Modeling and prediction for movies

Setup

Load packages

```
library(ggplot2)
library(dplyr)
library(statsr)
library(MASS)
library(olsrr)
suppressMessages(library("tidyverse"))
```

Load data

Make sure your data and R Markdown files are in the same directory. When loaded your data file will be called `movies`. Delete this note when before you submit your work.

```
load("movies.Rdata")
```

Part 1: Data

Data set we have here provides us information on how audience and critics like movies along with different features about the movies. This data set is acuqired from Rotten Tomatoes and IMDB, websites which are very popular for movies information and reviews. This data set has 651 randomly sampled movies which were released before 2016. As the technique used to collect the data is random sampling,we can say that the conclusions made from the dataset should be generalizable to over all population

We can only look for evidence for associations in the data set and we cannot derive casual relations because there is no random assignment is used for the variables understand consideration

---

Part 2: Research question

Movies are very popular entertainment sources. People wait for their favorite movies, love them, hate them, discuss about them.Because of being so common in daily life and having huge user base, it is a billion dollar industry. Consumers(audiences and critics both) are the ones who can make or break a movies' future. If we do not look into the data, we might think that consumers liking or disliking movies is totally random. But, there are so many attributes such as who is the lead actor/director of a movie, how intersting the trailer of a movie is etc, for a movie that might actually influence the consumers decison.

Here, we have a dataset of 651 movies with each movie having 32 variables or attributes. There might be many more features we might be missing that might effect movie popularity But, it is a good idea to make the best out of what we have in hand. In real world, not everytime we might have all the data what we need. If we can build a model that can factor in all or few features of this dataset, we should be able to understand, atleast to some extent, about what is the ideal receipe for making a good popular? or How to prioritize features while creating a movie? It interests me because great reviews are directly proportional to how much return on investment will a movie result in.

Interesting observation that could be made here is, model needs to know how popular the movie going to be from the data set but we don't have any column named popularity in the dataset.One more column which is close to be an indicator of popularity is `critics_score`. But, usually critics tend to review the movie even before it is released. So, it might be one of the independent varaibles which might influence a movie to be popular or not. Also, it seem like `imdb_rating` or `audience_score` are potential target variables which shows how much rating did the movie receive on IMDB and Rotten Tomatoes respectively.

```
summary(movies$imdb_rating)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.900   5.900   6.600   6.493   7.300   9.000
```

```
summary(movies$audience_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   46.00   65.00   62.36   80.00   97.00
```

```
summary(movies$imdb_num_votes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     180    4546   15116   57533   58300  893008
```

These two variables are on a different scale. Imdb rating is on a 10-point scale where as audience score varies between 11 - 97. As these two are equally good candidates for target variable, it might be a good idea to create a dervied variable, 'popularity' from these two.

```
movies[(movies$title == 'Driving Miss Daisy') | (movies$title == 'Saint of 9/11') , ][, c('title','crit
```

```
## # A tibble: 2 x 5
##   title              critics_score audience_score imdb_rating imdb_num_votes
##   <chr>                      <dbl>          <dbl>       <dbl>          <int>
## 1 Driving Miss Da~              81             81         7.4          69338
## 2 Saint of 9/11                 84             79         7.8            180
```

But, rating and popularity are two different things. A movie rating might be pretty high if you look a absolute number but the total number of votes might be very low.

'Saint of 9/11' movie released in 2006 in the dataset has high imdb_rating(7.8), audience_score(79) & critics_score(84). But, total number of votes this movie received is only 180 compared to max votes for any movie in the data set is 893008.

If you compare this movie with 'Driving Miss Daisy', both of these movies have very similary critics_score, audience_score, imdb_rating but they have very significant difference in total number of votes on imdb.

So,along with rating of a movie, it is a good idea to weigh in the number of votes it received to create our dependent variable. Also, after a movie certain level of popularity, it start getting more popular very quickly that is it grown exponentially in terms of popularity.

Considering these two factors, Bayesian average is being used here with 3rd quartile value of IMDB votes as average instead of max value of IMDB votes. This process is done after the cleaning of dataset because there are bunch of NA's which is spitting out errors.

Firstly, let's create an average rating based on imdb_rating & audience_score. To put them on same scale, imdb_ratings are multiplied by 10

```
movies$popularity <- ((movies$imdb_rating * 10) + movies$audience_score) / 2
movies[movies$title == 'Saint of 9/11',]
```

```
## # A tibble: 1 x 33
##   title      title_type  genre    runtime mpaa_rating studio thtr_rel_year
##   <chr>      <fct>       <fct>      <dbl> <fct>       <fct>          <dbl>
```

```
## 1 Saint of ~ Documentary Documen~      90 Unrated     IFC              2006
## # ... with 26 more variables: thtr_rel_month <dbl>, thtr_rel_day <dbl>,
## #   dvd_rel_year <dbl>, dvd_rel_month <dbl>, dvd_rel_day <dbl>,
## #   imdb_rating <dbl>, imdb_num_votes <int>, critics_rating <fct>,
## #   critics_score <dbl>, audience_rating <fct>, audience_score <dbl>,
## #   best_pic_nom <fct>, best_pic_win <fct>, best_actor_win <fct>,
## #   best_actress_win <fct>, best_dir_win <fct>, top200_box <fct>,
## #   director <chr>, actor1 <chr>, actor2 <chr>, actor3 <chr>,
## #   actor4 <chr>, actor5 <chr>, imdb_url <chr>, rt_url <chr>,
## #   popularity <dbl>
```

Using this, Ra = W * R +(1−W) * R0 formula we will be accounting for movies with different where

Ra= averaged ('bayesian') rating R= individual rating: average rating for one movie R0= global average rating for all the movies W= weight factor: votes/3rd quartile of votes in data

```
movies$popularity <- (movies$imdb_num_votes / quantile(movies$imdb_num_votes, 0.75))* movies$popularity
```

To summarize, research questions could be,

"Can we predict the derived dependent varibale `popularity` from various attributes of the movies in the dataset after it is released?"

This could be valuble to business because, companies producing movies could invest their money in marketeting and advertising cleverly depending upon likelyhood of it being popular or likeable for audience

---

Part 3: Exploratory data analysis

Before, we start building model, it is a good idea to dig into data and clean it up a bit. Also, we need to visualy see what kind of patterns are hidden inside the data.

```
str(movies)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    651 obs. of  33 variables:
##  $ title           : chr  "Filly Brown" "The Dish" "Waiting for Guffman" "The Age of Innocence" ...
##  $ title_type      : Factor w/ 3 levels "Documentary",..: 2 2 2 2 2 1 2 2 1 2 ...
##  $ genre           : Factor w/ 11 levels "Action & Adventure",..: 6 6 4 6 7 5 6 6 5 6 ...
##  $ runtime         : num  80 101 84 139 90 78 142 93 88 119 ...
##  $ mpaa_rating     : Factor w/ 6 levels "G","NC-17","PG",..: 5 4 5 3 5 6 4 5 6 6 ...
##  $ studio          : Factor w/ 211 levels "20th Century Fox",..: 91 202 167 34 13 163 147 118 88 84
##  $ thtr_rel_year   : num  2013 2001 1996 1993 2004 ...
##  $ thtr_rel_month  : num  4 3 8 10 9 1 1 11 9 3 ...
##  $ thtr_rel_day    : num  19 14 21 1 10 15 1 8 7 2 ...
##  $ dvd_rel_year    : num  2013 2001 2001 2001 2005 ...
##  $ dvd_rel_month   : num  7 8 8 11 4 4 2 3 1 8 ...
##  $ dvd_rel_day     : num  30 28 21 6 19 20 18 2 21 14 ...
##  $ imdb_rating     : num  5.5 7.3 7.6 7.2 5.1 7.8 7.2 5.5 7.5 6.6 ...
##  $ imdb_num_votes  : int  899 12285 22381 35096 2386 333 5016 2272 880 12496 ...
##  $ critics_rating  : Factor w/ 3 levels "Certified Fresh",..: 3 1 1 1 3 2 3 3 2 1 ...
##  $ critics_score   : num  45 96 91 80 33 91 57 17 90 83 ...
##  $ audience_rating : Factor w/ 2 levels "Spilled","Upright": 2 2 2 2 1 2 2 1 2 2 ...
##  $ audience_score  : num  73 81 91 76 27 86 76 47 89 66 ...
##  $ best_pic_nom    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_pic_win    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_actor_win  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 2 1 1 ...
```

```
##  $ best_actress_win: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_dir_win    : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ top200_box      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ director        : chr  "Michael D. Olmos" "Rob Sitch" "Christopher Guest" "Martin Scorsese" ...
##  $ actor1          : chr  "Gina Rodriguez" "Sam Neill" "Christopher Guest" "Daniel Day-Lewis" ...
##  $ actor2          : chr  "Jenni Rivera" "Kevin Harrington" "Catherine O'Hara" "Michelle Pfeiffer" .
##  $ actor3          : chr  "Lou Diamond Phillips" "Patrick Warburton" "Parker Posey" "Winona Ryder" .
##  $ actor4          : chr  "Emilio Rivera" "Tom Long" "Eugene Levy" "Richard E. Grant" ...
##  $ actor5          : chr  "Joseph Julian Soria" "Genevieve Mooy" "Bob Balaban" "Alec McCowen" ...
##  $ imdb_url        : chr  "http://www.imdb.com/title/tt1869425/" "http://www.imdb.com/title/tt020587
##  $ rt_url          : chr  "//www.rottentomatoes.com/m/filly_brown_2012/" "//www.rottentomatoes.com/m/
##  $ popularity      : num  63.7 66.5 71.3 69.9 62.6 ...
```

```r
table(movies$genre)
```

```
## 
##        Action & Adventure                 Animation 
##                        65                         9 
## Art House & International                    Comedy 
##                        14                        87 
##               Documentary                     Drama 
##                        52                       305 
##                    Horror Musical & Performing Arts 
##                        23                        12 
##        Mystery & Suspense                     Other 
##                        59                        16 
## Science Fiction & Fantasy 
##                         9
```

This gives us details on total number of movies in each genre. Drama movies being the highest number of movies and SCi-Fi movies , Animiation movies are lowest in number in the dataset.

Before that, There could be NA values which might return errors while doing calculations.It is better to check for them and get rid of those values by deleting entire row or imputing it by various means

```r
nulls <- movies %>%
  summarise_all(funs(sum(is.na(.))))

as.data.frame(nulls)
```

```
##   title title_type genre runtime mpaa_rating studio thtr_rel_year
## 1     0          0     0       1           0      8             0
##   thtr_rel_month thtr_rel_day dvd_rel_year dvd_rel_month dvd_rel_day
## 1              0            0            8             8           8
##   imdb_rating imdb_num_votes critics_rating critics_score audience_rating
## 1           0              0              0             0               0
##   audience_score best_pic_nom best_pic_win best_actor_win best_actress_win
## 1              0            0            0              0                0
##   best_dir_win top200_box director actor1 actor2 actor3 actor4 actor5
## 1            0          0        2      2      7      9     13     15
##   imdb_url rt_url popularity
## 1        0      0          0
```

There are very few null values per each column. It might not be worth the time to impute those values. Instead, we could remove the entire rows for columns studio, dvd_rel_year, ded_rel_month, dvd_rel_day.(Have not included actor1 through actor5 and director because those columns are not going to be considered for modelling anyway)

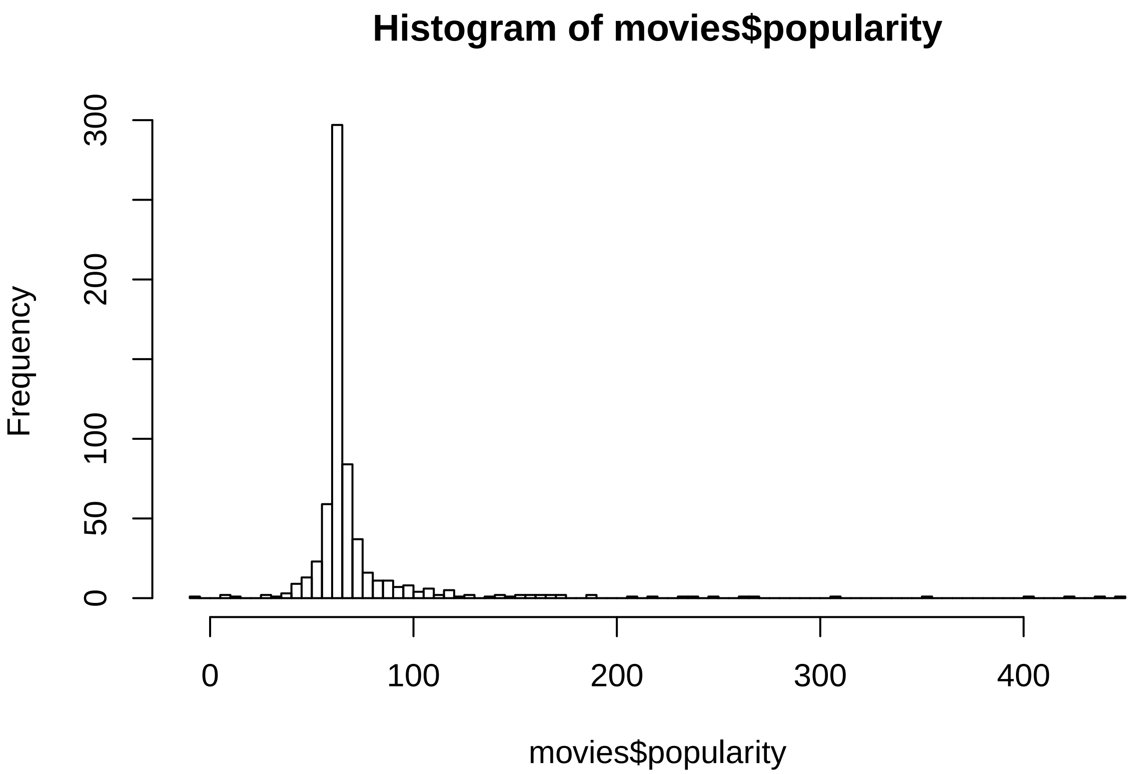
```
remove_nulls <- function(data, desiredCols) {
  completeVec <- complete.cases(data[, desiredCols])
  return(data[completeVec, ])
}

movies <- remove_nulls(movies, c('studio','dvd_rel_year','dvd_rel_month','dvd_rel_day','runtime'))
```

Let's see how the distribution of target variable is

```
summary(movies$popularity)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -8.613  61.606  63.786  72.981  67.834 447.580
```

```
hist(movies$popularity, breaks = 100)
```



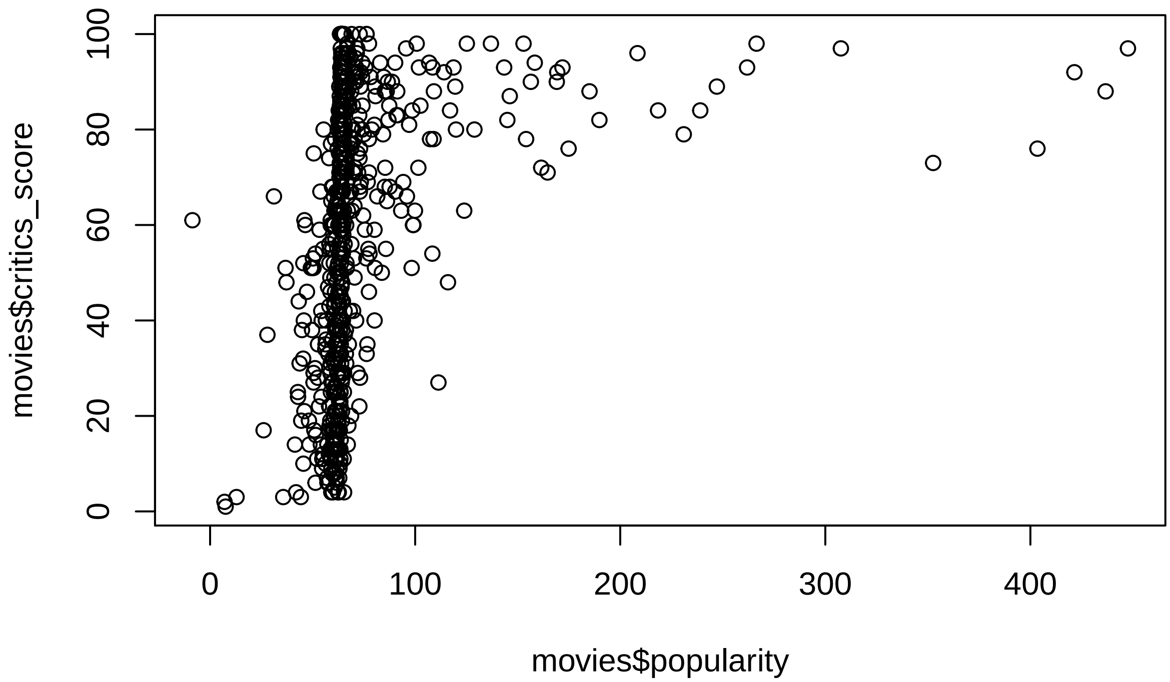
This histogram and summary stats where mean is greater than median shows that the distribution is right skewed with max values being 447.580

```
model_a <- lm(movies$popularity ~ movies$critics_score)
summary(model_a)
```

```
##
## Call:
## lm(formula = movies$popularity ~ movies$critics_score)
##
## Residuals:
##    Min      1Q Median     3Q    Max
## -82.97 -17.46  -6.81   4.99 356.81
##
## Coefficients:
```

```
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           46.55503    3.59499  12.950  < 2e-16 ***
## movies$critics_score   0.45580    0.05571   8.181 1.55e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.74 on 632 degrees of freedom
## Multiple R-squared:  0.09577,    Adjusted R-squared:  0.09434
## F-statistic: 66.93 on 1 and 632 DF,  p-value: 1.546e-15
```

```r
plot(movies$popularity,  movies$critics_score)
```



Above scatter plot shows that there seems to be no visbile trend that audience score increases/decreases as the critics score increase. From the linear model above, where R-Squared and Adjusted R-2 are close to 0.09 shows that model is terrible and addtional variables might need to be added to get a parsimonous model

---

Part 4: Modeling

Let's build a baseline model with all the remaining variables available which can help us decide what are the most useful varibales and how to get a parsimous model out of it

Firstly, we need to know what are the column names and what are their types. `str` fuction should give us that information. This can help us in first pass at getting rid of unnecessary features that might no contribute in any way to modelling like URL.

```r
str(movies)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    634 obs. of  33 variables:
##  $ title           : chr  "Filly Brown" "The Dish" "Waiting for Guffman" "The Age of Innocence" ...
##  $ title_type      : Factor w/ 3 levels "Documentary",..: 2 2 2 2 2 1 2 2 1 2 ...
##  $ genre           : Factor w/ 11 levels "Action & Adventure",..: 6 6 4 6 7 5 6 6 5 6 ...
##  $ runtime         : num  80 101 84 139 90 78 142 93 88 119 ...
```

```
##  $ mpaa_rating     : Factor w/ 6 levels "G","NC-17","PG",..: 5 4 5 3 5 6 4 5 6 6 ...
##  $ studio          : Factor w/ 211 levels "20th Century Fox",..: 91 202 167 34 13 163 147 118 88 84
##  $ thtr_rel_year   : num  2013 2001 1996 1993 2004 ...
##  $ thtr_rel_month  : num  4 3 8 10 9 1 1 11 9 3 ...
##  $ thtr_rel_day    : num  19 14 21 1 10 15 1 8 7 2 ...
##  $ dvd_rel_year    : num  2013 2001 2001 2001 2005 ...
##  $ dvd_rel_month   : num  7 8 8 11 4 4 2 3 1 8 ...
##  $ dvd_rel_day     : num  30 28 21 6 19 20 18 2 21 14 ...
##  $ imdb_rating     : num  5.5 7.3 7.6 7.2 5.1 7.8 7.2 5.5 7.5 6.6 ...
##  $ imdb_num_votes  : int  899 12285 22381 35096 2386 333 5016 2272 880 12496 ...
##  $ critics_rating  : Factor w/ 3 levels "Certified Fresh",..: 3 1 1 1 3 2 3 3 2 1 ...
##  $ critics_score   : num  45 96 91 80 33 91 57 17 90 83 ...
##  $ audience_rating : Factor w/ 2 levels "Spilled","Upright": 2 2 2 2 1 2 2 1 2 2 ...
##  $ audience_score  : num  73 81 91 76 27 86 76 47 89 66 ...
##  $ best_pic_nom    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_pic_win    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_actor_win  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 2 1 1 ...
##  $ best_actress_win: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_dir_win    : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ top200_box      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ director        : chr  "Michael D. Olmos" "Rob Sitch" "Christopher Guest" "Martin Scorsese" ...
##  $ actor1          : chr  "Gina Rodriguez" "Sam Neill" "Christopher Guest" "Daniel Day-Lewis" ...
##  $ actor2          : chr  "Jenni Rivera" "Kevin Harrington" "Catherine O'Hara" "Michelle Pfeiffer" .
##  $ actor3          : chr  "Lou Diamond Phillips" "Patrick Warburton" "Parker Posey" "Winona Ryder" .
##  $ actor4          : chr  "Emilio Rivera" "Tom Long" "Eugene Levy" "Richard E. Grant" ...
##  $ actor5          : chr  "Joseph Julian Soria" "Genevieve Mooy" "Bob Balaban" "Alec McCowen" ...
##  $ imdb_url        : chr  "http://www.imdb.com/title/tt1869425/" "http://www.imdb.com/title/tt020587
##  $ rt_url          : chr  "//www.rottentomatoes.com/m/filly_brown_2012/" "//www.rottentomatoes.com/m/
##  $ popularity      : num  63.7 66.5 71.3 69.9 62.6 ...
```

```r
apply(movies, 2, function(x) length(unique(x)))
```

```
##            title       title_type            genre          runtime
##              630                3               11               89
##      mpaa_rating           studio    thtr_rel_year   thtr_rel_month
##                6              210               43               12
##     thtr_rel_day     dvd_rel_year    dvd_rel_month      dvd_rel_day
##               31               22               12               31
##      imdb_rating   imdb_num_votes   critics_rating    critics_score
##               54              627                3               99
##  audience_rating   audience_score     best_pic_nom     best_pic_win
##                2               83                2                2
##   best_actor_win best_actress_win     best_dir_win       top200_box
##                2                2                2                2
##         director           actor1           actor2           actor3
##              518              472              559              587
##           actor4           actor5         imdb_url           rt_url
##              595              601              633              633
##       popularity
##              633
```

```r
# What are the unique values in column studio
for (i in unique(movies$title_type)) {print(i)}
```

```
## [1] "Feature Film"
```

```
## [1] "Documentary"
## [1] "TV Movie"
```

Variables `director`, `actor1`,`actor2`,`actor3`,`actor4` are basically actor names who played in the movie in the order of importance of the part they are playing. Common sense says that actors are a great contibutor to the movie success but if the data set has too many unique actors and director, model might create too many hot-encoded variables(they are considered category varaibles and for each unique value of a variable different level is created which is no help full to). Here, minimum unique value for these variables is 486 from above code chunk. So, we can get rid of these. For the same reason, we can also get rid of `title`, `studio`,`genre`. We are not getting rid of `title_type` because there are only 3 level in that categorical variables and it could be easily interpretable.

Other variables such as `imdb_url`, `rt_url` are URL of rating website. No way it can influence on movie rating.

'thtr_rel_year', here this variable will not make sense as they might be treated as numerical variables. They can be misleading. So, this could be removed. This is the same case with 'dvd_rel_year' . Where as month when the movie is released might give us some idea on seasonality during an year but it might be too difficult to interpret 12 months in a model. So, It is a good idea to bucked them as seasons like, Spring, Fall, Winter, Summer and used Season as new derived variable and remove month from the model. Same is the case with dvd_rel_month.

`dvd_rel_day` should also could be bucketed like that but spending patterns might be differnt on different days but popularity of movie prediction might not be a result of

There might be other variables which cannot influence or make very negligible influence on the score, those must be identified with a statistical methods such as forward elimation, back ward elimination etc.,

```r
to_remove <- c('director','actor1','actor2','actor3','actor4','actor5','studio','title','imdb_url','rt_u
'%ni%' <- Negate('%in%')
movies <- subset(movies,select = names(movies) %ni% to_remove)
str(movies)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    634 obs. of  16 variables:
##  $ title_type      : Factor w/ 3 levels "Documentary",..: 2 2 2 2 2 1 2 2 1 2 ...
##  $ genre           : Factor w/ 11 levels "Action & Adventure",..: 6 6 4 6 7 5 6 6 5 6 ...
##  $ runtime         : num  80 101 84 139 90 78 142 93 88 119 ...
##  $ mpaa_rating     : Factor w/ 6 levels "G","NC-17","PG",..: 5 4 5 3 5 6 4 5 6 6 ...
##  $ thtr_rel_month  : num  4 3 8 10 9 1 1 11 9 3 ...
##  $ dvd_rel_month   : num  7 8 8 11 4 4 2 3 1 8 ...
##  $ critics_rating  : Factor w/ 3 levels "Certified Fresh",..: 3 1 1 1 3 2 3 3 2 1 ...
##  $ critics_score   : num  45 96 91 80 33 91 57 17 90 83 ...
##  $ audience_rating : Factor w/ 2 levels "Spilled","Upright": 2 2 2 2 1 2 2 1 2 2 ...
##  $ best_pic_nom    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_pic_win    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_actor_win  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 2 1 1 ...
##  $ best_actress_win: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_dir_win    : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ top200_box      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ popularity      : num  63.7 66.5 71.3 69.9 62.6 ...
```

3. Change results in the new ones.

```r
thtr_rel_month_tb <- movies %>% dplyr::select(thtr_rel_month) %>%
  mutate(thtr_rel_month = ifelse(thtr_rel_month %in% c(1,2,3), 'Spring', ifelse(thtr_rel_month %in% c(4

movies$thtr_rel_month <- as.factor(pull(thtr_rel_month_tb))
```

```r
dvd_rel_month_tb <- movies %>% dplyr::select(dvd_rel_month) %>%
  mutate(dvd_rel_month = ifelse(dvd_rel_month %in% c(1,2,3), 'Spring', ifelse(dvd_rel_month %in% c(4, 5

movies$dvd_rel_month <- as.factor(pull(dvd_rel_month_tb))
```

Now we have 22 variables remaining. Now, we have to explore these varaibles in details to see for any noticable patterns

Let's build a baseline model with all these 23 variables.

```r
base_model  <- lm(popularity ~ title_type + genre + runtime + mpaa_rating
              + thtr_rel_month  + dvd_rel_month  + critics_rating + critics_score + audience_rating + be
summary(base_model)
```

```
##
## Call:
## lm(formula = popularity ~ title_type + genre + runtime + mpaa_rating +
##     thtr_rel_month + dvd_rel_month + critics_rating + critics_score +
##     audience_rating + best_pic_nom + best_pic_win + best_actor_win +
##     best_actress_win + best_dir_win + top200_box, data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -105.044  -12.529   -0.919    7.869  298.536
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     46.26300   21.45406   2.156 0.031452 *
## title_typeFeature Film          12.60142   13.78950   0.914 0.361168
## title_typeTV Movie              12.12814   22.08486   0.549 0.583100
## genreAnimation                   2.94911   14.05350   0.210 0.833857
## genreArt House & International  -4.44461   11.22170  -0.396 0.692193
## genreComedy                      5.93367    5.86347   1.012 0.311959
## genreDocumentary                 4.47292   14.44444   0.310 0.756925
## genreDrama                       0.06865    5.14861   0.013 0.989366
## genreHorror                      3.33298    8.75661   0.381 0.703617
## genreMusical & Performing Arts  -3.99567   11.87774  -0.336 0.736687
## genreMystery & Suspense          8.12657    6.55521   1.240 0.215568
## genreOther                      30.65429   10.11536   3.030 0.002547 **
## genreScience Fiction & Fantasy  12.74761   12.95120   0.984 0.325375
## runtime                          0.34293    0.08466   4.051 5.78e-05 ***
## mpaa_ratingNC-17               -30.63310   35.79688  -0.856 0.392480
## mpaa_ratingPG                   -8.53680    9.83332  -0.868 0.385660
## mpaa_ratingPG-13                -3.81786   10.10864  -0.378 0.705800
## mpaa_ratingR                    -0.35658    9.79925  -0.036 0.970985
## mpaa_ratingUnrated             -11.94484   11.31076  -1.056 0.291366
## thtr_rel_monthSpring            -4.27278    4.22994  -1.010 0.312842
## thtr_rel_monthSummer            -8.49845    4.09307  -2.076 0.038292 *
## thtr_rel_monthWinter            -3.87165    4.04498  -0.957 0.338878
## dvd_rel_monthSpring             -7.56670    4.11520  -1.839 0.066452 .
## dvd_rel_monthSummer             -3.90976    3.91189  -0.999 0.317978
## dvd_rel_monthWinter             -7.70243    4.15507  -1.854 0.064267 .
## critics_ratingFresh            -29.91835    4.18434  -7.150 2.54e-12 ***
## critics_ratingRotten           -30.02594    6.79899  -4.416 1.19e-05 ***
```

```
## critics_score                      0.05099    0.11167   0.457 0.648122
## audience_ratingUpright            12.85730    3.55611   3.616 0.000325 ***
## best_pic_nomyes                   33.18307    9.03514   3.673 0.000262 ***
## best_pic_winyes                   48.82854   15.77954   3.094 0.002064 **
## best_actor_winyes                 -2.11589    4.12045  -0.514 0.607784
## best_actress_winyes               -3.87644    4.54230  -0.853 0.393773
## best_dir_winyes                    3.89638    5.93455   0.657 0.511718
## top200_boxyes                     19.51129    9.44409   2.066 0.039260 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.1 on 599 degrees of freedom
## Multiple R-squared:  0.3691, Adjusted R-squared:  0.3333
## F-statistic: 10.31 on 34 and 599 DF,  p-value: < 2.2e-16
```

Here, R-Squared indicates that the model explains 44.79% variance in the dataset. Let's try to build a parsimonius model instead of using every variable available.

```r
set.seed(1)
model_1 <- lm(popularity ~ ., data = movies)

ols_step_backward_p(model_1)
```

```
## Backward Elimination Method
## ---------------------------
##
## Candidate Terms:
##
## 1 . title_type
## 2 . genre
## 3 . runtime
## 4 . mpaa_rating
## 5 . thtr_rel_month
## 6 . dvd_rel_month
## 7 . critics_rating
## 8 . critics_score
## 9 . audience_rating
## 10 . best_pic_nom
## 11 . best_pic_win
## 12 . best_actor_win
## 13 . best_actress_win
## 14 . best_dir_win
## 15 . top200_box
##
## We are eliminating variables based on p value...
##
## Variables Removed:
##
## - title_type
## - critics_score
## - best_actor_win
## - best_dir_win
## - best_actress_win
##
## No more variables satisfy the condition of p value = 0.3
```

```
##
##
## Final Model Output
## ------------------
##
##                         Model Summary
## ----------------------------------------------------------------
## R                       0.605       RMSE                33.996
## R-Squared               0.367       Coef. Var           46.583
## Adj. R-Squared          0.337       MSE               1155.759
## Pred R-Squared           -Inf       MAE                 16.756
## ----------------------------------------------------------------
##  RMSE: Root Mean Square Error
##  MSE: Mean Square Error
##  MAE: Mean Absolute Error
##
##                                 ANOVA
## ------------------------------------------------------------------------
##                    Sum of
##                   Squares         DF    Mean Square      F         Sig.
## ------------------------------------------------------------------------
## Regression     404612.834         28      14450.458    12.503    0.0000
## Residual       699234.255        605       1155.759
## Total         1103847.089        633
## ------------------------------------------------------------------------
##
##                                                 Parameter Estimates
## -----------------------------------------------------------------------------------------------------
##                              model       Beta    Std. Error    Std. Beta      t        Sig        lower
## -----------------------------------------------------------------------------------------------------
##                        (Intercept)     62.824        13.599                 4.620    0.000       36.118
##                     genreAnimation      1.906        13.959        0.005     0.137    0.891      -25.508
## genreArt House & International     -4.898        11.139       -0.016    -0.440    0.660      -26.775
##                        genreComedy      5.238         5.814        0.043     0.901    0.368       -6.180
##                   genreDocumentary     -6.618         7.895       -0.042    -0.838    0.402      -22.124
##                         genreDrama     -0.426         5.047       -0.005    -0.084    0.933      -10.338
##                        genreHorror      3.429         8.716        0.015     0.393    0.694      -13.687
## genreMusical & Performing Arts     -7.555        11.046       -0.025    -0.684    0.494      -29.248
##            genreMystery & Suspense      7.294         6.448        0.051     1.131    0.258       -5.370
##                         genreOther     30.434        10.067        0.111     3.023    0.003       10.664
## genreScience Fiction & Fantasy     13.036        12.901        0.035     1.011    0.313      -12.299
##                            runtime      0.342         0.081        0.159     4.220    0.000        0.183
##                   mpaa_ratingNC-17    -30.467        35.682       -0.029    -0.854    0.394     -100.542
##                      mpaa_ratingPG     -8.879         9.773       -0.082    -0.909    0.364      -28.072
##                   mpaa_ratingPG-13     -4.274        10.007       -0.042    -0.427    0.669      -23.926
##                       mpaa_ratingR     -0.442         9.716       -0.005    -0.045    0.964      -19.524
##                mpaa_ratingUnrated    -12.839        11.172       -0.081    -1.149    0.251      -34.779
##              thtr_rel_monthSpring     -4.404         4.210       -0.045    -1.046    0.296      -12.672
##              thtr_rel_monthSummer     -8.576         4.070       -0.089    -2.107    0.036      -16.569
##              thtr_rel_monthWinter     -4.063         4.023       -0.044    -1.010    0.313      -11.964
##               dvd_rel_monthSpring     -7.315         4.073       -0.077    -1.796    0.073      -15.314
##               dvd_rel_monthSummer     -3.969         3.890       -0.043    -1.020    0.308      -11.609
##               dvd_rel_monthWinter     -7.315         4.129       -0.074    -1.771    0.077      -15.425
##              critics_ratingFresh    -30.032         4.065       -0.336    -7.387    0.000      -38.015
```

```
##          critics_ratingRotten    -32.146         4.372        -0.384    -7.352    0.000      -40.733
##        audience_ratingUpright     13.340         3.426         0.158     3.894    0.000        6.612
##               best_pic_nomyes     31.808         8.884         0.140     3.580    0.000       14.361
##               best_pic_winyes     51.890        15.039         0.130     3.450    0.001       22.355
##                 top200_boxyes     19.071         9.401         0.069     2.029    0.043        0.609
## ----------------------------------------------------------------------------------------------------
##
##
##                              Elimination Summary
## ------------------------------------------------------------------------------------
##         Variable                          Adj.
## Step        Removed          R-Square    R-Square     C(p)         AIC         RMSE
## ------------------------------------------------------------------------------------
##    1    title_type             0.3682      0.3346     -4.1649    6307.1371    34.0641
##    2    critics_score          0.3681      0.3356     -6.0356    6305.2737    34.0394
##    3    best_actor_win         0.3678      0.3364     -7.7912    6303.5319    34.0181
##    4    best_dir_win           0.3673       0.337     -9.3156    6302.0341    34.0034
##    5    best_actress_win       0.3665      0.3372    -10.5678    6300.8229    33.9965
## ------------------------------------------------------------------------------------
```

```r
final_model <-lm(formula = popularity ~ genre + runtime + mpaa_rating +
      critics_rating + critics_score + best_pic_nom +
    best_pic_win + top200_box, data = movies)

summary(final_model)
```

```
##
## Call:
## lm(formula = popularity ~ genre + runtime + mpaa_rating + critics_rating +
##     critics_score + best_pic_nom + best_pic_win + top200_box,
##     data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -105.811  -12.315   -1.078    7.783  312.179
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     50.12923   15.34649   3.266 0.001150 **
## genreAnimation                   3.99132   14.10186   0.283 0.777246
## genreArt House & International  -2.37199   11.16713  -0.212 0.831860
## genreComedy                      4.37032    5.85492   0.746 0.455691
## genreDocumentary                -2.67314    7.99491  -0.334 0.738226
## genreDrama                       0.32212    5.09789   0.063 0.949638
## genreHorror                     -0.22603    8.78548  -0.026 0.979483
## genreMusical & Performing Arts  -6.42136   11.15256  -0.576 0.564980
## genreMystery & Suspense          5.42751    6.50328   0.835 0.404281
## genreOther                      30.75679   10.12447   3.038 0.002484 **
## genreScience Fiction & Fantasy  10.38136   12.99937   0.799 0.424830
## runtime                          0.35517    0.07998   4.441 1.06e-05 ***
## mpaa_ratingNC-17               -25.33506   35.92629  -0.705 0.480957
## mpaa_ratingPG                   -8.46670    9.88297  -0.857 0.391949
## mpaa_ratingPG-13                -3.82887   10.16994  -0.376 0.706684
## mpaa_ratingR                     1.07547    9.85038   0.109 0.913096
```

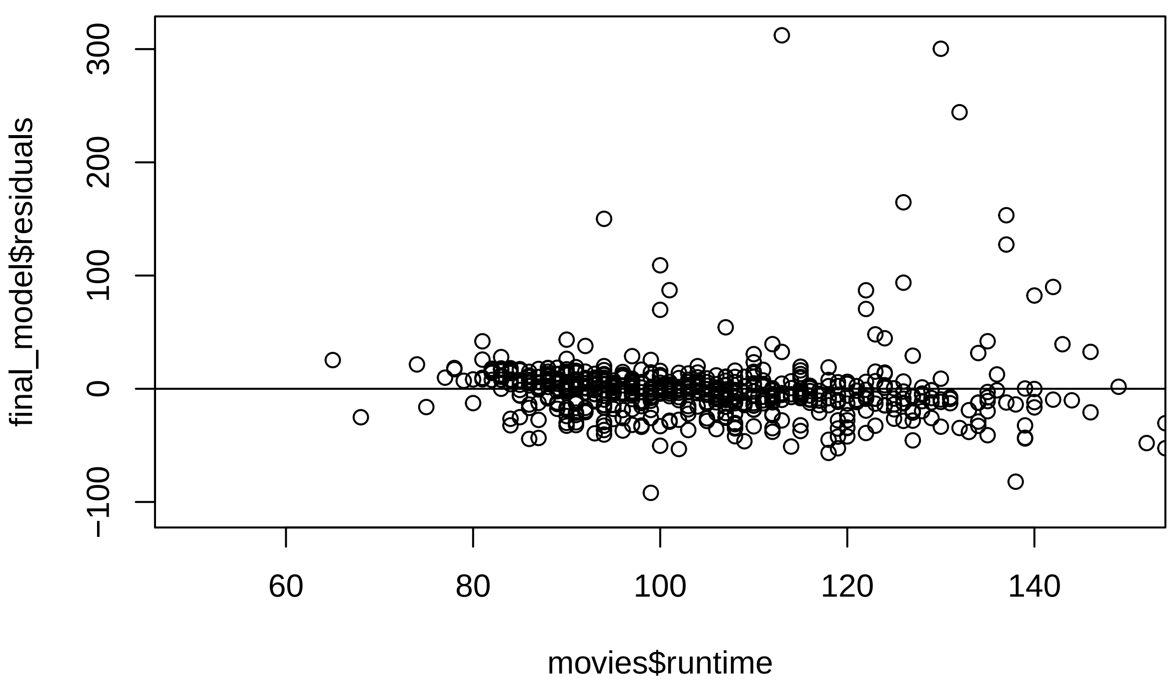
```
## mpaa_ratingUnrated            -11.96694   11.27301  -1.062 0.288857
## critics_ratingFresh           -30.15271    4.14500  -7.274 1.07e-12 ***
## critics_ratingRotten          -30.67574    6.81099  -4.504 8.00e-06 ***
## critics_score                   0.13614    0.10822   1.258 0.208866
## best_pic_nomyes                33.41794    8.90515   3.753 0.000192 ***
## best_pic_winyes                50.47567   15.08091   3.347 0.000867 ***
## top200_boxyes                  20.06001    9.47297   2.118 0.034612 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.43 on 611 degrees of freedom
## Multiple R-squared:  0.3438, Adjusted R-squared:  0.3201
## F-statistic: 14.55 on 22 and 611 DF,  p-value: < 2.2e-16
```

Adjusted R2(0.4258) is slightly higher compared to the base_model's R2(0.4156) we built using 22 predictors. Even though this is slightly better, this model is better than base model because number of predictors we are using is only 9

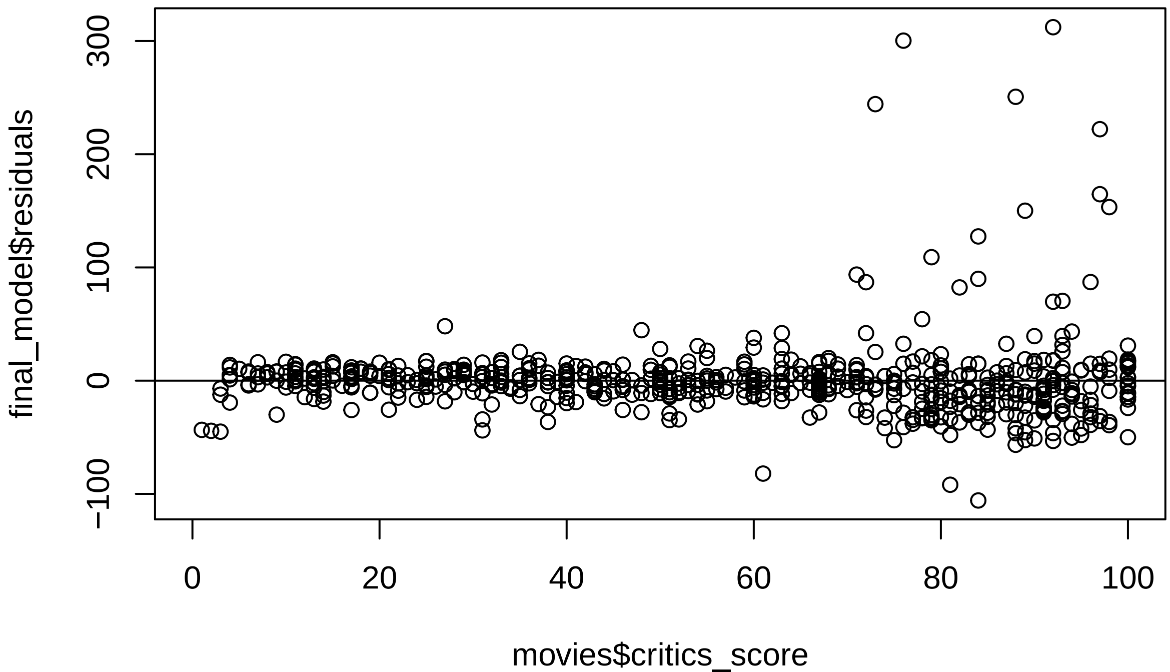
To diagonise the model, we need to check following conditions to say that model is valid .

1. Linear Relations between X and Y or random scatter. We are looking for residual to be scatter when it is plotted against numerical explanatory variables Out of 12 variables that we used to build the model, there are only three numerical variables runtime, imdb_rating, critics_score

```r
plot(final_model$residuals ~ movies$runtime, xlim= c(50,150))
abline(0, 0)
```

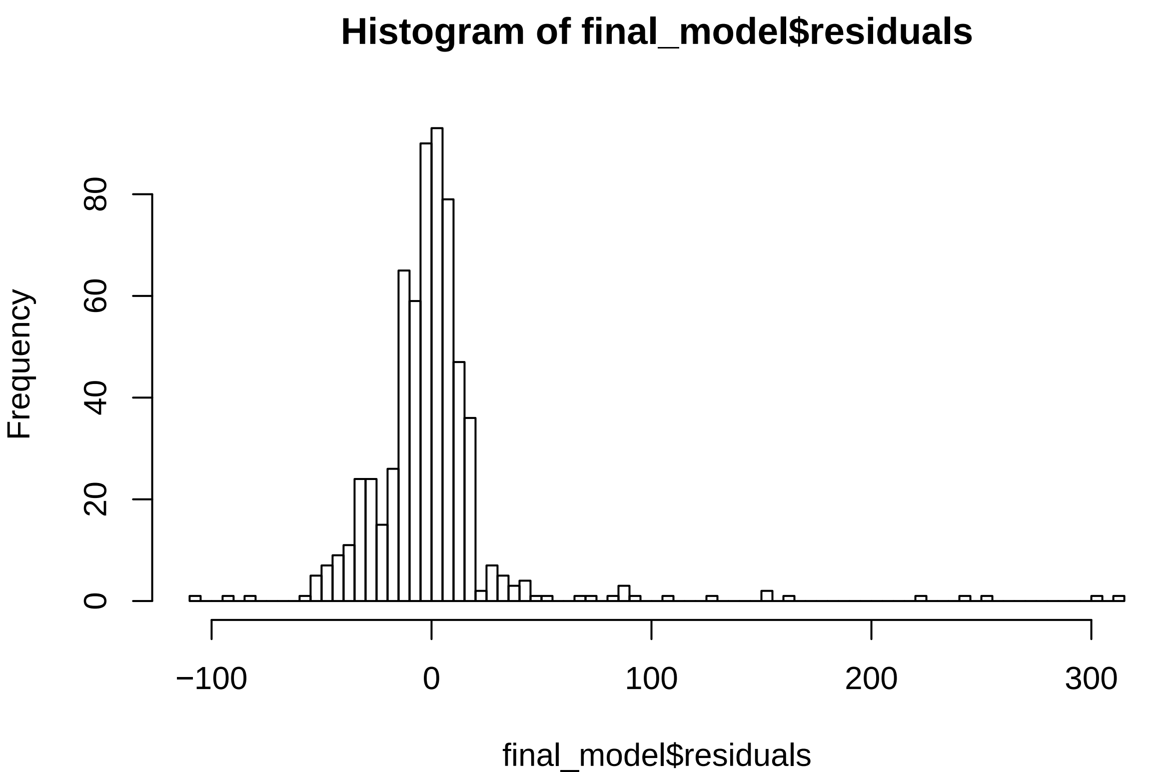



```r
plot(final_model$residuals ~ movies$critics_score, xlim= c(0,100))
abline(0, 0)
```



In all the three residual plots above, we don't see a fan-shapped data. They are pretty scattered around zero which means we are satisfying our condition here. 2. Nearly normal residuals: Let's check if the residuals are normally distributed here.

```r
hist(final_model$residuals, breaks = 100)
```




```r
summary(final_model$residuals)
```

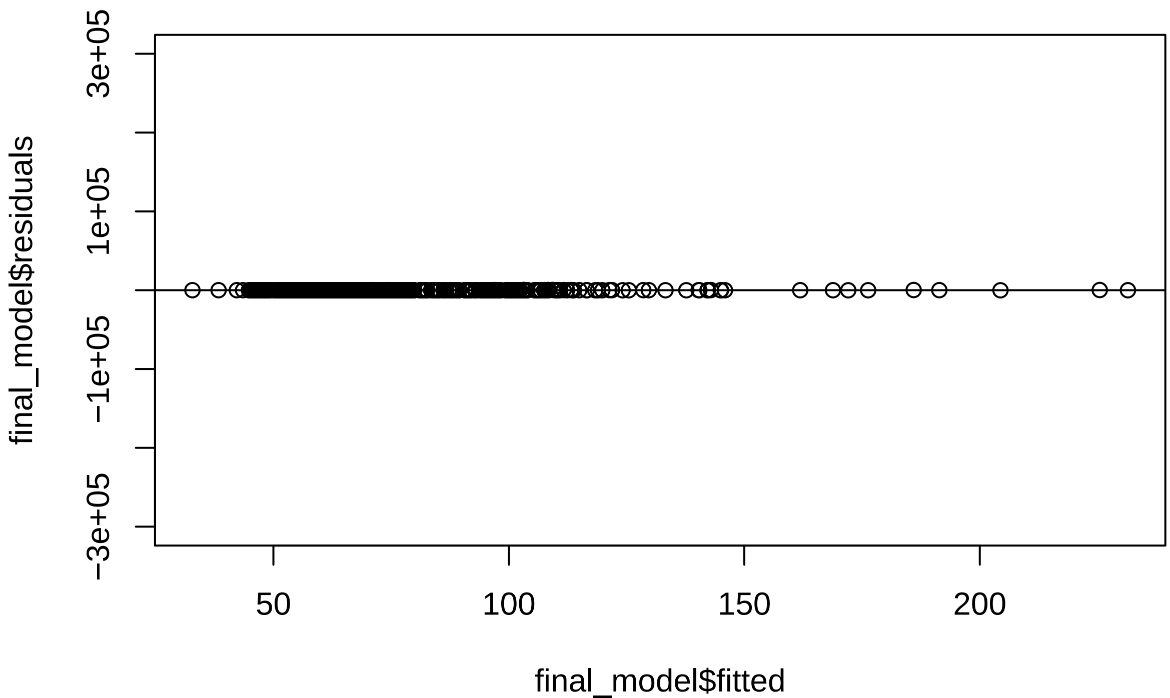
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
```

```
## -105.811  -12.315   -1.078    0.000    7.783  312.179
```

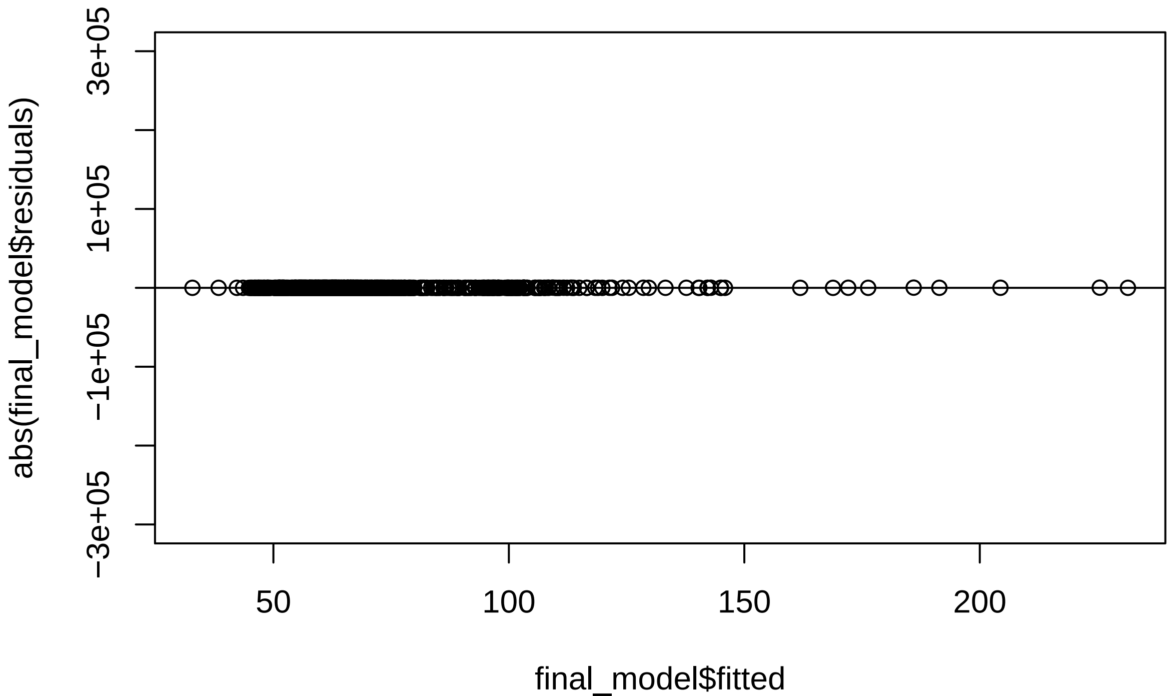
Here, the histogram shows that residuals are normally distributed with skew to the right and the mean is zero. This condition is also satisfied.

3. Constant variablity

```r
plot(final_model$residuals ~ final_model$fitted, ylim = c(-300000, 300000))
abline(0, 0)
```




```r
plot(abs(final_model$residuals) ~ final_model$fitted, ylim = c(-300000, 300000))
abline(0, 0)
```

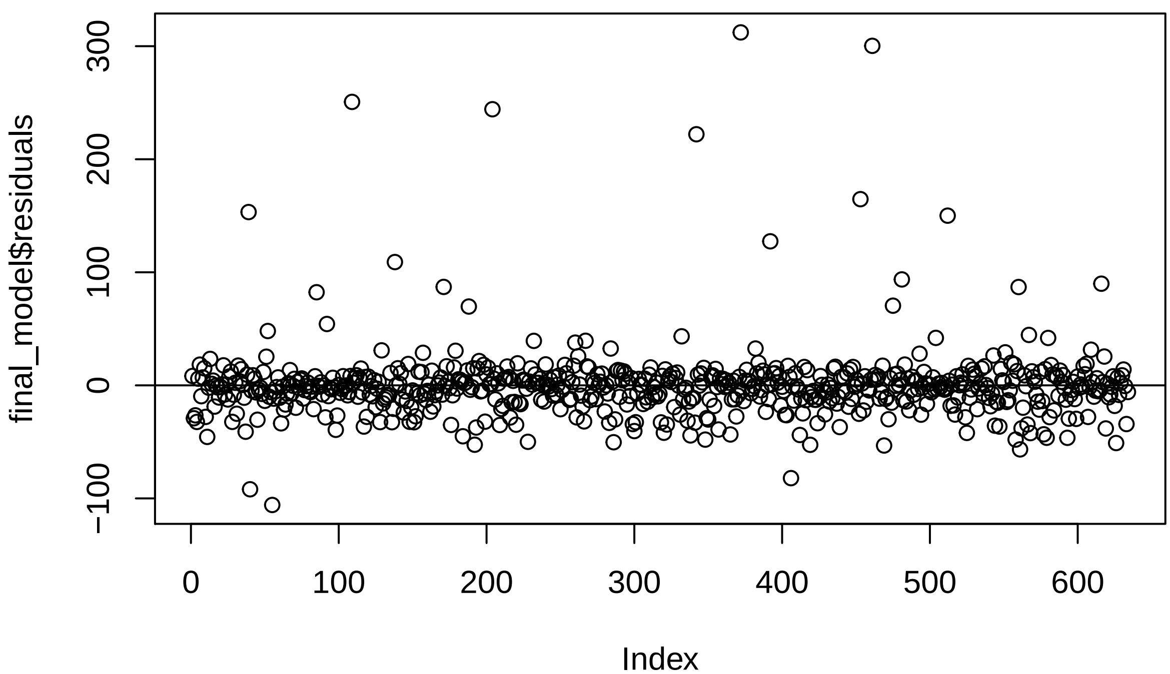


Here, the residual plot is fan-shapped that means it doesn't have constant variabilty. For lower values of Y predicted, predictions are more reliable than higher values. This condition is not satisfied

4. Independent residuals/observations

This is not exactly a time series data. So, we can that the observations are independent variables as they are randomly sampled from a pool of movies.If we look at the residual distributions, they are pretty random too. Hence, this condition is satisfied.

```
plot(final_model$residuals)
abline(0, 0)
```



```r
summary(final_model)
```

```
##
## Call:
## lm(formula = popularity ~ genre + runtime + mpaa_rating + critics_rating +
##     critics_score + best_pic_nom + best_pic_win + top200_box,
##     data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -105.811  -12.315   -1.078    7.783  312.179
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    50.12923   15.34649   3.266 0.001150 **
## genreAnimation                  3.99132   14.10186   0.283 0.777246
## genreArt House & International -2.37199   11.16713  -0.212 0.831860
## genreComedy                     4.37032    5.85492   0.746 0.455691
## genreDocumentary               -2.67314    7.99491  -0.334 0.738226
## genreDrama                      0.32212    5.09789   0.063 0.949638
## genreHorror                    -0.22603    8.78548  -0.026 0.979483
## genreMusical & Performing Arts -6.42136   11.15256  -0.576 0.564980
## genreMystery & Suspense         5.42751    6.50328   0.835 0.404281
## genreOther                     30.75679   10.12447   3.038 0.002484 **
## genreScience Fiction & Fantasy 10.38136   12.99937   0.799 0.424830
## runtime                         0.35517    0.07998   4.441 1.06e-05 ***
## mpaa_ratingNC-17              -25.33506   35.92629  -0.705 0.480957
## mpaa_ratingPG                  -8.46670    9.88297  -0.857 0.391949
## mpaa_ratingPG-13               -3.82887   10.16994  -0.376 0.706684
## mpaa_ratingR                    1.07547    9.85038   0.109 0.913096
## mpaa_ratingUnrated            -11.96694   11.27301  -1.062 0.288857
## critics_ratingFresh           -30.15271    4.14500  -7.274 1.07e-12 ***
## critics_ratingRotten          -30.67574    6.81099  -4.504 8.00e-06 ***
## critics_score                   0.13614    0.10822   1.258 0.208866
## best_pic_nomyes                33.41794    8.90515   3.753 0.000192 ***
## best_pic_winyes                50.47567   15.08091   3.347 0.000867 ***
## top200_boxyes                  20.06001    9.47297   2.118 0.034612 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.43 on 611 degrees of freedom
## Multiple R-squared:  0.3438, Adjusted R-squared:  0.3201
## F-statistic: 14.55 on 22 and 611 DF,  p-value: < 2.2e-16
```

Part 5: Prediction

To test this model using a movie released in 2016, I used 'Captain America: Civil War' as target movie. Information about this movie is obtained from IMDB.com and rottentomatoes.com

Below are the variables for this movie:

1. runtime : 147

2. mpaa_rating : PG-13
3. thtr_rel_year : 2016
4. genre : Science Fiction & Fantasy
5. critics_rating : Fresh
6. citics_score : 91
7. best_pic_nom : yes
8. best_pic_win : no
9. top200_box : yes
10. imdb_ratings : 7.8
11. imdb_num_votes : 506314
12. audience_score : 89

```
test_movie <- data.frame(title="Captain America : Civil War",critics_score=91, genre="Science Fiction &
,imdb_rating = 7.8, imdb_num_votes = 506314, audience_score= 89 )

test_movie$popularity <- ((test_movie$imdb_rating * 10) + test_movie$audience_score) / 2


test_movie$popularity <- (test_movie$imdb_num_votes / quantile(test_movie$imdb_num_votes, 0.75))* test_m

cat("Predicted Popularity:", predict(final_model, test_movie))
```

```
## Predicted Popularity: 124.5457
```

```
cat("\nActual Popularity: ", test_movie$popularity)
```

```
##
## Actual Popularity:  83.5
```

Part 6: Conclusion

Considering the movies$popularity ranging from -8 to 447.58, `Captain America : Civil War` movie prediction being around 40 units more than what actual calculated value is not bad. If we reduce the scale of target variable from -8 to 447.58 to 0 and 1, then predicted value and actual values are going to be 0.19 and 0.28, which gives us directionally good information on where this movie stands in terms of our popularity score.

More than accuracy, it is more intersting to look at the parameters that are influencing the prediction. Things like being in top 200 among box offices, critics_score, genre will definetly make an impact on the movie popularity. It is clear from the R2 value that we need more varibales which can go into model to get more accurate model. Even though the model is statiscally significant as we used back ward elimination P-value method to get rid of varibales, we can only explain around 36% of variablity in the data with variables that are used in the final model. As most of the variables used in the final model are categorical, it might be help full to have more numerical data for us to get better accuracy. Other variables that could be used are budget of the movie, promotional ad spent, trailer views on youtube, social media followers etc.,