Modeling and prediction for movies

kimnewzealand

12 June 2017

Setup

Load packages

```
library(ggplot2)
library(dplyr)
library(knitr)
options(digits=3)
```

Load data

```
setwd("~/LinRegProject")
load("movies.Rdata")
```

Part 1: Data

We will first take a look at the data structure of the datset.

There are 651 observations of randomly sampled movies produced and released in the year range 1970, 2014 from Rotten Tomatoes and IMDb, and 32 variables in the movies dataset. There are movies from different studios.

There is 1 duplicated movies in the dataset: Man on Wire so we will keep only the unique records.

```
# Keep unique records
movies <- unique(movies)
```

SAMPLING METHOD

The sources of the sample data, according to Wikipedia:

- The Internet Movie Database (abbreviated IMDb) is a very large online database of information related to films.
- Box Office Mojo - is a website that tracks box office revenue which was purchased by IMDb.
- Rotten Tomatoes is an American review aggregator website for film and television.

These sources are two representations of the population of movies in the US, therefore dependent.

The sampling method is a simple random sample, selected randomly from dependent sources (IMDb and Rotten Tomatoes) with each having an equal probably of selection. The within sample responses can be assumed to be independent and less than 10% of the US population, which is in millions. Since this is not an experiment, random assignment was not used.

This data is generalisable to population of movies in the US however it should not be used to establish causality.

Part 2: Research question

At Paramount pictures we are interested in the popularity of a movie, specifically what factors may influence the audience popularity rating to be considered in the planning of the next movie releases in order to boost box office sales.

Part 3: Exploratory data analysis

OUTCOME VARIABLE

The following variables are measures of popularity, of which we will pick one for the outcome. There may be collinearity in that they are correlated.

We are given that the two Rotten Tomatoes ratings are categorical variables however we need further information on the other variables, and to check if there are any missing values:

iMDB popularity measures:
- *imdb_rating*: Rating on IMDB. as per the website : We take all the individual votes cast by IMDb registered users and use them to calculate a single rating
- *imdb_num_votes*: Number of votes on IMDB

```
kable(t(as.matrix(summary(movies$imdb_rating))),format='pandoc')
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.9	5.9	6.6	6.49	7.3	9

```
kable(t(as.matrix(summary(movies$imdb_num_votes))),format='pandoc')
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
180	4540	15100	57600	58500	893000

Rotten Tomatoes popularity measures:
- critics_rating: Categorical variable for critics rating on Rotten Tomatoes (Certified - Fresh, Fresh, Rotten)
- critics_score: Critics score on Rotten Tomatoes
- audience_rating: Categorical variable for audience rating on Rotten Tomatoes (Spilled, Upright)
- audience_score: Audience score on Rotten Tomatoes
- top200_box: Whether or not the movie is in the Top 200 Box Office list on BoxOfficeMojo (no, yes)

```
kable(t(as.matrix(summary(movies$critics_rating))),format='pandoc')
```

Certified Fresh	Fresh	Rotten
134	209	307

```
kable(t(as.matrix(summary(movies$critics_score))),format='pandoc')
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	33	61	57.6	83	100

```
kable(t(as.matrix(summary(movies$audience_rating))),format='pandoc')
```

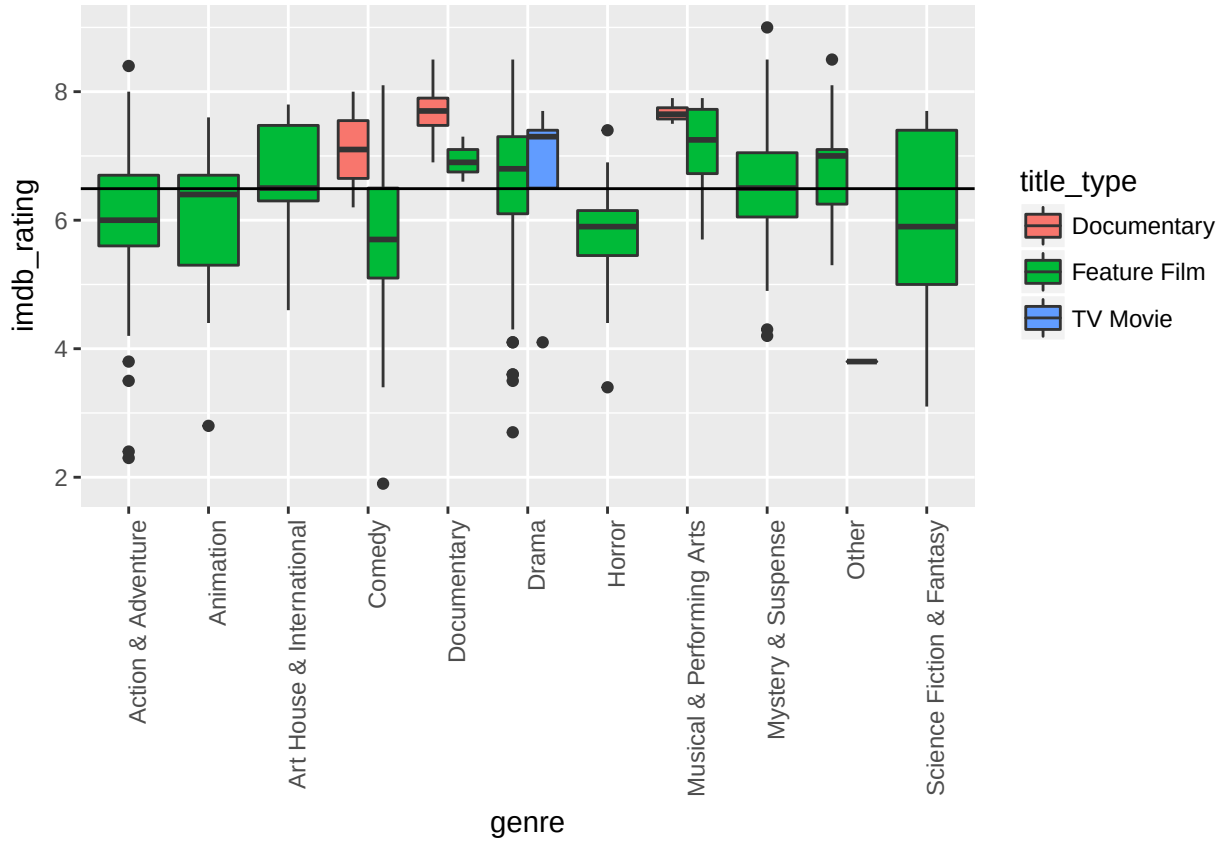
Spilled	Upright
275	375

```
kable(t(as.matrix(summary(movies$audience_score))),format='pandoc')
```

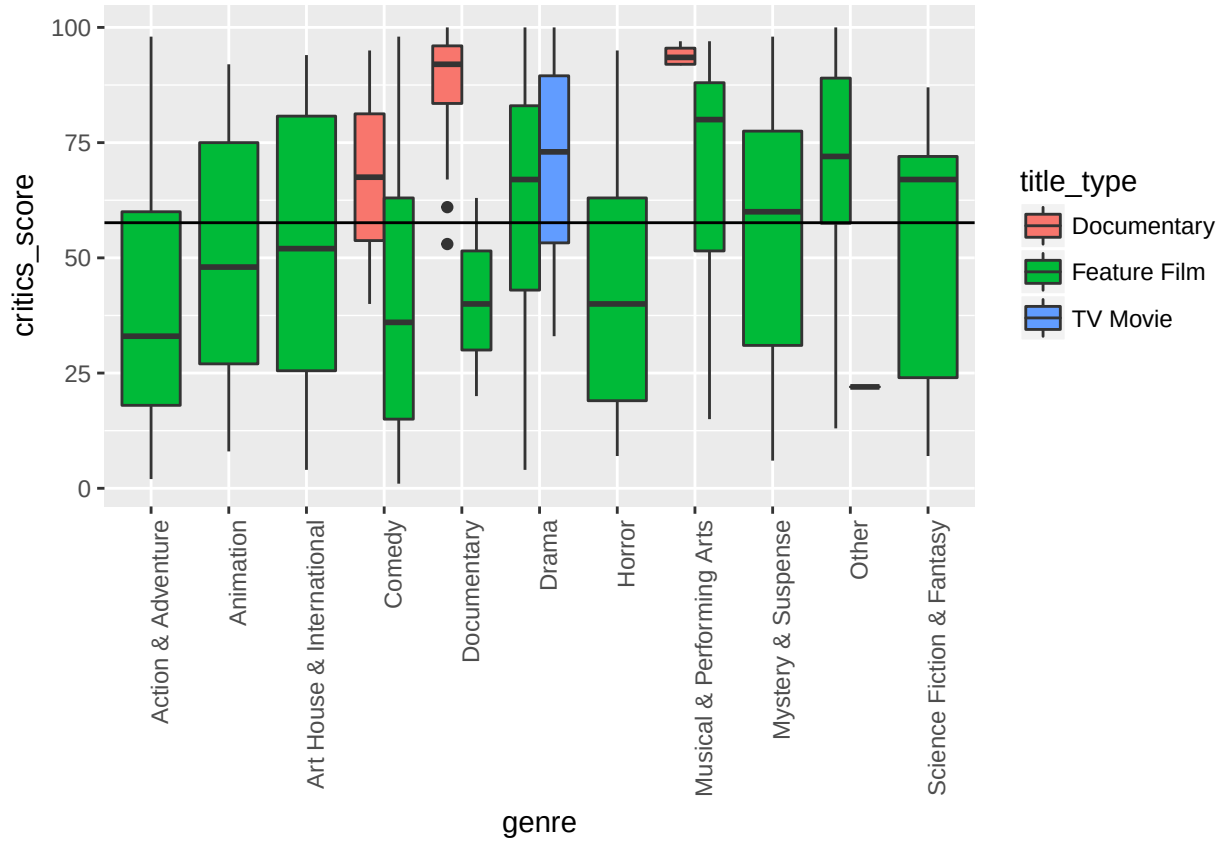
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
11	46	65	62.3	80	97

The *critics_rating* and *audience_rating* are categorical summaries of the *critics_score* and *audience_score*. The *imdb_rating* is a also a summary of *imdb_num_votes*. The *top200_box* is categorical whereas we will need a numerical prediction. Therefore we will look at visual summaries of the other 3 measures, *imdb_rating*, *critics_score* and *audience_score*.

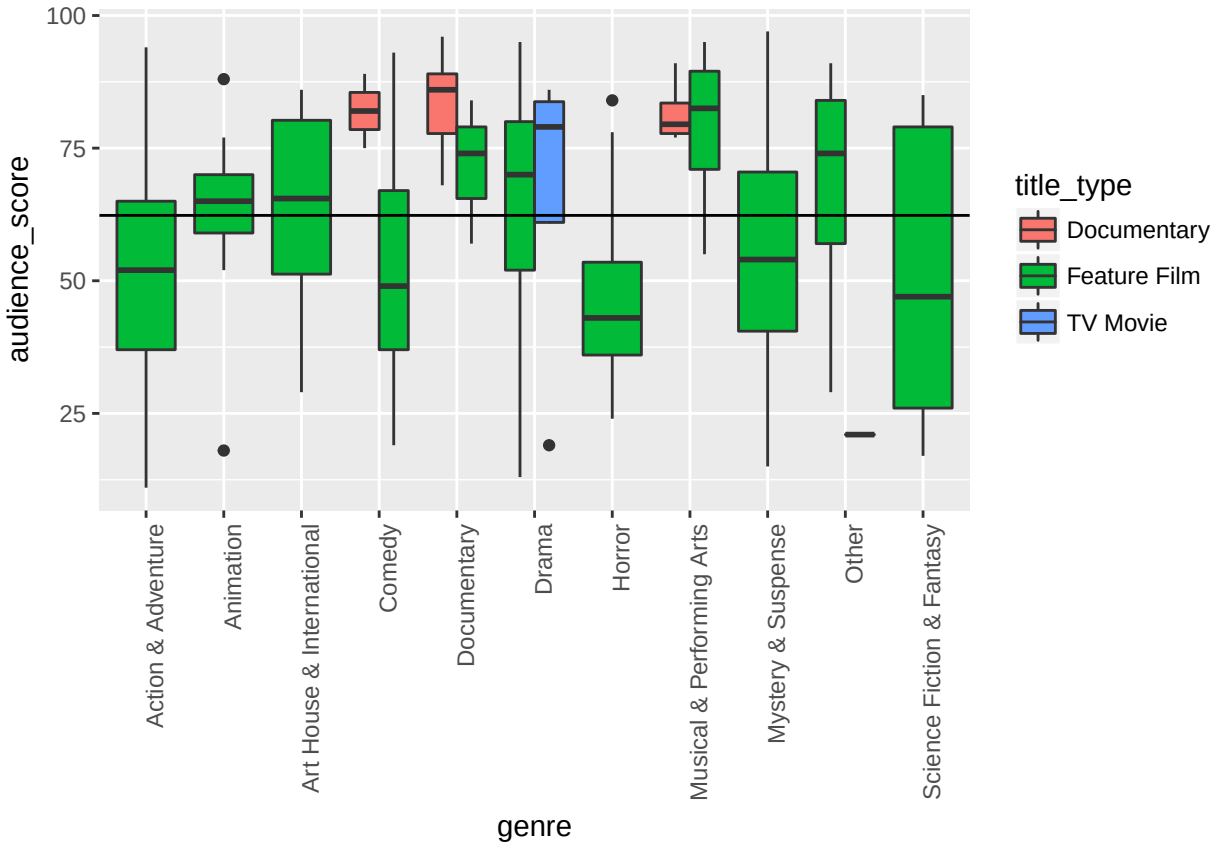
```
# View the IMDb rating against the genre in a boxplot, with a horozontal line plotted for mean imdb rat
g <- ggplot(movies,aes(genre,imdb_rating))
g + geom_boxplot(aes(fill=title_type)) +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=1)) +
  geom_hline(yintercept=mean(movies$imdb_rating),aes(color=blue))
```




```
# View the Rotten Tomatoes criticss score against the genre and title type in a boxplot with a horozont
g <- ggplot(movies,aes(genre,critics_score))
g + geom_boxplot(aes(fill=title_type)) +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=1)) +
  geom_hline(yintercept=mean(movies$critics_score),aes(color=blue))
```

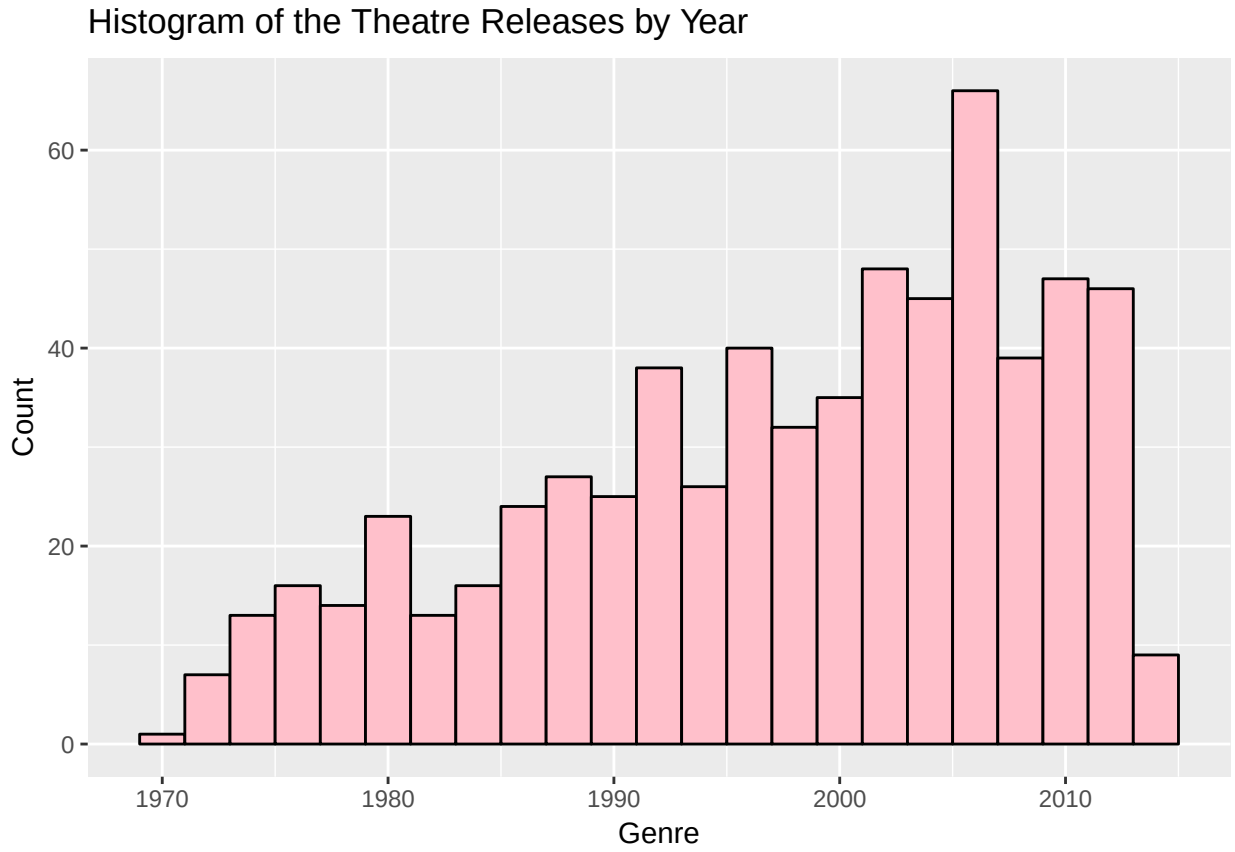



```
# View the Rotten Tomatoes audience score against the genre in a boxplot
g <- ggplot(movies,aes(genre,audience_score))
g + geom_boxplot(aes(fill=title_type)) +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=1)) +
  geom_hline(yintercept=mean(movies$audience_score),aes(color=blue))
```



These three plots, *audience_score*, *critics_score* and *imdb rating* are very similar plot by genre and title type therefore appear correlated but the variability is higher for *critics_score*, then the *audience_score* then the *imdb_rating*. We will use the imdb_rating with the least variability as this may have better predictive value.

```
# Plot a histogram of the movie releases by year
g <- ggplot(movies,aes(thtr_rel_year))
g + geom_histogram(binwidth=2,fill="pink",colour="black")+
   labs(title="Histogram of the Theatre Releases by Year", x="Genre",y="Count")
```



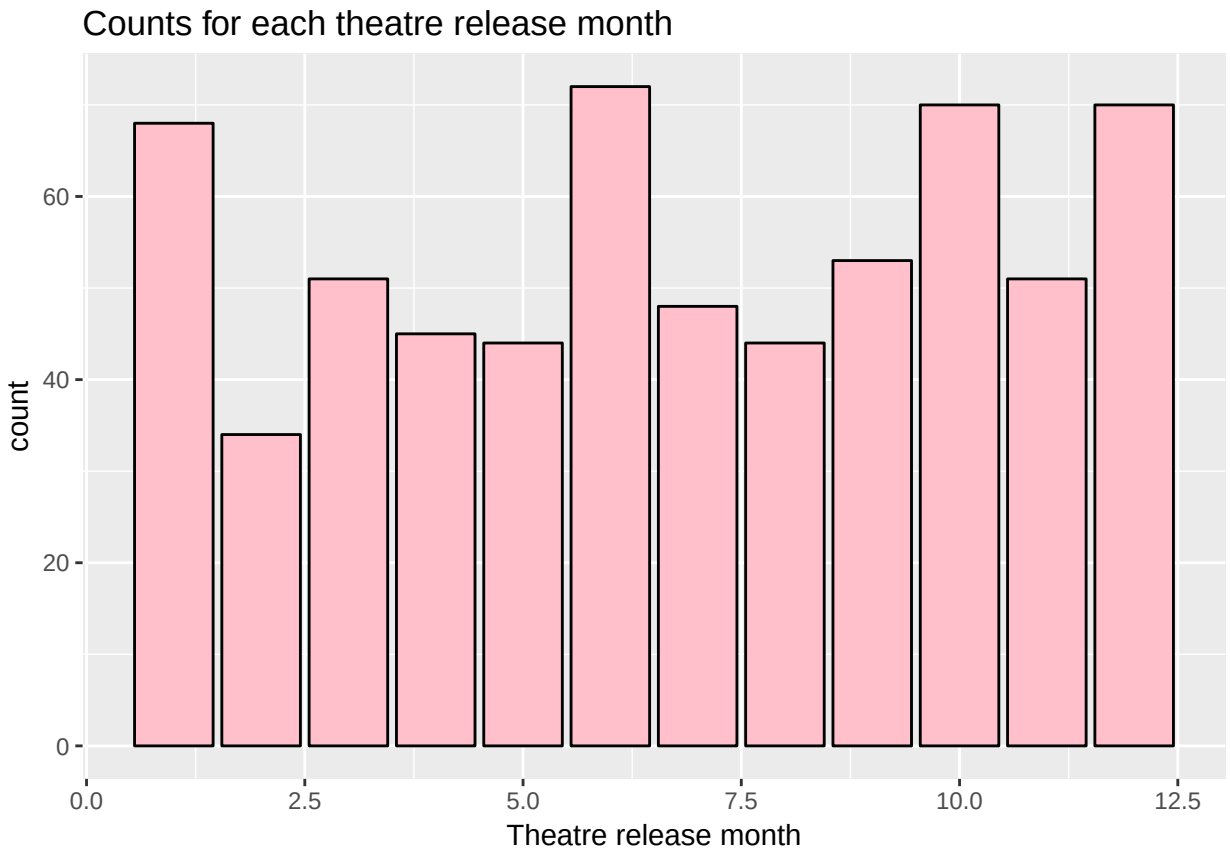
The number of movies by the theatre release year has a left skew and not normally distributed ie the number of releases have generally increased year to year.

```
# Remove descriptive variables title,imdb_url , rt_url ; the duplicated  critics_rating and audience_ra
movies2 <- movies %>%
  select(-c(title,imdb_url ,critics_rating,audience_rating, critics_score,audience_score, imdb_num_votes
```

We will consider thtr_rel_month and dvd_rel_month as categorical variables to explore seasonal variations in the data.

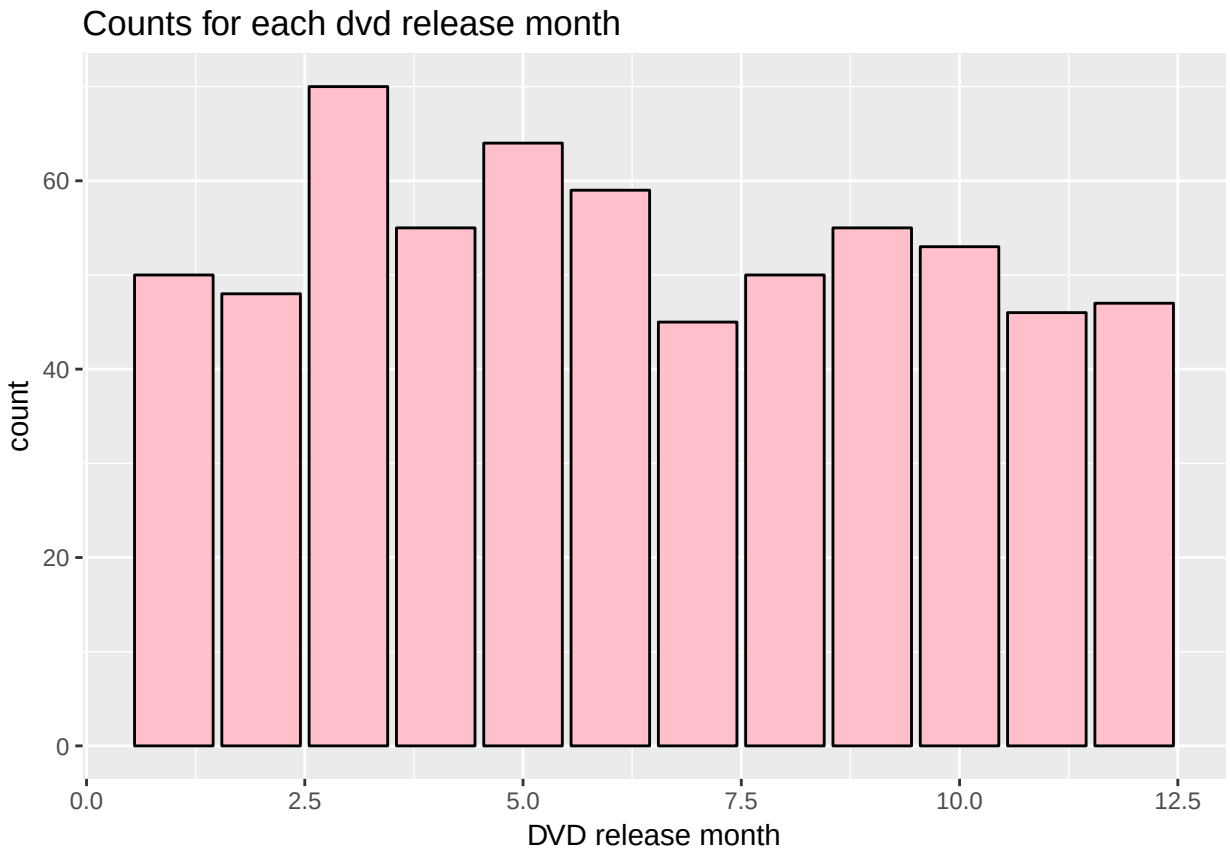
We consider thtr_rel_month and dvd_rel_month as a categorical variable as it may explain seasonal variations within the data.

```
month <- c('Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec')
movies2['thtr_rel_month'] <- factor(movies2$thtr_rel_month, labels=month)
movies2['dvd_rel_month'] <- factor(movies2$dvd_rel_month, labels=month)
# Plot the theatre release month counts
g <- ggplot(movies, aes(thtr_rel_month))
g+ geom_bar(color='black',fill='pink')+labs(title = "Counts for each theatre release month", x = "Theati
```




```
# Plot the DVD release month counts
g <- ggplot(movies, aes(dvd_rel_month))
g + geom_bar(color='black',fill='pink')+labs(title = "Counts for each dvd release month", x = "DVD relea
```

```
## Warning: Removed 8 rows containing non-finite values (stat_count).
```



There may be some seasonality to the months of release in theatres and on DVD.

Part 4: Modeling

The method that we will use is Multivariate Linear Regression to predict a numerical variable in the dataset using the relevant plots and statistics.

First we set the hypothesis test for the model as a whole:

The null hypothesis $H_0 : \beta_1 = \beta_i ... = \beta_k = 0$ that at no explanatory variables are a significant predictor of the outcome.

The alternative hypothesis H_A : at least one β_k is different to 0, and a explanatory variable is a significant predictor.

We will use a stepwise backwards model selection, reviewing the p-values of the coefficients and adjusted R-squared.

```
# Summary of the full model:
model1 <- lm(imdb_rating ~ .,movies2)
summary(model1)
```

```
##
## Call:
## lm(formula = imdb_rating ~ ., data = movies2)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
## -3.580 -0.475  0.077  0.551  2.076
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    41.637494  17.203217    2.42   0.0158 *
## title_typeFeature Film         -0.848134   0.337352   -2.51   0.0122 *
## title_typeTV Movie             -1.565671   0.528759   -2.96   0.0032 **
## genreAnimation                 -0.166166   0.362654   -0.46   0.6470
## genreArt House & International  0.765429   0.285628    2.68   0.0076 **
## genreComedy                    -0.120673   0.154185   -0.78   0.4341
## genreDocumentary                0.791386   0.363102    2.18   0.0297 *
## genreDrama                      0.584469   0.132089    4.42  1.1e-05 ***
## genreHorror                    -0.161712   0.230691   -0.70   0.4836
## genreMusical & Performing Arts  0.915572   0.311642    2.94   0.0034 **
## genreMystery & Suspense         0.398298   0.172281    2.31   0.0211 *
## genreOther                      0.550727   0.260430    2.11   0.0349 *
## genreScience Fiction & Fantasy -0.033589   0.342544   -0.10   0.9219
## runtime                         0.010034   0.002249    4.46  9.8e-06 ***
## mpaa_ratingNC-17               -0.567833   0.699013   -0.81   0.4169
## mpaa_ratingPG                  -0.574947   0.258495   -2.22   0.0265 *
## mpaa_ratingPG-13               -0.797075   0.269784   -2.95   0.0033 **
## mpaa_ratingR                   -0.466636   0.259733   -1.80   0.0729 .
## mpaa_ratingUnrated             -0.221062   0.304158   -0.73   0.4676
## thtr_rel_year                   0.000877   0.004892    0.18   0.8578
## thtr_rel_monthFeb               0.019631   0.195285    0.10   0.9200
## thtr_rel_monthMar              -0.190643   0.176610   -1.08   0.2808
## thtr_rel_monthApr              -0.170602   0.182085   -0.94   0.3492
## thtr_rel_monthMay              -0.219148   0.183065   -1.20   0.2317
## thtr_rel_monthJun              -0.227131   0.162721   -1.40   0.1633
## thtr_rel_monthJul              -0.099972   0.183652   -0.54   0.5864
## thtr_rel_monthAug              -0.058697   0.186047   -0.32   0.7525
## thtr_rel_monthSep              -0.196568   0.173603   -1.13   0.2580
## thtr_rel_monthOct              -0.079928   0.163349   -0.49   0.6248
## thtr_rel_monthNov              -0.101749   0.181723   -0.56   0.5758
## thtr_rel_monthDec              -0.003020   0.166893   -0.02   0.9856
## thtr_rel_day                    0.003737   0.004230    0.88   0.3774
## dvd_rel_year                   -0.018514   0.010906   -1.70   0.0901 .
## dvd_rel_monthFeb                0.210474   0.188550    1.12   0.2648
## dvd_rel_monthMar               -0.030846   0.176066   -0.18   0.8610
## dvd_rel_monthApr               -0.074807   0.186822   -0.40   0.6890
## dvd_rel_monthMay               -0.115114   0.178288   -0.65   0.5187
## dvd_rel_monthJun               -0.079465   0.182135   -0.44   0.6628
## dvd_rel_monthJul                0.235331   0.192776    1.22   0.2227
## dvd_rel_monthAug                0.235997   0.193481    1.22   0.2230
## dvd_rel_monthSep                0.044702   0.181553    0.25   0.8056
## dvd_rel_monthOct                0.135982   0.189036    0.72   0.4722
## dvd_rel_monthNov                0.355859   0.192944    1.84   0.0656 .
## dvd_rel_monthDec                0.272356   0.187539    1.45   0.1470
## dvd_rel_day                    -0.001182   0.004145   -0.29   0.7756
## best_pic_nomyes                 0.802813   0.238211    3.37   0.0008 ***
## best_pic_winyes                 0.198944   0.419769    0.47   0.6357
## best_actor_winyes              -0.022398   0.109784   -0.20   0.8384
## best_actress_winyes            -0.010312   0.120618   -0.09   0.9319
```

```
## best_dir_winyes                0.333399   0.157028    2.12   0.0342 *
## top200_boxyes                  0.562689   0.246992    2.28   0.0231 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.895 on 590 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.36,   Adjusted R-squared:  0.306
## F-statistic: 6.64 on 50 and 590 DF,  p-value: <2e-16
```

The full model adjusted R-squared is 0.306.

```
# Model removing title_type
model2 <- lm(imdb_rating ~  . - title_type,movies2)
summary(model2)$adj.r.squared
```

```
## [1] 0.297
```

```
# Model removing the genre
model3 <- lm(imdb_rating ~ .-genre ,movies)
summary(model3)$adj.r.squared
```

```
## [1] NaN
```

```
# Model removing the runtime
model4 <- lm(imdb_rating ~ .-runtime ,movies2)
summary(model4)$adj.r.squared
```

```
## [1] 0.284
```

```
# Model removing the thtr_rel_year
model5 <- lm(imdb_rating ~ .-thtr_rel_year ,movies2)
summary(model5)$adj.r.squared
```

```
## [1] 0.307
```

```
# Model removing the thtr_rel_month
model6 <- lm(imdb_rating ~ .-thtr_rel_month ,movies2)
summary(model6)$adj.r.squared
```

```
## [1] 0.313
```

```
# Model removing the dvd_rel_year
model7 <- lm(imdb_rating ~ .-dvd_rel_year ,movies2)
summary(model7)$adj.r.squared
```

```
## [1] 0.304
```

```
# Model removing the thtr_rel_day
model8 <- lm(imdb_rating ~ .-thtr_rel_day  ,movies2)
summary(model8)$adj.r.squared
```

```
## [1] 0.306
```

```
# Model removing the dvd_rel_month
model9 <- lm(imdb_rating ~ .-dvd_rel_month  ,movies2)
summary(model9)$adj.r.squared
```

```
## [1] 0.3
```

```
# Model removing the dvd_rel_day
model10 <- lm(imdb_rating ~ .-dvd_rel_day ,movies2)
summary(model10)$adj.r.squared
```

```
## [1] 0.307
```

```
# Model removing the best_pic_nom
model11 <- lm(imdb_rating ~ .-best_pic_nom  ,movies2)
summary(model11)$adj.r.squared
```

```
## [1] 0.294
```

```
# Model removing the best_pic_win
model12 <- lm(imdb_rating ~ .-best_pic_win  ,movies2)
summary(model12)$adj.r.squared
```

```
## [1] 0.307
```

```
# Model removing the best_actor_win
model13 <- lm(imdb_rating ~ .-best_actor_win ,movies2)
summary(model13)$adj.r.squared
```

```
## [1] 0.307
```

```
# Model removing the best_actress_win
model14 <- lm(imdb_rating ~ .-best_actress_win ,movies2)
summary(model14)$adj.r.squared
```

```
## [1] 0.307
```

```
# Model removing the best_dir_win
model15 <- lm(imdb_rating ~ .-best_dir_win  ,movies2)
summary(model15)$adj.r.squared
```

```
## [1] 0.302
```

```
# Model removing the  top200_box
model16 <- lm(imdb_rating ~ .-top200_box ,movies2)
summary(model16)$adj.r.squared
```

```
## [1] 0.301
```

We will remove the predictors that reduce the adjusted R-Squared and come up with a final model with the with the highest adjusted R-Squared.

```
# Final linear model
modelfinal <- lm(imdb_rating ~ title_type + genre + runtime  + mpaa_rating + thtr_rel_month + dvd_rel_y
summary(modelfinal)
```

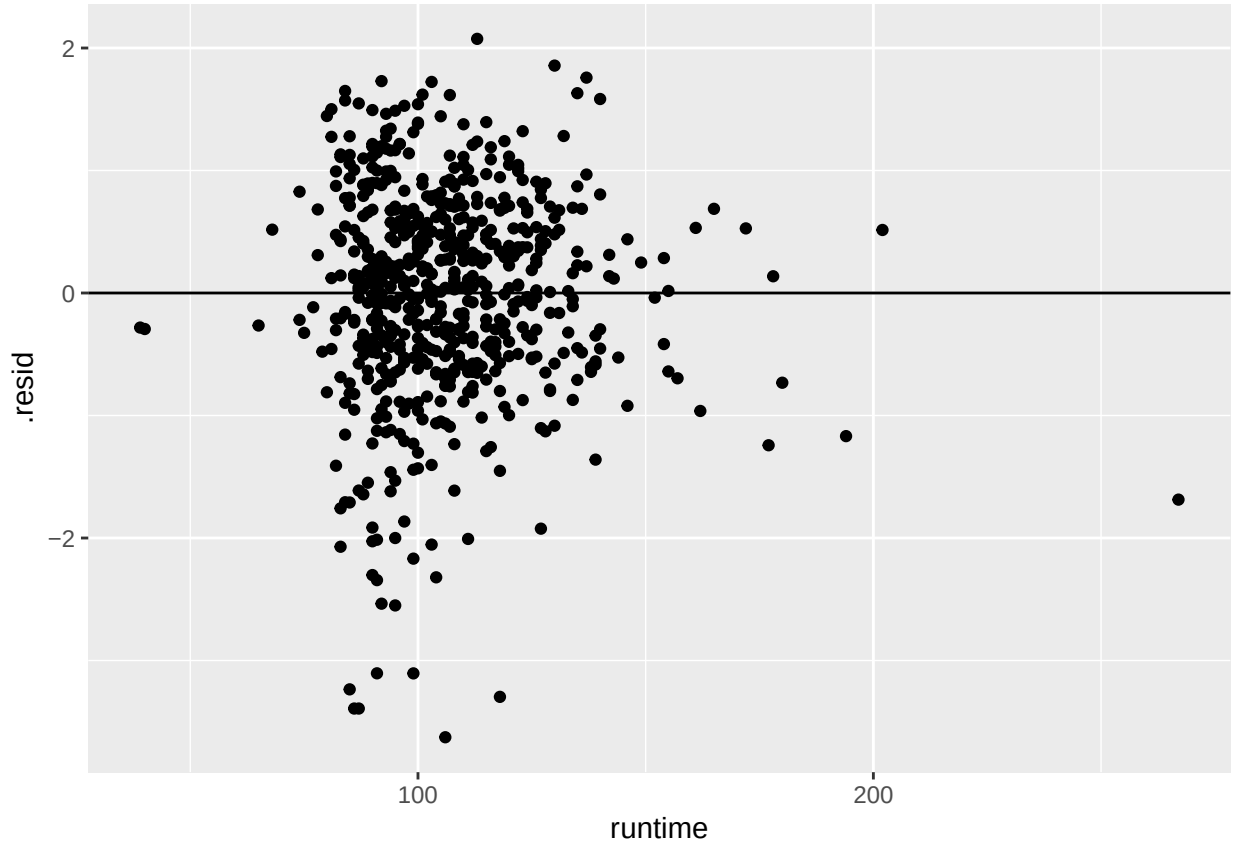
```
##
## Call:
## lm(formula = imdb_rating ~ title_type + genre + runtime + mpaa_rating +
##     thtr_rel_month + dvd_rel_year + dvd_rel_month + best_pic_nom +
##     best_dir_win + top200_box, data = movies2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.626 -0.482  0.060  0.575  2.075
##
## Coefficients:
```

```
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     39.43144   16.40740    2.40   0.0166 *
## title_typeFeature Film          -0.85338    0.33591   -2.54   0.0113 *
## title_typeTV Movie              -1.54445    0.52527   -2.94   0.0034 **
## genreAnimation                  -0.16150    0.35871   -0.45   0.6527
## genreArt House & International  0.78397    0.28240    2.78   0.0057 **
## genreComedy                     -0.12374    0.15230   -0.81   0.4169
## genreDocumentary                 0.78885    0.36124    2.18   0.0294 *
## genreDrama                       0.58343    0.12949    4.51  8.0e-06 ***
## genreHorror                     -0.15755    0.22724   -0.69   0.4884
## genreMusical & Performing Arts  0.91450    0.30946    2.96   0.0032 **
## genreMystery & Suspense          0.39705    0.16928    2.35   0.0193 *
## genreOther                       0.52681    0.25726    2.05   0.0410 *
## genreScience Fiction & Fantasy -0.05448    0.33989   -0.16   0.8727
## runtime                          0.00995    0.00217    4.58  5.8e-06 ***
## mpaa_ratingNC-17                -0.58241    0.69286   -0.84   0.4009
## mpaa_ratingPG                   -0.57484    0.25690   -2.24   0.0256 *
## mpaa_ratingPG-13                -0.78612    0.26401   -2.98   0.0030 **
## mpaa_ratingR                    -0.46305    0.25565   -1.81   0.0706 .
## mpaa_ratingUnrated              -0.22477    0.29975   -0.75   0.4536
## thtr_rel_monthFeb                0.03521    0.19255    0.18   0.8550
## thtr_rel_monthMar               -0.17083    0.17456   -0.98   0.3282
## thtr_rel_monthApr               -0.13912    0.17846   -0.78   0.4360
## thtr_rel_monthMay               -0.19628    0.17965   -1.09   0.2750
## thtr_rel_monthJun               -0.20509    0.15977   -1.28   0.1998
## thtr_rel_monthJul               -0.06931    0.17973   -0.39   0.6999
## thtr_rel_monthAug               -0.03641    0.18179   -0.20   0.8413
## thtr_rel_monthSep               -0.17956    0.17095   -1.05   0.2940
## thtr_rel_monthOct               -0.05929    0.15919   -0.37   0.7097
## thtr_rel_monthNov               -0.08385    0.17934   -0.47   0.6403
## thtr_rel_monthDec                0.01721    0.16282    0.11   0.9159
## dvd_rel_year                    -0.01653    0.00818   -2.02   0.0437 *
## dvd_rel_monthFeb                 0.20634    0.18507    1.11   0.2653
## dvd_rel_monthMar                -0.03199    0.17475   -0.18   0.8548
## dvd_rel_monthApr                -0.05902    0.18420   -0.32   0.7488
## dvd_rel_monthMay                -0.10630    0.17703   -0.60   0.5484
## dvd_rel_monthJun                -0.07628    0.17942   -0.43   0.6709
## dvd_rel_monthJul                 0.22644    0.19123    1.18   0.2368
## dvd_rel_monthAug                 0.23311    0.19158    1.22   0.2242
## dvd_rel_monthSep                 0.04161    0.17975    0.23   0.8170
## dvd_rel_monthOct                 0.14059    0.18673    0.75   0.4518
## dvd_rel_monthNov                 0.36112    0.19043    1.90   0.0584 .
## dvd_rel_monthDec                 0.26599    0.18539    1.43   0.1519
## best_pic_nomyes                  0.84261    0.20871    4.04  6.1e-05 ***
## best_dir_winyes                  0.34948    0.14944    2.34   0.0197 *
## top200_boxyes                    0.56696    0.24552    2.31   0.0213 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.891 on 596 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.359,  Adjusted R-squared:  0.311
## F-statistic: 7.58 on 44 and 596 DF,  p-value: <2e-16
```

In order to perform diagnostics for this model, check the following conditions:

1) Linear relationship between each (numerical) explanatory variable

```
# Check the residuals plot of each of the numerical variables, runtime
g <- ggplot(modelfinal,aes(x=runtime,y=.resid))
g + geom_point()+
  geom_hline(yintercept=0)
```

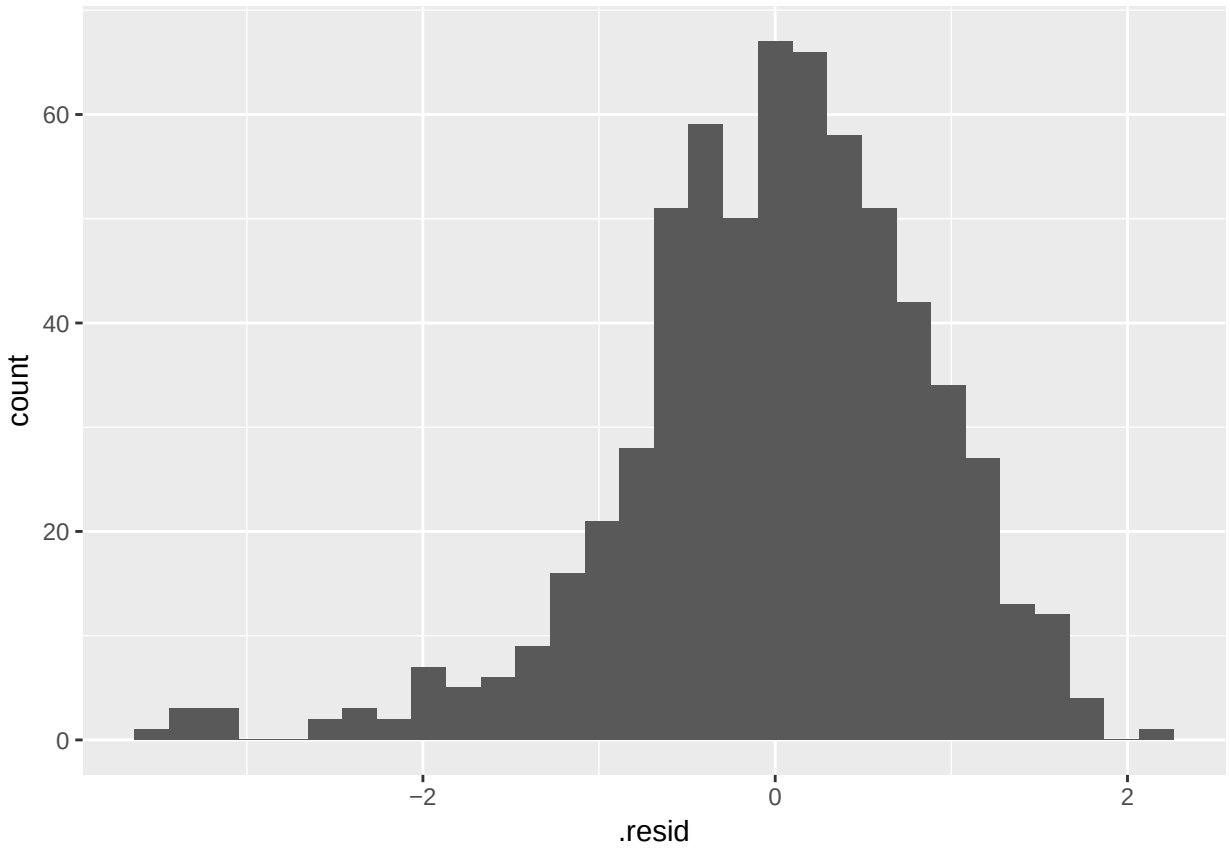


These residuals are scattered around 0 so this condition is met.

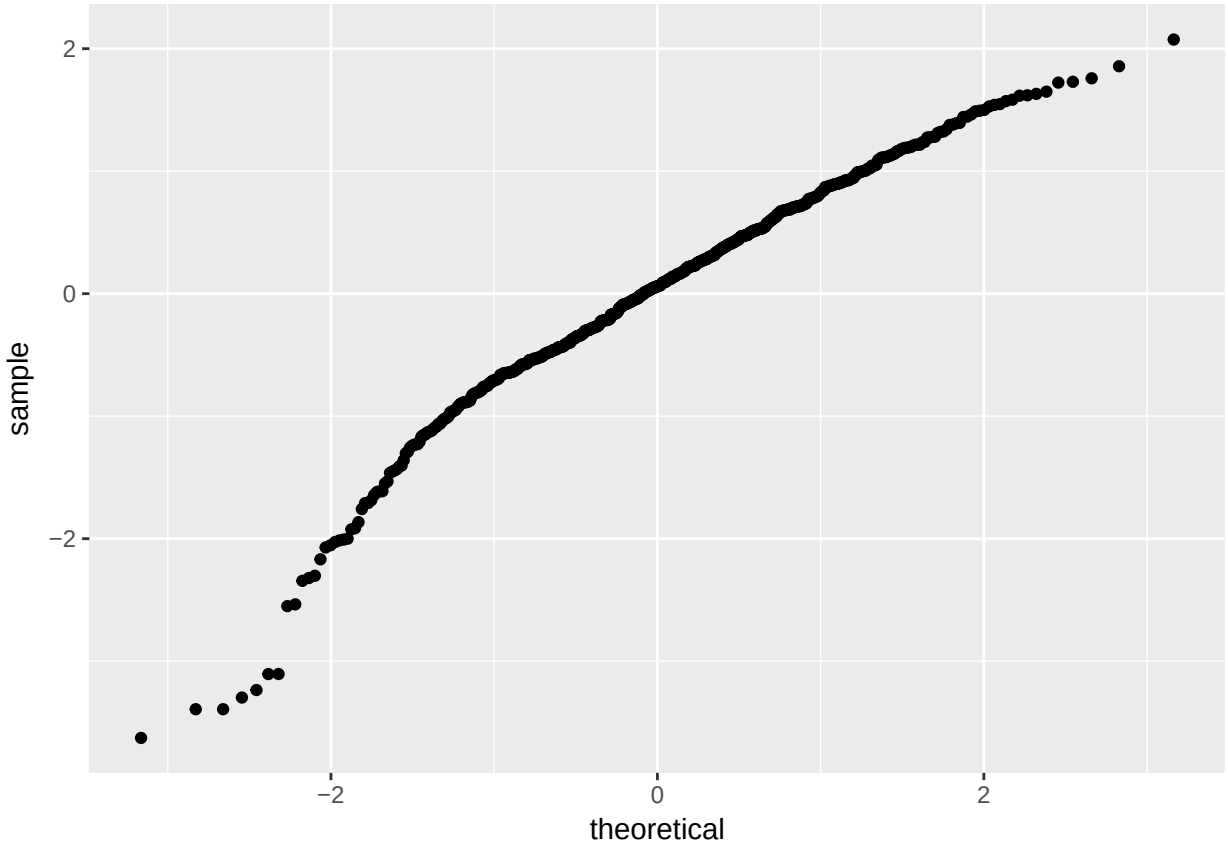
2) Nearly normal distribution of residuals

```
# Check using a histogram of the residuals
g <- ggplot(modelfinal,aes(.resid))
g + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



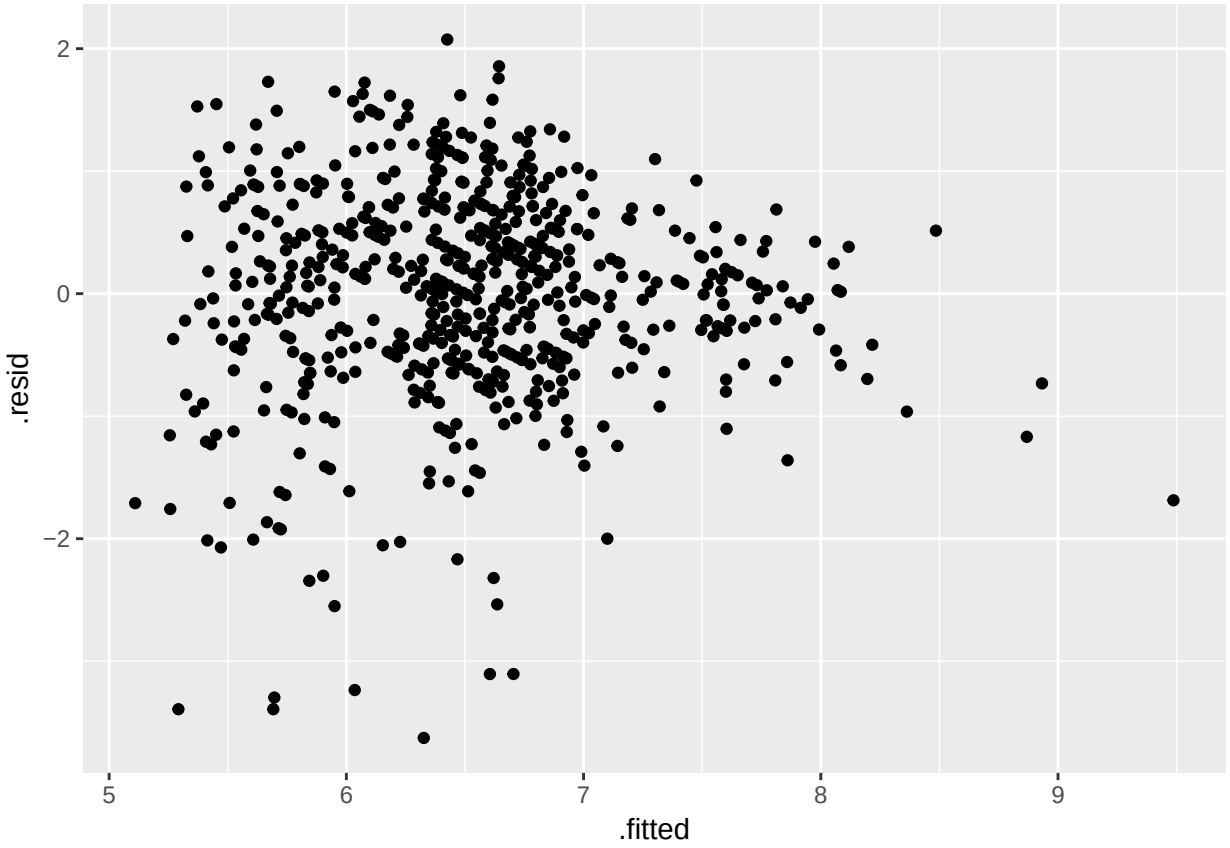

```
# Check using a QQ plot the MLR:
g <- ggplot(modelfinal,aes(sample= .resid))
g + stat_qq()
```



The histogram plot has an almost normal distribution, but we can say that this condition is met.

3) Constant variability of residuals

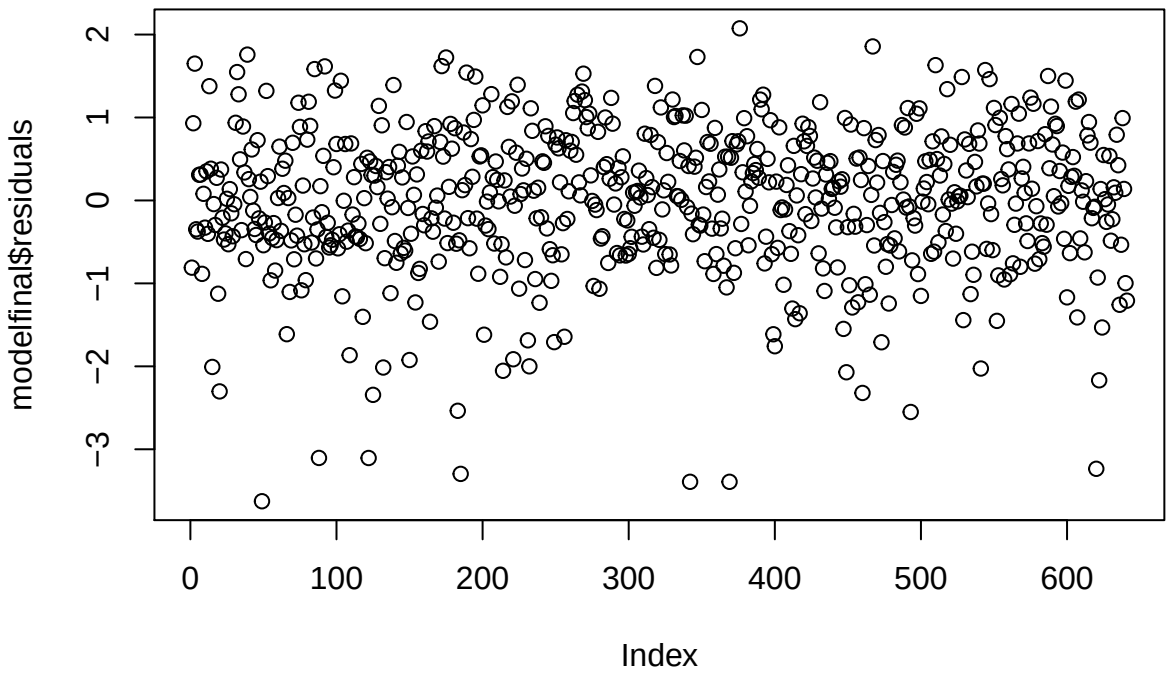
```
# Check using a scatterplot of the fitted values against the residuals
g <- ggplot(modelfinal,aes(y=.resid,x=.fitted))
g + geom_point()
```



Since these are randomly scattered around 0, with some uncertainty where the fan is wider, but this condition is met.

4) Independence of residuals (and hence observations)

```
# Check using a scatterplot of the residuals versus order of data collection
plot(modelfinal$residuals)
```



There are no patterns therefore this condition appears to be met.

Part 5: Prediction

We will use the move 2016 La La Land sourced manually the Rotten Tomatoes and iMDB websites.
https://www.rottentomatoes.com/m/la_la_land/
http://www.imdb.com/title/tt3783958/

Note that this has Genre: Comedy, Drama, Musical & Performing Arts, and we only have 1 field for Genre. For this prediction we will use Musical & Performing Arts.

```
# Create a dataframe for 2016 movie La La Land from the Rotten tomates and iMDB websites
LaLaLand <- c("Feature Film","Musical & Performing Arts",128,"PG-13",2016,"Dec",9,2017,"Apr",25,8.2,"ye
LaLaLand <- as.data.frame((t(LaLaLand)),stringsAsFactors = FALSE)
names(LaLaLand) <- names(movies2)
LaLaLand$runtime <- as.numeric(LaLaLand$runtime)
LaLaLand$thtr_rel_year <- as.numeric(LaLaLand$thtr_rel_year)
LaLaLand$thtr_rel_day <- as.numeric(LaLaLand$thtr_rel_day)
LaLaLand$dvd_rel_year <- as.numeric(LaLaLand$dvd_rel_year)
LaLaLand$dvd_rel_day <- as.numeric(LaLaLand$dvd_rel_day)
LaLaLand$imdb_rating <- as.numeric(LaLaLand$imdb_rating)
```

We can predict the iMDB rating with a confidence interval for each of the slope parameter of each predictor: β_k which would interpret as 95% confident that, all else being equal, the model predicts the iMDB rating.

```r
# Predict the imdb_rating
predict(modelfinal,LaLaLand,interval='prediction',level=0.95)
```

```
##    fit  lwr  upr
## 1 8.36 6.39 10.3
```

Part 6: Conclusion

As the F statistic p-value is $< 5\%$ we can reject the null hypothesis in favour of the alternate hypothesis, there is more than one β_k that is significant predictor of popularity of a movie, the iMDB rating. A multivariate linear model was created and the conditions were checked and met for this model.

In using this model to predict a iMDB rating for a 2016 movie La La Land, the prediction 8.36 compared to the current rating 8.2.

However this movie was categorised under multiple genres "Comedy, Drama, Musical & Performing Arts" in Rotten Tomatoes, so the prediction value would vary by the selected single Genre field. Alternatively the source dataset could be further enhanced with multiple genre fields.

Interestingly, the timing of theatre and DVD release was included as a factor therefore in addition to the other factors, these release dates should be considered when planning the next movie to boost box office sales.