblical     -0.234842   0.219368  -1.071  0.28439
## genreBiography     0.673782   0.061036  11.039  < 2e-16 ***
## genreComedy        0.373338   0.032474  11.496  < 2e-16 ***
## genreCrime         0.992926   0.068733  14.446  < 2e-16 ***
## genreDocumentary   1.495481   0.042743  34.988  < 2e-16 ***
## genreDrama         1.135861   0.031819  35.697  < 2e-16 ***
## genreFantasy       0.360219   0.046785   7.700 1.40e-14 ***
## genreGangster      1.424176   0.182215   7.816 5.60e-15 ***
## genreGrotesque     1.329212   0.090661  14.661  < 2e-16 ***
## genreHistory       0.622965   0.126255   4.934 8.08e-07 ***
## genreHorror        0.200867   0.040913   4.910 9.16e-07 ***
## genreMélo          1.507836   0.133436  11.300  < 2e-16 ***
## genreMusical       0.851162   0.071939  11.832  < 2e-16 ***
## genreMythology    -0.537315   0.166471  -3.228  0.00125 **
## genreNoir          1.872135   0.091534  20.453  < 2e-16 ***
## genreRomance      -0.335022   0.056585  -5.921 3.23e-09 ***
## genreShort Movie   0.884989   0.592708   1.493  0.13541
## genreSperimental   1.462927   0.113190  12.925  < 2e-16 ***
## genreSport         0.196348   0.500258   0.392  0.69470
## genreSpy           0.613464   0.091713   6.689 2.28e-11 ***
## genreSuper-hero    0.634050   0.153466   4.132 3.61e-05 ***
## genreThriller      0.438733   0.037153  11.809  < 2e-16 ***
## genreWar           1.124741   0.073688  15.264  < 2e-16 ***
## genreWestern       0.665353   0.049173  13.531  < 2e-16 ***
## duration           0.006455   0.000258  25.022  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.321 on 37683 degrees of freedom
## Multiple R-squared:  0.1261, Adjusted R-squared:  0.1254
## F-statistic: 201.3 on 27 and 37683 DF,  p-value: < 2.2e-16
```

```
## Anova Table (Type II tests)
```

```
##
## Response: avg_vote
##           Sum Sq    Df F value    Pr(>F)
## genre       7972    26  175.62 < 2.2e-16 ***
## duration    1093     1  626.11 < 2.2e-16 ***
## Residuals  65788 37683
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the summary, we can see that holding the genre constant, the duration is significant in predicting the average vote. Further, holding the duration constant, the genres that decrease the average vote are the same from the single regression, those being biblical, mythology, and romance. In viewing the ANOVA, we can see that both duration, and genre as a whole are significant in predicting the average vote.

Let's run a linear model using all of the variables

```
## Anova Table (Type II tests)
##
## Response: avg_vote
##             Sum Sq    Df F value    Pr(>F)
## genre         5337    26  153.50 < 2.2e-16 ***
## duration       557     1  416.62 < 2.2e-16 ***
## total_votes   1741     1 1301.74 < 2.2e-16 ***
## humor          799     1  597.42 < 2.2e-16 ***
## rhythm         914     1  683.35 < 2.2e-16 ***
## effort        1459     1 1091.21 < 2.2e-16 ***
## tension       1070     1  799.92 < 2.2e-16 ***
## erotism        219     1  163.49 < 2.2e-16 ***
## year_mod      4734    13  272.34 < 2.2e-16 ***
## Residuals    50366 37664
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Call:
## lm(formula = avg_vote ~ ., data = film2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7816 -0.7279  0.0650  0.7869  5.0472
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.1295208  1.1568526   5.298 1.17e-07 ***
## genreAdventure       0.0264138  0.0405486   0.651 0.514785
## genreAnimation       0.9552268  0.0460391  20.748  < 2e-16 ***
## genreBiblical       -0.6772944  0.1923076  -3.522 0.000429 ***
## genreBiography       0.4651130  0.0537978   8.646  < 2e-16 ***
## genreComedy          0.1001990  0.0304239   3.293 0.000991 ***
## genreCrime           0.4961565  0.0609186   8.145 3.92e-16 ***
## genreDocumentary     1.4399959  0.0388877  37.030  < 2e-16 ***
## genreDrama           0.7532523  0.0288007  26.154  < 2e-16 ***
## genreFantasy        -0.0373674  0.0412406  -0.906 0.364897
## genreGangster        0.4999271  0.1600149   3.124 0.001784 **
```

```
## genreGrotesque       0.5430550  0.0805268   6.744 1.57e-11 ***
## genreHistory         0.0170656  0.1109855   0.154 0.877796
## genreHorror         -0.0050499  0.0361324  -0.140 0.888850
## genreMélo            0.5353789  0.1176193   4.552 5.34e-06 ***
## genreMusical         0.4749035  0.0641913   7.398 1.41e-13 ***
## genreMythology      -0.8075641  0.1469787  -5.494 3.95e-08 ***
## genreNoir            0.9701189  0.0809668  11.982  < 2e-16 ***
## genreRomance        -0.3483442  0.0498209  -6.992 2.76e-12 ***
## genreShort Movie    -0.2173529  0.5897257  -0.369 0.712453
## genreSperimental     1.2211775  0.1000287  12.208  < 2e-16 ***
## genreSport           0.1508110  0.4378604   0.344 0.730527
## genreSpy            -0.0032595  0.0811300  -0.040 0.967953
## genreSuper-hero     -0.1756325  0.1352256  -1.299 0.194017
## genreThriller        0.2959624  0.0327342   9.041  < 2e-16 ***
## genreWar             0.3552854  0.0656319   5.413 6.22e-08 ***
## genreWestern         0.0939378  0.0447030   2.101 0.035614 *
## duration             0.0047321  0.0002318  20.411  < 2e-16 ***
## total_votes          0.0038408  0.0001065  36.080  < 2e-16 ***
## humor                0.2446666  0.0100100  24.442  < 2e-16 ***
## rhythm              -0.2563408  0.0098061 -26.141  < 2e-16 ***
## effort               0.2401006  0.0072684  33.033  < 2e-16 ***
## tension              0.2665809  0.0094255  28.283  < 2e-16 ***
## erotism             -0.1411918  0.0110423 -12.786  < 2e-16 ***
## year_mod[1900,1910) -0.2679525  1.7392373  -0.154 0.877561
## year_mod[1910,1920)  0.2143896  1.1657187   0.184 0.854084
## year_mod[1920,1930)  0.1907501  1.1588039   0.165 0.869252
## year_mod[1930,1940) -0.2224094  1.1574113  -0.192 0.847617
## year_mod[1940,1950) -0.6617584  1.1570702  -0.572 0.567376
## year_mod[1950,1960) -0.8349391  1.1568454  -0.722 0.470460
## year_mod[1960,1970) -1.2016818  1.1567903  -1.039 0.298901
## year_mod[1970,1980) -1.3882119  1.1567593  -1.200 0.230113
## year_mod[1980,1990) -1.3660393  1.1567564  -1.181 0.237641
## year_mod[1990,2000) -1.2958687  1.1567215  -1.120 0.262595
## year_mod[2000,2010) -1.7095907  1.1566611  -1.478 0.139406
## year_mod[2010,2020) -1.8311035  1.1566067  -1.583 0.113391
## year_mod[2020,2030) -1.5082789  1.1572028  -1.303 0.192452
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.156 on 37664 degrees of freedom
## Multiple R-squared:  0.3309, Adjusted R-squared:  0.3301
## F-statistic:   405 on 46 and 37664 DF,  p-value: < 2.2e-16
```

Based on the ANOVA for this test, we can see that all of the variables used the the regression are significant.

LASSO

Trees

Logistic Regression

Now let's use our data to perform a Logistic Regression to determine whether the movie is good or bad. We will use the average vote numbers to make a binary variable that is 1 if the rating is greater than 6, and 0 otherwise.

```
## < table of extent 0 >
```

Now we want to split up our data in testing and training.

Text Analysis

Description EDA

```
## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : escap could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : time could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : relationship could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : take could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : discov could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : american could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : work could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : group could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : meet could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : film could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : former could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : children could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : father could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : hand could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : get could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : know could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : marri could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : place could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : stori could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : mother could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : year could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : new could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : forc could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : love could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : war could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : begin could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : even could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : three could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : manag could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : old could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : polic could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : woman could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : one could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : howev could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : investig could not be fit on page. It will not be plotted.

## Warning in wordcloud(names(total_freq)[1:100], total_freq[1:100], colors =
## cor.special, : man could not be fit on page. It will not be plotted.
```