Modeling and prediction for movies

Setup

Load packages

```
library(ggplot2)
library(dplyr)
library(statsr)
```

Load data

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

Part 1: Data

The data set is comprised of 651 randomly sampled movies produced and released before 2016.

Part 2: Research question

What is the parsimonious model for predicting how much audiences like movies (variable: audience_score) from numerous variables about the movies?

In general, many reasons would impact audience's feeling about a movie. In this study, I will focus on 8 variables which are of high interest for me. The 8 variables I will discuss are the genre of movie (variable_1: genre), the MPAA rating of the movie (variable_2: mpaa_rating), the runtime of movie (variable_3: runtime), critics score (variable_4: critics_score), whether or not the movie was nominated for best picture Oscar (variable_5: best_pic_nom), whether or not one of the main actors or actress in the movie won an Oscar (variable_6: best_actor_win, variable_7: best_actress_win), and whether or not the direct or of the movie even won an Oscar (Variable_8: best_dir_win).

Part 3: Exploratory data analysis

Get familar and clear the 9 variables (1 response variable, 8 explanatory variables)

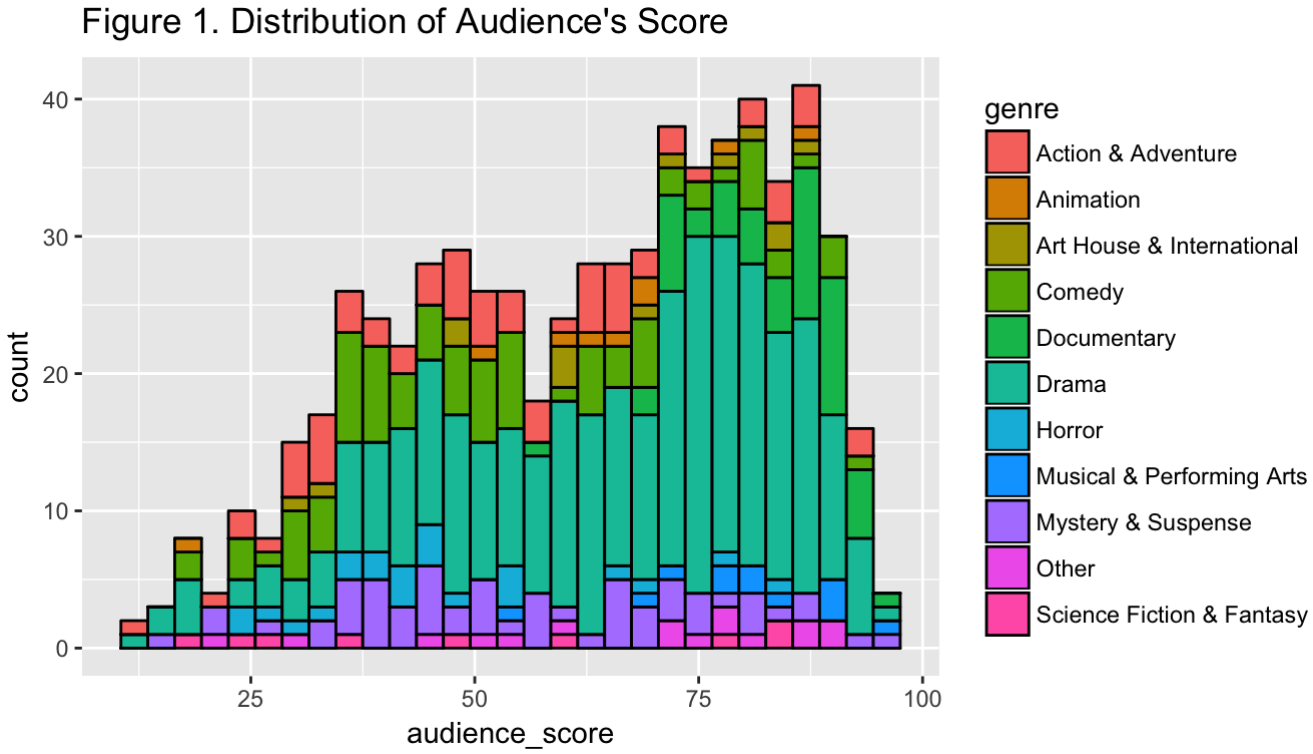
I will remove the rows with missing values.

```
#remove all rows with any missing values from the 9 variables
movies <- movies %>%
  filter(!is.na(audience_score), !is.na(genre), !is.na(mpaa_rating), !is.na(runtime),
         !is.na(critics_score), !is.na(best_pic_nom), !is.na(best_actor_win), !is.na(bes
t_actress_win), !is.na(best_dir_win))
```

```
#get familar with the response variable: audience_score
summary(movies$audience_score)
```

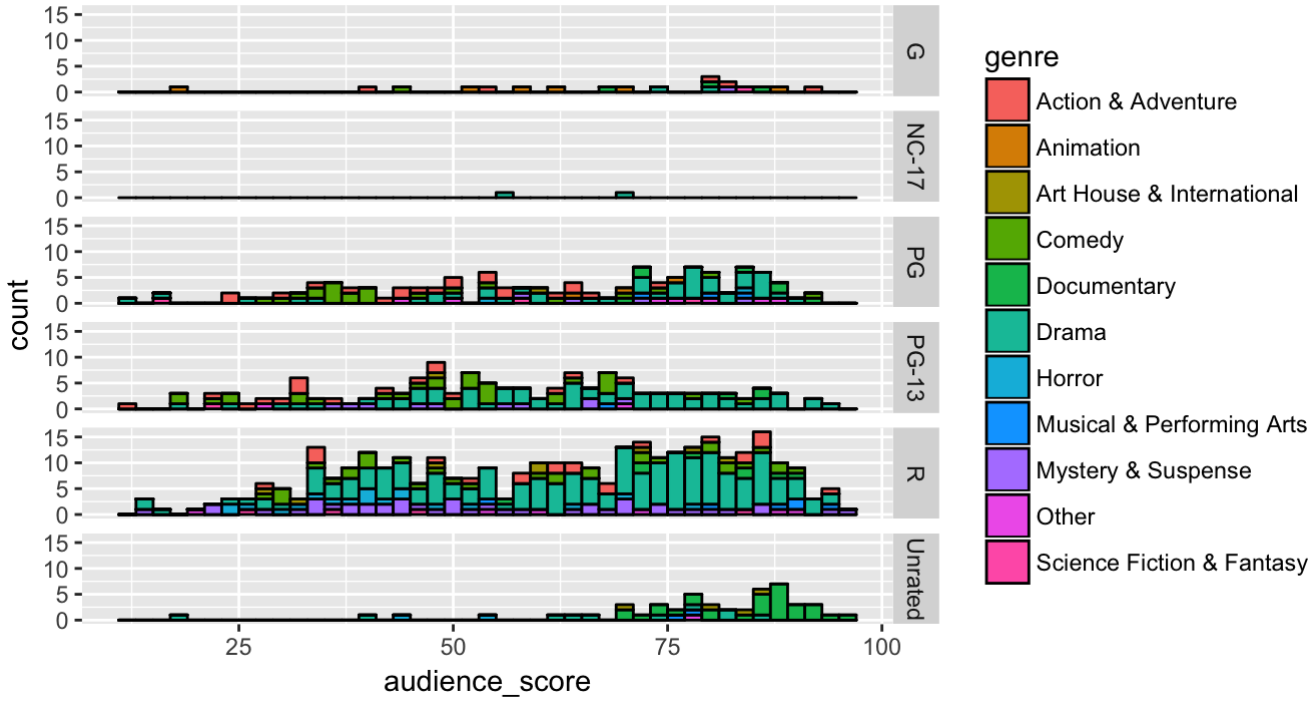
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   46.00   65.00   62.35   80.00   97.00
```

```
#get familar with the relationships between the response variable and explanatory variab
les (genre,mpaa_rating)
ggplot(data=movies, aes(x=audience_score, fill=genre)) +
  geom_histogram(binwidth=3, colour="Black") +
  ggtitle("Figure 1. Distribution of Audience's Score")
```




```
ggplot(data=movies, aes(x=audience_score, fill=genre)) +
  geom_histogram(binwidth=2, colour="Black") +
  facet_grid(mpaa_rating ~.) +
  ggtitle("Figure 2. Distribution of Audience's Score with Different MPAA Ratings")
```

Figure 2. Distribution of Audience's Score with Different MPAA Ratings



From Figure 1, the distribution of audience' scores are a bit left skewed, however not too bad. In addition, we could see that most of the movies are drama.

From Figure 2, referring to the MPAA Ratings, most of the movies are identified as "R". While the movies identified as "MC-17" are very rare.

Finally, since these 2 explanatory variables are categorical, I won't check the linear relationships between each explanatory variable and the response variable.

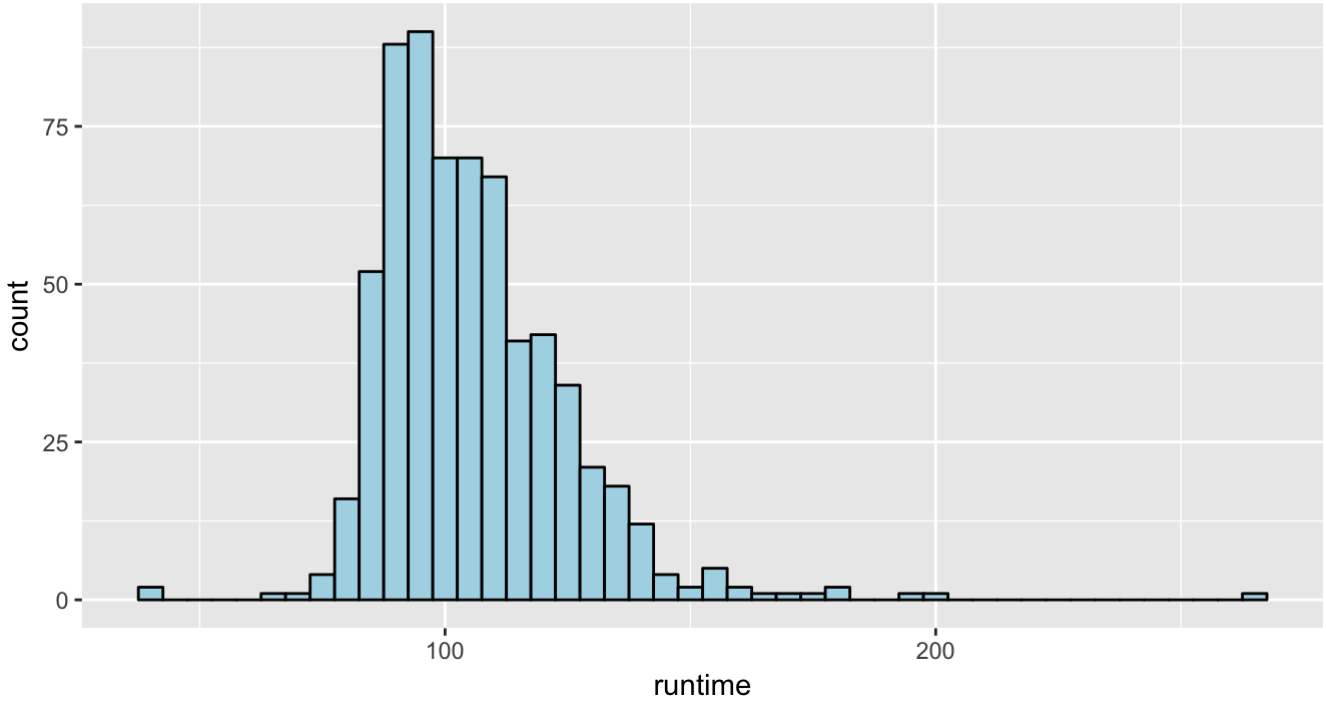
Simple linear regression one: runtime VS audience_score

```
#get familar with the continuous explanatory variable runtime, and check for outliers, a
nd clear the variable
summary(movies$runtime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    39.0    92.0   103.0   105.8   115.8   267.0
```

```
ggplot(data=movies %>% filter(!is.na(runtime)), aes(x=runtime)) +
  geom_histogram(binwidth=5, fill="LightBlue", colour="Black") +
  ggtitle("Figure 3. Distribution of Movie Run Time")
```

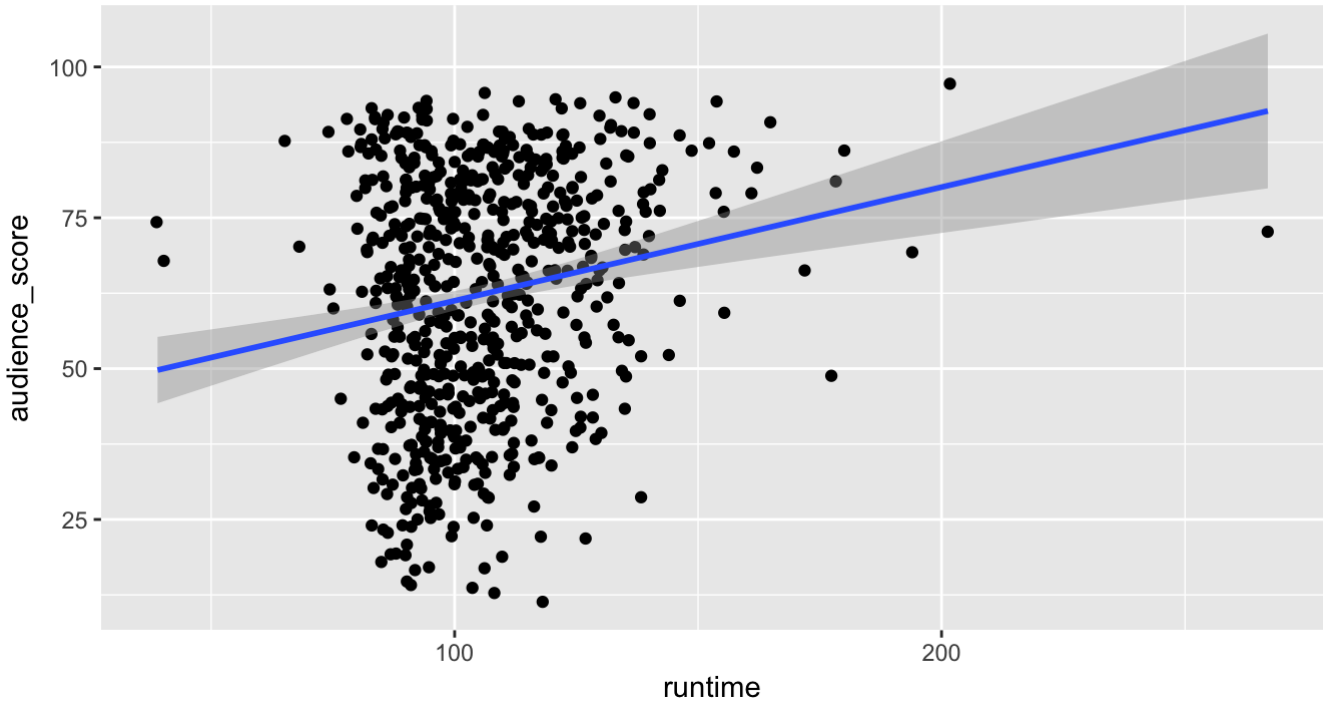
Figure 3. Distribution of Movie Run Time



From the Figure 3 above, the distribution of the movie runtime looks a bit right skewed, and with several extremes, but not bad.

```
#check whether the explanatory variable runtime is linearily related with the response v
ariable
ggplot(data=movies, aes(x=runtime, y=audience_score)) +
  geom_jitter() +
  geom_smooth(method="lm") +
  ggtitle("Figure 4. Scatter Plot of Movie Run Time and Audience Score")
```

Figure 4. Scatter Plot of Movie Run Time and Audience Score



```
movies %>%
  summarise(cor(runtime, audience_score))
```

```
## # A tibble: 1 x 1
##   `cor(runtime, audience_score)`
##                            <dbl>
## 1                          0.181
```

Both the scatter plot and correlation value R indicate that the explanatory variable runtime and response variable audience score are not strongly linearily related.

```
#create the simple liner regression predicting audience_score from runtime
m1 <- lm(audience_score ~ runtime, data=movies)
summary(m1)
```

```
##
## Call:
## lm(formula = audience_score ~ runtime, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.641 -15.626   3.008  17.080  34.950
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.4203     4.3256   9.807  < 2e-16 ***
## runtime       0.1883     0.0402   4.684 3.43e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.92 on 648 degrees of freedom
## Multiple R-squared:  0.03275,    Adjusted R-squared:  0.03125
## F-statistic: 21.94 on 1 and 648 DF,  p-value: 3.431e-06
```

The Adjusted R-square value is 3%, which is very low, indicates a weak model.

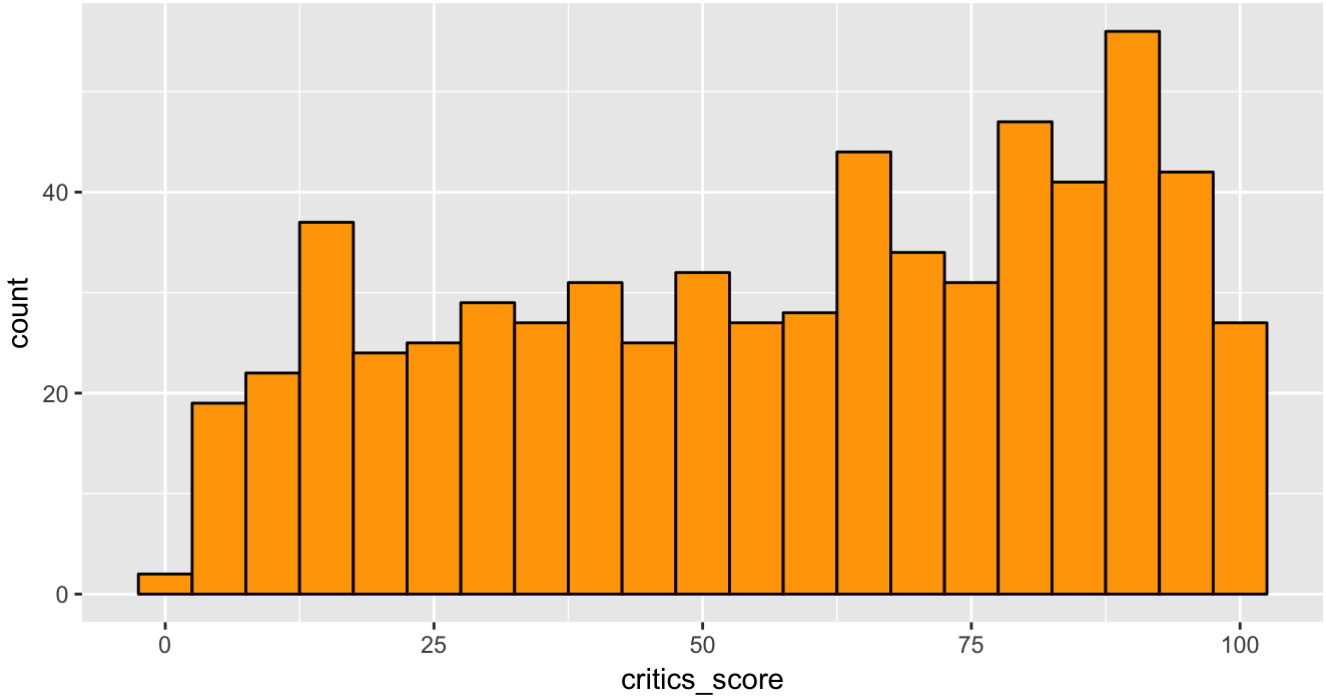
Simple linear regression two: critics_score VS audience_score

```
#get familar with the continuous explanatory variable crtics_score
summary(movies$critics_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   33.00   61.00   57.65   83.00  100.00
```

```
ggplot(data=movies, aes(x=critics_score)) +
  geom_histogram(binwidth=5, fill="Orange", colour="Black") +
  ggtitle("Figure 5. Distribution of Critics Score")
```

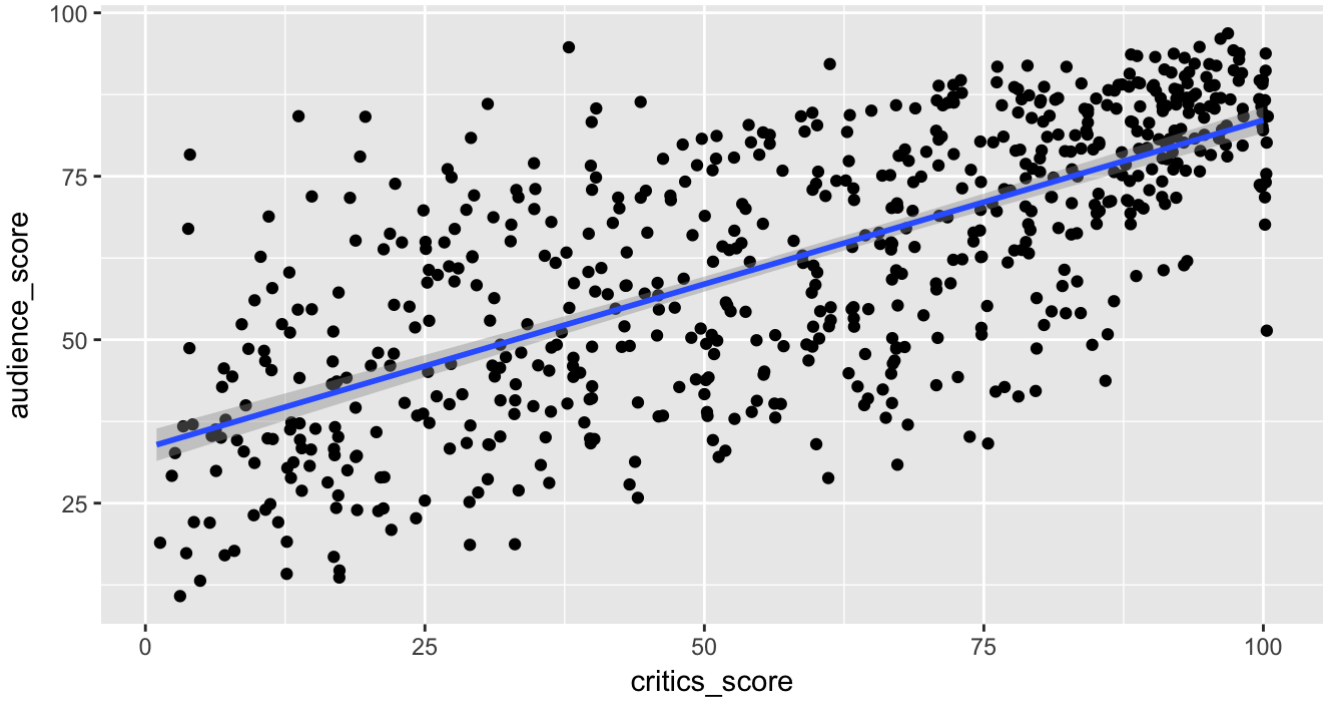
Figure 5. Distribution of Critics Score



From the Figure 5, the variable critics_score looks uniform, with no outliers.

```
#check whether the explanatory variable crtics_score is linearily related with the respo
nse variable
ggplot(data=movies, aes(x=critics_score, y=audience_score)) +
  geom_jitter() +
  geom_smooth(method="lm") +
  ggtitle("Figure 6. Scatter Plot of critics_score and Audience Score")
```

Figure 6. Scatter Plot of critics_score and Audience Score



```
movies %>%
  summarise(cor(critics_score, audience_score))
```

```
## # A tibble: 1 x 1
##   `cor(critics_score, audience_score)`
##                                  <dbl>
## 1                                0.704
```

From Figure 6, as expected the relationship between critics score and audience score is quite strong. The blue line is the model. The shaded gray area around the line tells us the variability we might expect in our predictions. It looks ok. The correlation value R also indicates a quite strong linear relationship of 0.7.

```
#create the simple liner regression predicting audience_score from critics_score
m2 <- lm(audience_score ~ critics_score, data=movies)
summary(m2)
```

```
##
## Call:
## lm(formula = audience_score ~ critics_score, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.047  -9.580   0.711  10.418  43.545
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   33.43408    1.27665   26.19   <2e-16 ***
## critics_score  0.50150    0.01987   25.25   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.38 on 648 degrees of freedom
## Multiple R-squared:  0.4958, Adjusted R-squared:  0.4951
## F-statistic: 637.3 on 1 and 648 DF,  p-value: < 2.2e-16
```

The Adjusted R-square is 49.52%, looks good.

```
#get familar with other 4 categorical variables, best_pic_nom, best_actor_win, best_actr
ess_win, and best_dir_win
#1. best_pic_nom
movies %>%
  group_by(best_pic_nom) %>%
  summarise(count=n())
```

```
## # A tibble: 2 x 2
##   best_pic_nom count
##   <fct>        <int>
## 1 no             628
## 2 yes             22
```

```
#2. best_actor_win
movies %>%
  group_by(best_actor_win) %>%
  summarise(count=n())
```

```
## # A tibble: 2 x 2
##   best_actor_win count
##   <fct>          <int>
## 1 no               557
## 2 yes               93
```

```
#3. best_actress_win
movies %>%
  group_by(best_actress_win) %>%
  summarise(count=n())
```

```
## # A tibble: 2 x 2
##   best_actress_win count
##   <fct>            <int>
## 1 no                 578
## 2 yes                 72
```

```
#4. best_dir_win
movies %>%
  group_by(best_dir_win) %>%
  summarise(count=n())
```

```
## # A tibble: 2 x 2
##   best_dir_win count
##   <fct>        <int>
## 1 no             607
## 2 yes             43
```

Part 4: Modeling

Multiple linear regression, all 8 explanatory variables VS audience_score

Step 1 model, pick the model with highest Adjust R-square with 7 explanatory variables

```
full_model <- lm(audience_score ~ genre + mpaa_rating + runtime + critics_score + best_p
ic_nom +
                   best_actor_win + best_actress_win + best_dir_win, data=movies)
summary(full_model)
```

```
##
## Call:
## lm(formula = audience_score ~ genre + mpaa_rating + runtime +
##     critics_score + best_pic_nom + best_actor_win + best_actress_win +
##     best_dir_win, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.925  -8.724   0.496   9.019  40.511
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    30.45231    4.99379   6.098 1.88e-09 ***
## genreAnimation                  4.76350    5.43925   0.876  0.38149
## genreArt House & International  6.05549    4.21007   1.438  0.15084
## genreComedy                    -0.23346    2.32776  -0.100  0.92014
## genreDocumentary               10.55201    3.15405   3.346  0.00087 ***
## genreDrama                      1.96563    2.02654   0.970  0.33245
## genreHorror                    -8.60164    3.47500  -2.475  0.01358 *
## genreMusical & Performing Arts 10.30645    4.47868   2.301  0.02171 *
## genreMystery & Suspense        -4.07155    2.61229  -1.559  0.11959
## genreOther                      1.37374    3.95303   0.348  0.72832
## genreScience Fiction & Fantasy -6.88180    4.96554  -1.386  0.16627
## mpaa_ratingNC-17              -11.86636   10.57008  -1.123  0.26202
## mpaa_ratingPG                  -2.32631    3.84474  -0.605  0.54536
## mpaa_ratingPG-13               -3.39588    3.95404  -0.859  0.39076
## mpaa_ratingR                   -1.84726    3.81159  -0.485  0.62810
## mpaa_ratingUnrated             -3.57441    4.35333  -0.821  0.41192
## runtime                         0.07355    0.03307   2.224  0.02651 *
## critics_score                   0.43798    0.02276  19.243  < 2e-16 ***
## best_pic_nomyes                 9.61051    3.26223   2.946  0.00334 **
## best_actor_winyes              -1.51852    1.65537  -0.917  0.35932
## best_actress_winyes            -2.15047    1.83379  -1.173  0.24136
## best_dir_winyes                -0.32060    2.30845  -0.139  0.88959
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.93 on 628 degrees of freedom
## Multiple R-squared:  0.5416, Adjusted R-squared:  0.5263
## F-statistic: 35.34 on 21 and 628 DF,  p-value: < 2.2e-16
```

From the output, some of the variables are not statistically significant to predict the audience scores. The adjust R-square is 52.63%.

Next, I will use the backwise method to remove the variable one by one from the full model, I will record the adjusted R-square, and pick the model with the highest increase in the adjusted R-squaer, and repeat until none of th models yield an increase in adjust R-square.

```
#remove variable best_dir_win based on full model
Step1_model_dir <- lm(audience_score ~ genre + mpaa_rating + runtime + critics_score + b
est_pic_nom +
                   best_actor_win + best_actress_win, data=movies)
summary(Step1_model_dir)$adj.r.squared
```

```
## [1] 0.527033
```

```
#remove variable best_actress_win based on full model
Step1_model_actress <- lm(audience_score ~ genre + mpaa_rating + runtime + critics_score
 + best_pic_nom +
                   best_actor_win + best_dir_win, data=movies)
summary(Step1_model_actress)$adj.r.squared
```

```
## [1] 0.5260119
```

```
#remove variable best_actor_win based on full model
Step1_model_actor <- lm(audience_score ~ genre + mpaa_rating + runtime + critics_score +
 best_pic_nom +
                  best_actress_win + best_dir_win, data=movies)
summary(Step1_model_actor)$adj.r.squared
```

```
## [1] 0.5264138
```

```
#remove variable best_pic_nom based on full model
Step1_model_pic <- lm(audience_score ~ genre + mpaa_rating + runtime + critics_score +
                   best_actor_win + best_actress_win + best_dir_win, data=movies)
summary(Step1_model_pic)$adj.r.squared
```

```
## [1] 0.5205114
```

```
#remove variable critics_score based on full model
Step1_model_critics_score <- lm(audience_score ~ genre + mpaa_rating + runtime + best_pi
c_nom +
                   best_actor_win + best_actress_win + best_dir_win, data=movies)
summary(Step1_model_critics_score)$adj.r.squared
```

```
## [1] 0.2481648
```

From the output, after the variable critics_score was removed, the adjust R-square is 5.95%, decreased significantly compared to the full model. This is consistent with the model 2 that the variable critics_score contributes 50% variability to the model 2.

```
#remove variable runtime based on full model
Step1_model_runtime <- lm(audience_score ~ genre + mpaa_rating + critics_score + best_pi
c_nom +
                    best_actor_win + best_actress_win + best_dir_win, data=movies)
summary(Step1_model_runtime)$adj.r.squared
```

```
## [1] 0.5233231
```

```
#remove variable mpaa_rating based on full model
Step1_model_mpaa_rating <- lm(audience_score ~ genre + runtime + critics_score + best_pi
c_nom +
                    best_actor_win + best_actress_win + best_dir_win, data=movies)
summary(Step1_model_mpaa_rating)$adj.r.squared
```

```
## [1] 0.5279892
```

```
#remove variable genre based on full model
Step1_model_genre <- lm(audience_score ~ mpaa_rating + runtime + critics_score + best_pi
c_nom +
                    best_actor_win + best_actress_win + best_dir_win, data=movies)
summary(Step1_model_genre)$adj.r.squared
```

```
## [1] 0.5026506
```

```
#compare the adjusted R-square after each varaible was removed, and the full model
FULL_MODEL <- c("audience ~ genre + mpaa + runtime + critics + pic + actor + actress + d
ir", "NULL", "52.63%")
Step1_M1<- c("audience ~ genre + mpaa + runtime + critics + pic + actor + actress", "di
r", "52.7033%")

Step1_M2 <- c("audience ~ genre + mpaa + runtime + critics + pic + actor + dir", "actres
s", "52.60119%")

Step1_M3 <- c("audience ~ genre + mpaa + runtime + critics + pic + actress + dir", "acto
r", "52.64138%")

Step1_M4 <- c("audience ~ genre + mpaa + runtime + critics + actor + actress + dir", "pi
c", "52.05114%")

Step1_M5 <- c("audience ~ genre + mpaa + runtime + pic + actor + actress + dir", "critic
s", "24.81648%")

Step1_M6 <- c("audience ~ genre + mpaa + critics + pic + actor + actress + dir", "runtim
e", "52.32901%")

Step1_M7 <- c("audience ~ genre + runtime + critics + pic + actor + actress + dir", "mpa
a", "52.79892%")

Step1_M8 <- c("audience ~ mpaa + runtime + critics + pic + actor + actress + dir", "genr
e", "50.26506")
```

```
#adjusted R-square comparisons with one variable removed each time
STEP_1_Models <- rbind(FULL_MODEL, Step1_M1,Step1_M2, Step1_M3, Step1_M4, Step1_M5, Step
1_M6, Step1_M7, Step1_M8)

colnames(STEP_1_Models) <- c("Model", "Variable Removed", "Adjusted R-square")
STEP_1_Models
```

```
##            Model

## FULL_MODEL "audience ~ genre + mpaa + runtime + critics + pic + actor + actress + di
r"
## Step1_M1   "audience ~ genre + mpaa + runtime + critics + pic + actor + actress"

## Step1_M2   "audience ~ genre + mpaa + runtime + critics + pic + actor + dir"

## Step1_M3   "audience ~ genre + mpaa + runtime + critics + pic + actress + dir"

## Step1_M4   "audience ~ genre + mpaa + runtime + critics + actor + actress + dir"

## Step1_M5   "audience ~ genre + mpaa + runtime + pic + actor + actress + dir"

## Step1_M6   "audience ~ genre + mpaa + critics + pic + actor + actress + dir"

## Step1_M7   "audience ~ genre + runtime + critics + pic + actor + actress + dir"

## Step1_M8   "audience ~ mpaa + runtime + critics + pic + actor + actress + dir"

##            Variable Removed Adjusted R-square
## FULL_MODEL "NULL"           "52.63%"
## Step1_M1   "dir"            "52.7033%"
## Step1_M2   "actress"        "52.60119%"
## Step1_M3   "actor"          "52.64138%"
## Step1_M4   "pic"            "52.05114%"
## Step1_M5   "critics"        "24.81648%"
## Step1_M6   "runtime"        "52.32901%"
## Step1_M7   "mpaa"           "52.79892%"
## Step1_M8   "genre"          "50.26506"
```

From the above output, we can determine that when the variable mpaa_rating was removed, the Adjusted R-square was highest.

Next, based on the model when mpaa_rating was removed, I will remove the variable one by one again, and pick the model with the highest Adjusted R-square.

Step 2 model, pick the model with highest Adjust R-square with 6 explanatory variables

```
#remove variable best_dir_win, based on Step1 model
Step2_model_dir <- lm(audience_score ~ genre + runtime + critics_score + best_pic_nom +
                  best_actor_win + best_actress_win, data=movies)
summary(Step2_model_dir)$adj.r.squared
```

```
## [1] 0.5287301
```

```
#remove variable best_actress_win based on Step1 model
Step2_model_actress <- lm(audience_score ~ genre + runtime + critics_score + best_pic_no
m +
                    best_actor_win + best_dir_win, data=movies)
summary(Step2_model_actress)$adj.r.squared
```

```
## [1] 0.5276834
```

```
#remove variable best_actor_win based on Step1 model
Step2_model_actor <- lm(audience_score ~ genre + runtime + critics_score + best_pic_nom
+
                   best_actress_win + best_dir_win, data=movies)
summary(Step2_model_actor)$adj.r.squared
```

```
## [1] 0.5280099
```

From the output, after the variable best_actor_win was removed, the adjust R-square is 50.16%, decreased slightly compared to the full model.

```
#remove variable best_pic_nom based on Step1 model
Step2_model_pic <- lm(audience_score ~ genre + runtime + critics_score +
                    best_actor_win + best_actress_win + best_dir_win, data=movies)
summary(Step2_model_pic)$adj.r.squared
```

```
## [1] 0.5221492
```

```
#remove variable critics_score based on Step1 model
Step2_model_critics_score <- lm(audience_score ~ genre + runtime + best_pic_nom +
                    best_actor_win + best_actress_win + best_dir_win, data=movies)
summary(Step2_model_critics_score)$adj.r.squared
```

```
## [1] 0.2335755
```

```
#remove variable runtim based on Step1 model
Step2_model_runtime <- lm(audience_score ~ genre + critics_score + best_pic_nom +
                    best_actor_win + best_actress_win + best_dir_win, data=movies)
summary(Step2_model_runtime)$adj.r.squared
```

```
## [1] 0.5254747
```

```
#remove variable genre based on Step1 model
Step2_model_genre <- lm(audience_score ~ runtime + critics_score + best_pic_nom +
                    best_actor_win + best_actress_win + best_dir_win, data=movies)
summary(Step2_model_genre)$adj.r.squared
```

```
## [1] 0.5020108
```

```
# compare the adjusted R-square after each varaible was removed, and the full model
Step1_MODEL <- c("audience ~ genre + runtime + critics + pic + actor + actress + dir",
"mpaa", "52.7989%")

Step2_M1<- c("audience ~ genre + runtime + critics + pic + actor + actress", "mpaa + di
r", "52.873%")

Step2_M2 <- c("audience ~ genre + runtime + critics + pic + actor + dir", "mpaa + actres
s", "52.76834%")

Step2_M3 <- c("audience ~ genre + runtime + critics + pic + actress + dir", "mpaa + acto
r", "52.8099%")

Step2_M4 <- c("audience ~ genre + runtime + critics + actor + actress + dir", "mpaa + pi
c", "52.21492%")

Step2_M5 <- c("audience ~ genre + runtime + pic + actor + actress + dir", "mpaa + critic
s", "23.35755%")

Step2_M6 <- c("audience ~ genre + critics + pic + actor + actress + dir", "mpaa + runtim
e", "52.54065%")

Step2_M7 <- c("audience ~ runtime + critics + pic + actor + actress + dir", "mpaa + genr
e", "50.20")
```

```
#adjusted R-square comparisons with one variable removed each time

STEP_2_Models <- rbind(Step1_MODEL, Step2_M1, Step2_M2, Step2_M3, Step2_M4, Step2_M5, St
ep2_M6, Step2_M7)

colnames(STEP_2_Models) <- c("Model", "Variable Removed", "Adjusted R-square")
STEP_2_Models
```

```
##             Model
## Step1_MODEL "audience ~ genre + runtime + critics + pic + actor + actress + dir"
## Step2_M1    "audience ~ genre + runtime + critics + pic + actor + actress"
## Step2_M2    "audience ~ genre + runtime + critics + pic + actor + dir"
## Step2_M3    "audience ~ genre + runtime + critics + pic + actress + dir"
## Step2_M4    "audience ~ genre + runtime + critics + actor + actress + dir"
## Step2_M5    "audience ~ genre + runtime + pic + actor + actress + dir"
## Step2_M6    "audience ~ genre + critics + pic + actor + actress + dir"
## Step2_M7    "audience ~ runtime + critics + pic + actor + actress + dir"
##             Variable Removed Adjusted R-square
## Step1_MODEL "mpaa"           "52.7989%"
## Step2_M1    "mpaa + dir"     "52.873%"
## Step2_M2    "mpaa + actress" "52.76834%"
## Step2_M3    "mpaa + actor"   "52.8099%"
## Step2_M4    "mpaa + pic"     "52.21492%"
## Step2_M5    "mpaa + critics" "23.35755%"
## Step2_M6    "mpaa + runtime" "52.54065%"
## Step2_M7    "mpaa + genre"   "50.20"
```

From the output above, the model with the highest Adjusted R-square is the model when variable best_dir_win was removed further, the Adjust R-square is 52.87%, which is slightly higher than the step 1 model, with Adjusted R-square 52.7%, and the full model, with Adjusted R-square 52.63%.

Step 3 model, pick the model with highest Adjust R-square with 5 explanatory variables

```
#remove variable best_actress_win based on Step2 model
Step3_model_actress <- lm(audience_score ~ genre + runtime + critics_score + best_pic_no
m +
                   best_actor_win, data=movies)
summary(Step3_model_actress)$adj.r.squared
```

```
## [1] 0.5284216
```

```
#remove variable best_actor_win based on Step2 model
Step3_model_actor <- lm(audience_score ~ genre + runtime + critics_score + best_pic_nom
+
                  best_actress_win, data=movies)
summary(Step3_model_actor)$adj.r.squared
```

```
## [1] 0.5287463
```

```
#remove variable best_pic_nom based on Step2 model
Step3_model_pic <- lm(audience_score ~ genre + runtime + critics_score +
                   best_actor_win + best_actress_win, data=movies)
summary(Step3_model_pic)$adj.r.squared
```

```
## [1] 0.5228868
```

```
#remove variable critics_score based on Step2 model
Step3_model_critics_score <- lm(audience_score ~ genre + runtime + best_pic_nom +
                   best_actor_win + best_actress_win, data=movies)
summary(Step3_model_critics_score)$adj.r.squared
```

```
## [1] 0.2310918
```

```
#remove variable runtim based on Step2 model
Step3_model_runtime <- lm(audience_score ~ genre + critics_score + best_pic_nom +
                   best_actor_win + best_actress_win, data=movies)
summary(Step3_model_runtime)$adj.r.squared
```

```
## [1] 0.5261487
```

```
#remove variable genre based on Step2 model
Step3_model_genre <- lm(audience_score ~ runtime + critics_score + best_pic_nom +
                   best_actor_win + best_actress_win, data=movies)
summary(Step3_model_genre)$adj.r.squared
```

```
## [1] 0.502434
```

```
#compare the adjusted R-square after each varaible was removed, and the Step 2 model
Step2_MODEL<- c("audience ~ genre + runtime + critics + pic + actor + actress", "mpaa +
 dir", "52.873%")

Step3_M1 <- c("audience ~ genre + runtime + critics + pic + actor", "mpaa + dir + actres
s", "52.84%")

Step3_M2 <- c("audience ~ genre + runtime + critics + pic + actress", "mpaa + dir + acto
r", "52.87463%")

Step3_M3 <- c("audience ~ genre + runtime + critics + actor + actress", "mpaa + dir + pi
c", "52.28868%")

Step3_M4 <- c("audience ~ genre + runtime + pic + actor + actress", "mpaa + dir + critic
s", "23.10918%")

Step3_M5 <- c("audience ~ genre + critics + pic + actor + actress", "mpaa + dir + runtim
e", "52.60799%")

Step3_M6 <- c("audience ~ runtime + critics + pic + actor + actress", "mpaa + dir + genr
e", "50.2434")
```

```
STEP_3_Models <- rbind(Step2_MODEL, Step3_M1, Step3_M2, Step3_M3, Step3_M4, Step3_M5, St
ep3_M6)

colnames(STEP_3_Models) <- c("Model", "Variable Removed", "Adjusted R-square")
STEP_3_Models
```

```
##             Model
## Step2_MODEL "audience ~ genre + runtime + critics + pic + actor + actress"
## Step3_M1    "audience ~ genre + runtime + critics + pic + actor"
## Step3_M2    "audience ~ genre + runtime + critics + pic + actress"
## Step3_M3    "audience ~ genre + runtime + critics + actor + actress"
## Step3_M4    "audience ~ genre + runtime + pic + actor + actress"
## Step3_M5    "audience ~ genre + critics + pic + actor + actress"
## Step3_M6    "audience ~ runtime + critics + pic + actor + actress"
##             Variable Removed       Adjusted R-square
## Step2_MODEL "mpaa + dir"           "52.873%"
## Step3_M1    "mpaa + dir + actress" "52.84%"
## Step3_M2    "mpaa + dir + actor"   "52.87463%"
## Step3_M3    "mpaa + dir + pic"     "52.28868%"
## Step3_M4    "mpaa + dir + critics" "23.10918%"
## Step3_M5    "mpaa + dir + runtime" "52.60799%"
## Step3_M6    "mpaa + dir + genre"   "50.2434"
```

From the output above, the model with the highest Adjusted R-square is the model when variable best_act_win was removed further, the Adjust R-square is 52.87463%, which is slightly higher than the step 2 model,with Adjusted R-square 52.873%, the step 1 model, with Adjusted R-square 52.79892%, and the full model, with Adjusted R-square 52.63%.

Even though the increase is very tiny, I prefer the step 3 model, since it has fewer variabels.

Step 4 model, pick the model with highest Adjust R-square with 4 explanatory variables

```
#remove variable best_actress_win based on Step3 model
Step4_model_actress <- lm(audience_score ~ genre + runtime + critics_score + best_pic_no
m, data=movies)
summary(Step4_model_actress)$adj.r.squared
```

```
## [1] 0.528337
```

```
#remove variable best_actor_win based on Step3 model
Step4_model_pic <- lm(audience_score ~ genre + runtime + critics_score +
                  best_actress_win, data=movies)
summary(Step4_model_pic)$adj.r.squared
```

```
## [1] 0.5232491
```

```
#remove variable critics_score based on Step3 model
Step4_model_critics_score <- lm(audience_score ~ genre + runtime +
                   best_pic_nom + best_actress_win, data=movies)
summary(Step4_model_critics_score)$adj.r.squared
```

```
## [1] 0.2313458
```

```
#remove variable runtim based on Step3 model
Step4_model_runtime <- lm(audience_score ~ genre + critics_score +
                    best_pic_nom + best_actress_win, data=movies)
summary(Step4_model_runtime)$adj.r.squared
```

```
## [1] 0.526628
```

```
#remove variable genre based on Step3 model
Step4_model_genre <- lm(audience_score ~ runtime + critics_score +
                    best_pic_nom + best_actress_win, data=movies)
summary(Step4_model_genre)$adj.r.squared
```

```
## [1] 0.5019667
```

```
#comparison the adjusted R-square after each varaible was removed, and the Step 3 model
Step3_MODEL<- c("audience ~ genre + runtime + critics + pic + actress", "mpaa + dir + ac
tor", "52.87463%")

Step4_M1 <- c("audience ~ genre + runtime + critics + pic", "mpaa + dir + actor + actres
s ","52.8337")

Step4_M2 <- c("audience ~ genre + runtime + critics + actress", "mpaa + dir + actor + pi
c", "52.32491%")

Step4_M3 <- c("audience ~ genre + runtime + pic + actress", "mpaa + dir + actor + critic
s", "23.13458%")

Step4_M4 <- c("audience ~ genre + critics + pic + actress", "mpaa + dir + actor + runtim
e", "52.65605%")

Step4_M5 <- c("audience ~ runtime + critics + pic + actress", "mpaa + dir + actor + genr
e", "50.19667")
```

```
#adjusted R-square comparisons with one variable removed each time
STEP_4_Models <- rbind(Step3_MODEL, Step4_M1, Step4_M2, Step4_M3, Step4_M4, Step4_M5)

colnames(STEP_4_Models) <- c("Model", "Variable Removed", "Adjusted R-square")
STEP_4_Models
```

```
##             Model
## Step3_MODEL "audience ~ genre + runtime + critics + pic + actress"
## Step4_M1    "audience ~ genre + runtime + critics + pic"
## Step4_M2    "audience ~ genre + runtime + critics + actress"
## Step4_M3    "audience ~ genre + runtime + pic + actress"
## Step4_M4    "audience ~ genre + critics + pic + actress"
## Step4_M5    "audience ~ runtime + critics + pic + actress"
##             Variable Removed               Adjusted R-square
## Step3_MODEL "mpaa + dir + actor"           "52.87463%"
## Step4_M1    "mpaa + dir + actor + actress " "52.8337"
## Step4_M2    "mpaa + dir + actor + pic"     "52.32491%"
## Step4_M3    "mpaa + dir + actor + critics" "23.13458%"
## Step4_M4    "mpaa + dir + actor + runtime" "52.65605%"
## Step4_M5    "mpaa + dir + actor + genre"   "50.19667"
```

From the output above, the Adjusted R-square doesnn't increase compared to the Step 3 model, thus, I will keep the Step 3 model, with 3 variables removed (mpaa + dir + pic), and the Adjusted R-square is 52.88%.

Thus the final model is audience ~ genre + runtime + critics + actor + actress. The details are as below.

Firstly, I will check the model as a whole with ANOVA.

The null hypothesis for is that:

H0: None of the four variables are statistically significant to predict the response variable audience_score.

H1: At least one of the four variables is statistically significant to predict the response variable audience_score.

```
ANOVA <- aov(audience_score ~ genre + runtime + critics_score +
                  best_pic_nom + best_actress_win, data=movies)
summary(ANOVA)
```

```
##                   Df Sum Sq Mean Sq F value  Pr(>F)
## genre             10  51633    5163  26.760 < 2e-16 ***
## runtime            1   6236    6236  32.317   2e-08 ***
## critics_score      1  83578   83578 433.155 < 2e-16 ***
## best_pic_nom       1   1458    1458   7.555 0.00616 **
## best_actress_win   1    300     300   1.552 0.21323
## Residuals        635 122524     193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the ANOVA output above, except the variable best_actress_win, all other 3 variables are statistically significant, with p value less than 0.05.

I would still include the varaible best_actress_win at the final model, even if it is not statistically significant. However, this indicator based on the adjusted R-square method, when I add this variable, the Adjusted R-square increased, this gives the model higher predicted power even though the indicator may bot be statistically significant.

Next, I will determine the final equation model. Based on the equation, I could predicte the audience_score when the values of other variables are known.

```
Final_Model <- lm(audience_score ~ genre + runtime + critics_score +
                  best_pic_nom + best_actress_win, data=movies)
summary(Final_Model)
```

```
##
## Call:
## lm(formula = audience_score ~ genre + runtime + critics_score +
##     best_pic_nom + best_actress_win, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.498  -9.310   0.551   9.289  41.352
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    29.17308    3.70312   7.878 1.45e-14 ***
## genreAnimation                  6.06235    4.98043   1.217 0.223968
## genreArt House & International  5.98558    4.10178   1.459 0.144987
## genreComedy                    -0.50511    2.29810  -0.220 0.826102
## genreDocumentary                9.85007    2.79853   3.520 0.000463 ***
## genreDrama                      1.81989    1.96828   0.925 0.355520
## genreHorror                    -8.35275    3.39092  -2.463 0.014032 *
## genreMusical & Performing Arts 10.26484    4.43744   2.313 0.021028 *
## genreMystery & Suspense        -4.10073    2.53260  -1.619 0.105905
## genreOther                      1.24587    3.92266   0.318 0.750886
## genreScience Fiction & Fantasy -6.49455    4.94510  -1.313 0.189546
## runtime                         0.06146    0.03129   1.964 0.049920 *
## critics_score                   0.44005    0.02194  20.059  < 2e-16 ***
## best_pic_nomyes                 9.36084    3.22615   2.902 0.003842 **
## best_actress_winyes            -2.27080    1.82251  -1.246 0.213235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.89 on 635 degrees of freedom
## Multiple R-squared:  0.5389, Adjusted R-squared:  0.5287
## F-statistic: 53.01 on 14 and 635 DF,  p-value: < 2.2e-16
```

The equation for predicting audience score from genre, runtime, best_pic_nom, and best_actress_win is as below:

Predicted audience_score = 29.17 + 6.06 GenreAnimation + 5.98558 genreArt House & International - 0.50511 genreComedy +9.85007 genreDocumentar + 1.81989 genreDrama - 8.35275 genreHorror + 10.26484 genreMusical & Performing Arts - 4.10073 genreMystery & Suspense + 1.24587 genreOther - 6.49455 genreScience Fiction & Fantasy + 0.06146 runtime + 0.44005 critics_score + 9.36084 best_pic_nomyes - 2.27080 best_actress_winyes

After the determination of the model, I would like to check the conditions of the Multiple linear regression. The conditions include:

1. linear relationship between (numerical) explanatory variable and response variable

2. nearly normal residuals with mean 0

3. constant variability of residuals

4. independent residuals

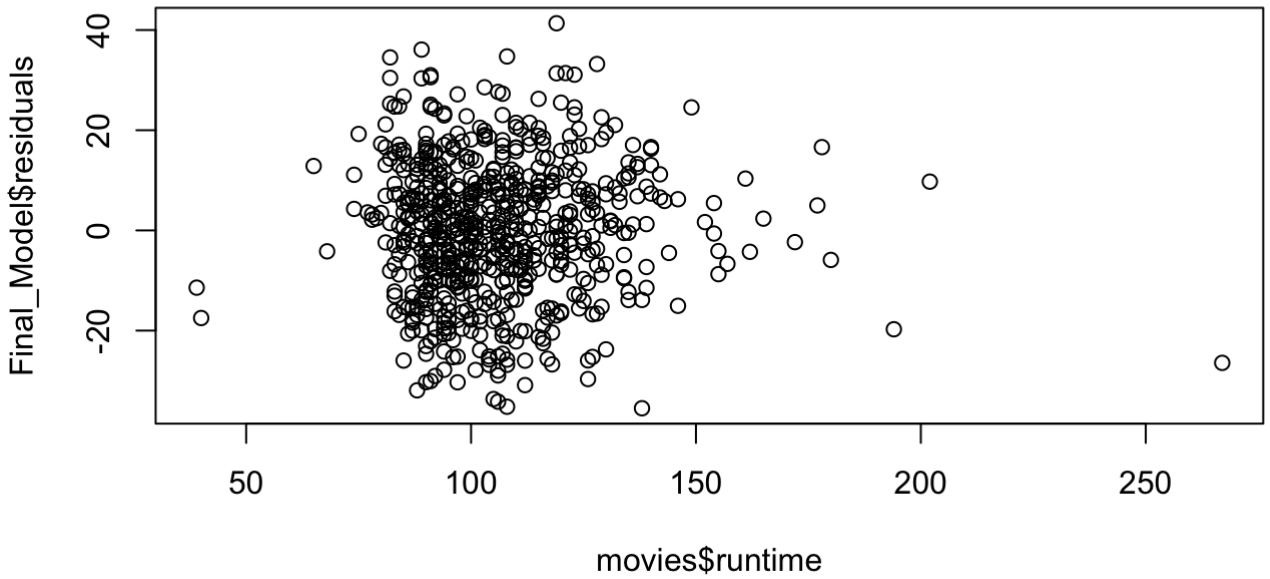
Next, I will check the conditions one by one.

Firstly, check the linear relationships between the two numerical explanatory varaibles (runtime and critics_score) and the response variable.

From Figure 4 and Figure 6 along with the correlation values R respectively, the response variable audience_score has a weak lienar relationship with the explanatory variable runtime (R=0.18), but has a strong linear relationship with the explanatory variable critics_score (R=0.7).

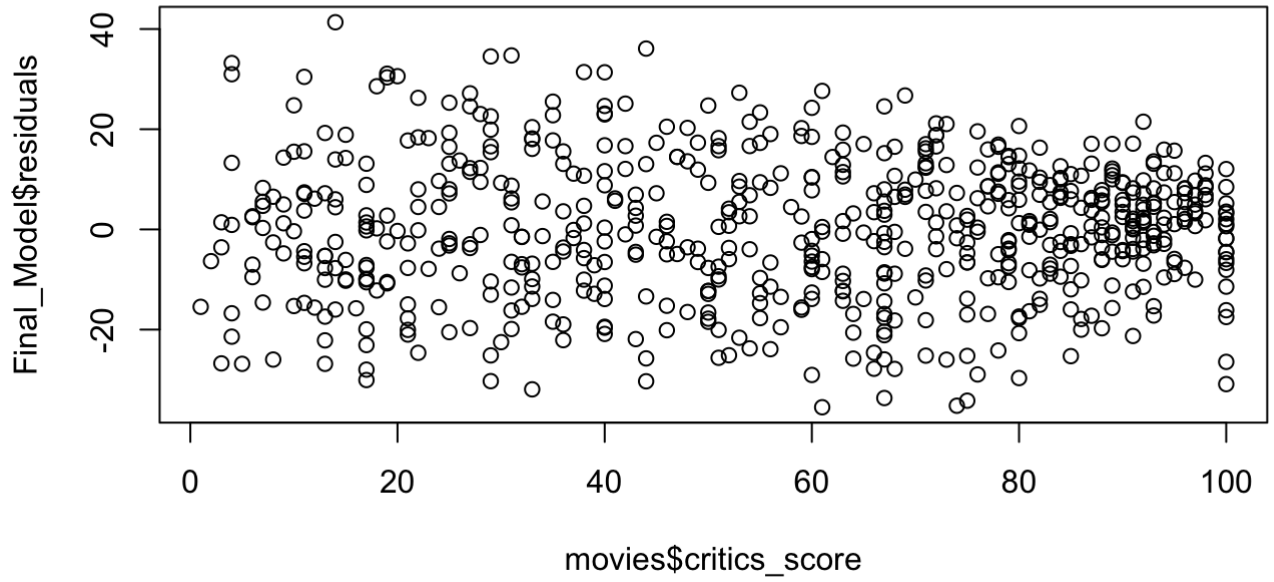
I will also use residuals plots to check the linearity relationships.

```
#1. check the 1st condition: linearlity using residuals plots with residuals VS runtime
Final_Model <- lm(audience_score ~ genre + runtime + critics_score +
                    best_pic_nom + best_actress_win, data=movies)
plot(Final_Model$residuals ~ movies$runtime)
```



From the plot of residuals vS runtime, the residuals are randomly scattered around 0.

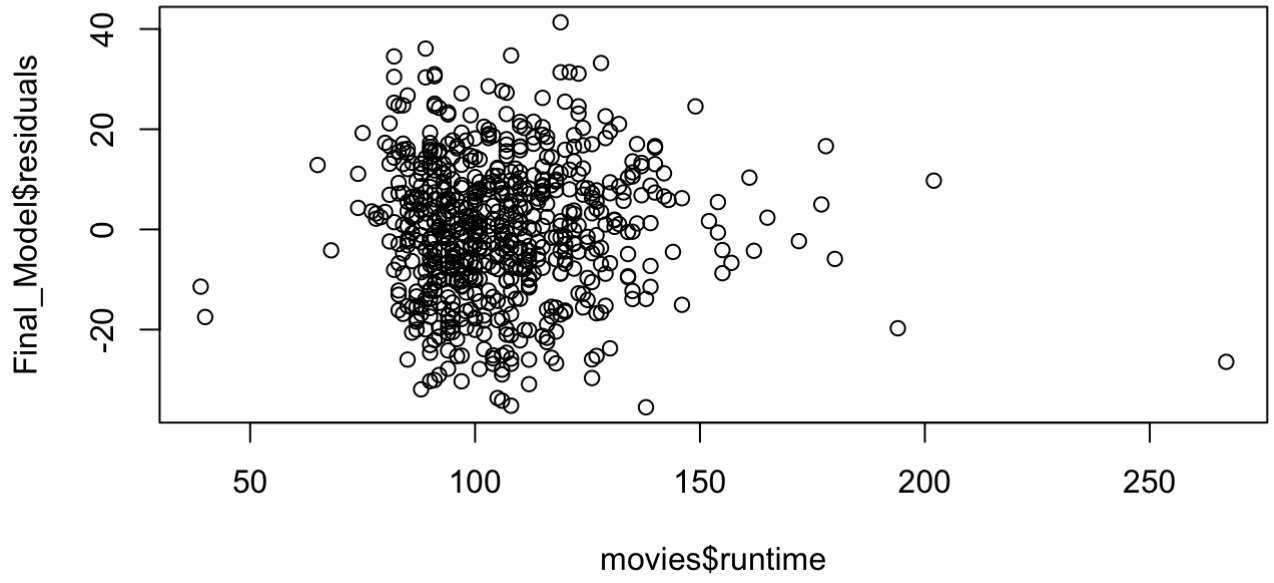
```
#1. check the 1st condition: linearlity using residuals plots with residuals VS critics_
score
Final_Model <- lm(audience_score ~ genre + runtime + critics_score +
                    best_pic_nom + best_actress_win, data=movies)
plot(Final_Model$residuals ~ movies$critics_score)
```



From the plot of residuals vS critics_score, the residuals are randomly scattered around 0.

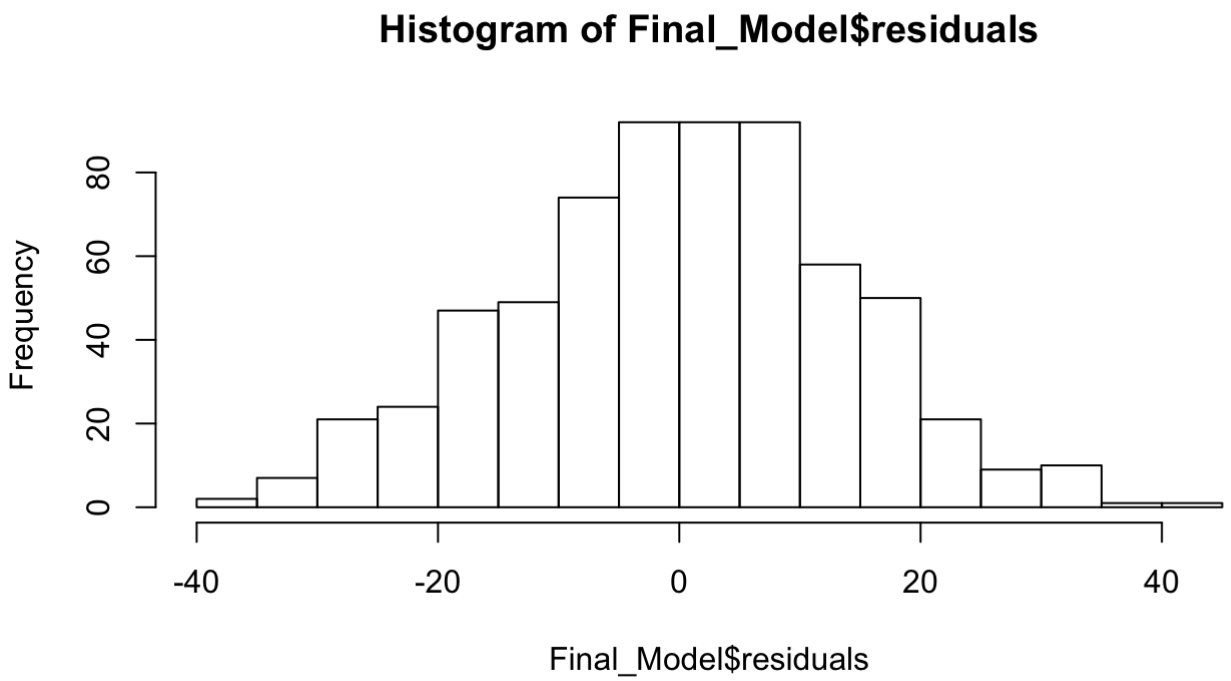
Thus, we meet the first condition here.

```
#1. check the 1st condition: linearlity using residuals plots with residuals VS runtime
Final_Model <- lm(audience_score ~ genre + runtime + critics_score +
                  best_pic_nom + best_actress_win, data=movies)
plot(Final_Model$residuals ~ movies$runtime)
```

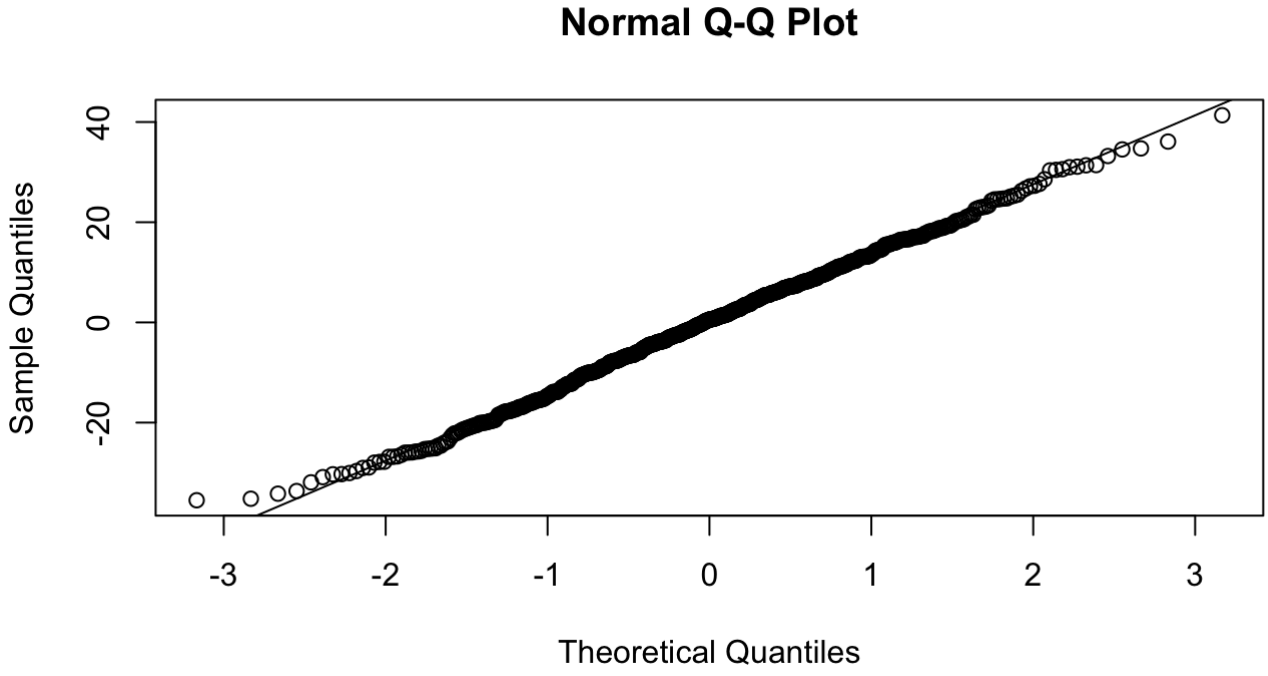


Secondly, check the normality of the residuals.

```
#2. check the 2nd condition: nearly normal residuals with mean 0
hist(Final_Model$residuals)
```




```
qqnorm(Final_Model$residuals)
qqline(Final_Model$residuals)
```



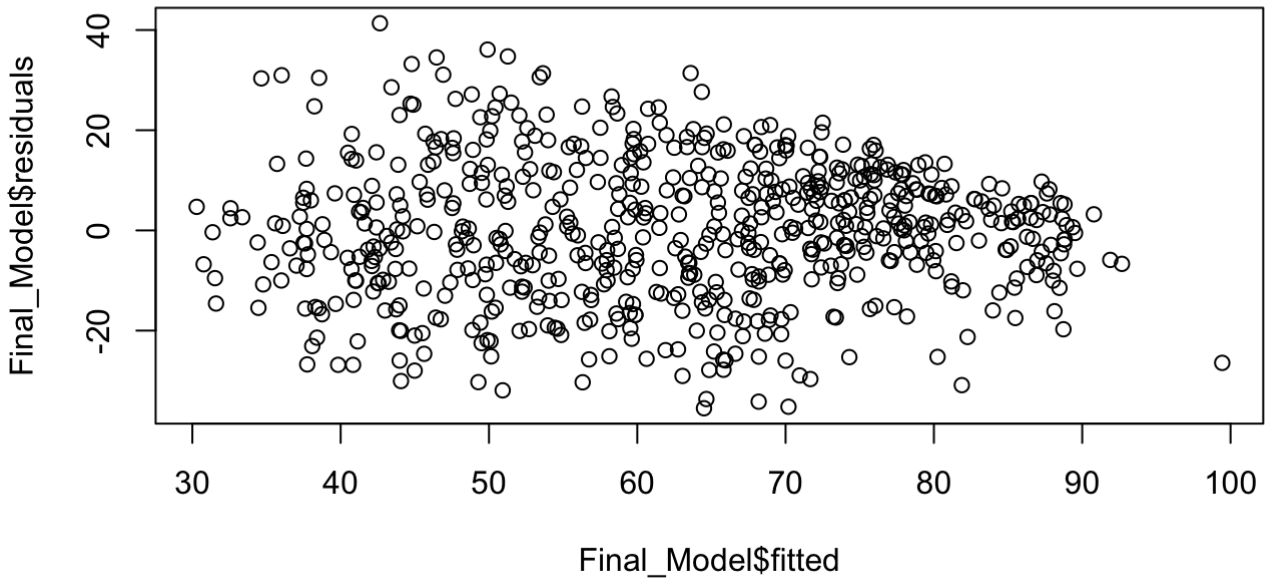
From the histogram, we can view a quite good normal distribution. From the Q-Q plot, except some points at the tails, we don't see any huge deviations from the mean. So we can say this condition seems to be fairly satisfied.

Thirdly, check the constant variability of the residuals

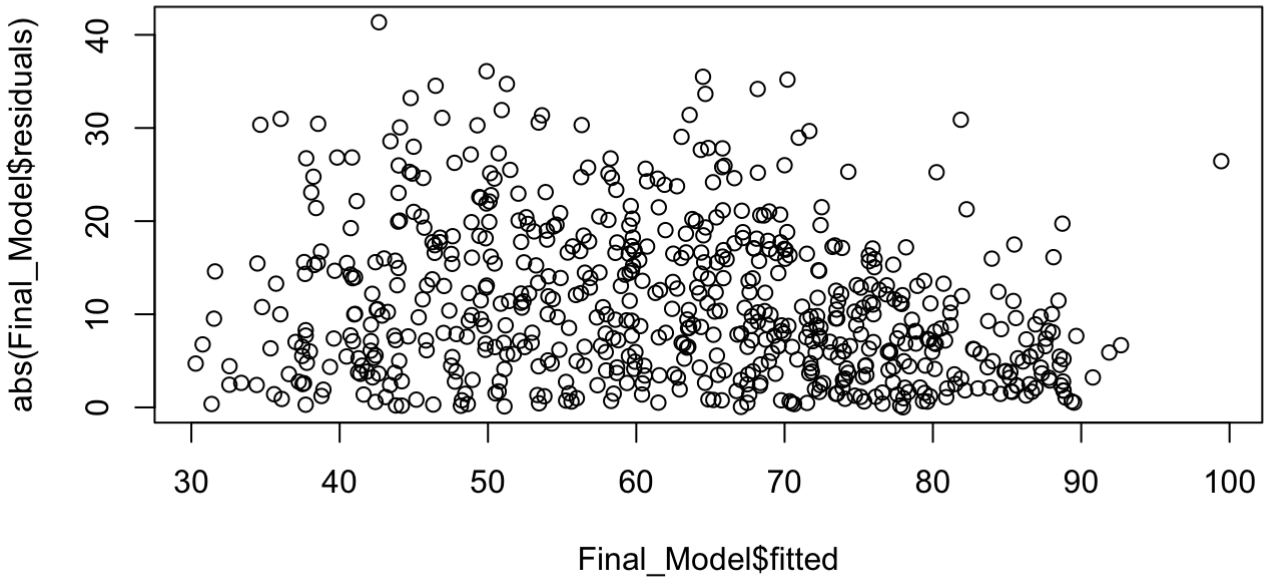
the residuals should be equally varaible for low and high values of the predicted response varaible.

I will check the residuals plots of residuals VS predicted values.

```
#3. check the 3rd condition: constant variability of residuals
plot(Final_Model$residuals ~ Final_Model$fitted)
```




```
plot(abs(Final_Model$residuals) ~ Final_Model$fitted)
```

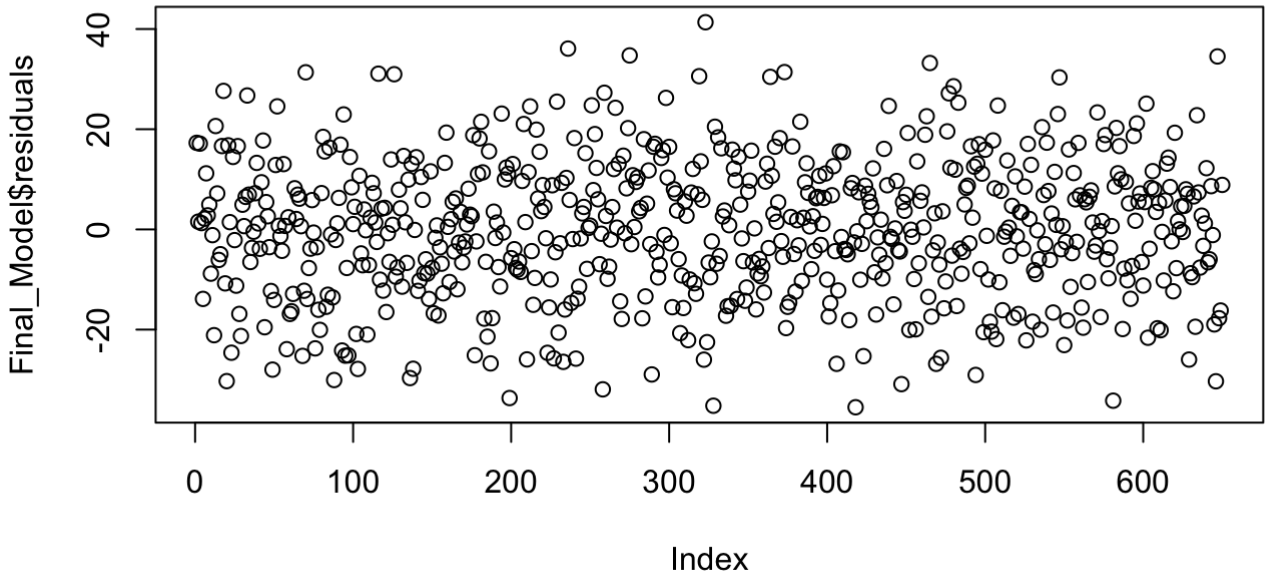


The first plot above is the residuals VS the fitted. We can see a little bit fan shape, but not bad. It appears that the variability of the residuals stay constant as the values of the predicted values change. So the constant variability condition appears to be met.

The second plot above is the absolute values of residuals VS the fitted. So if we would see a fan shape in the first plot, we would see a triangle in the absoute residuals VS fitted plot.

Fourthly, check the independence of the residuals

```
#4. check the 4th condition: independent residuals
plot(Final_Model$residuals)
```



From the plot above, we take a look to see if the order of data collection plot looks wonky in any way. If there was come non-independent structure, we would see these residuals increasing or decreasing, but we don't see any such pattern. So the condition of independent residuals is met.

To conclude, all the conditions for the multiple linear regression are met. And based on the backword method, the final model is parsimonious based on the variables discussed in this study.

Part 5: Prediction

Based on the final model equation:

Predicted audience_score = 29.17 + 6.06 GenreAnimation + 5.98558 genreArt House & International - 0.50511 genreComedy +9.85007 genreDocumentar + 1.81989 genreDrama - 8.35275 genreHorror + 10.26484 genreMusical & Performing Arts - 4.10073 genreMystery & Suspense + 1.24587 genreOther - 6.49455 genreScience Fiction & Fantasy + 0.06146 runtime + 0.44005 critics_score + 9.36084 best_pic_nomyes - 2.27080 best_actress_winyes

I am interested to know the audience_score for a movie with the following characteristics, genre = Comedy, runtime = 120 minutes, critics_score = 90, best_pic_nom = yes, and best_actress_win = yes

```
new_audience_score <- data.frame(genre = "Comedy", runtime = 120, critics_score = 90, be
st_pic_nom = "yes", best_actress_win = "yes")
predict(Final_Model,new_audience_score)
```

```
##        1
## 82.73827
```

Thus, the predicted audience_score for the above characterisitcs is 82.738.

I can also construct a prediction interval around this prediction, which will provide a measure of uncertainty around the prediction.

```
predict(Final_Model,new_audience_score, interval = "prediction", level = 0.05)
```

```
##        fit      lwr      upr
## 1 82.73827 81.83874 83.63779
```

Hence, the model predicts, with 95% confidence, that the movie with the above characteristics is expected to have an audience_score between 81.84 and 83.64.

Part 6: Conclusion

The parsimonious model for predicting audience score from genre, runtime, best_pic_nom, and best_actress_win is as below:

Predicted audience_score = 29.17 + 6.06 GenreAnimation + 5.98558 genreArt House & International - 0.50511 genreComedy +9.85007 genreDocumentar + 1.81989 genreDrama - 8.35275 genreHorror + 10.26484 genreMusical & Performing Arts - 4.10073 genreMystery & Suspense + 1.24587 genreOther - 6.49455 genreScience Fiction & Fantasy + 0.06146 runtime + 0.44005 critics_score + 9.36084 best_pic_nomyes - 2.27080 best_actress_winyes

It means that when keep all other variables constant, the movies when gene = Musical & Performing Arts are most popular among audience, while the movies when gene = Horror are least popular among audience.

Referring to runtime, as the runtime increases by one minute, while keep all other variables constant, the audience_score will increase by 0.06, on average, which are only valid within the range of the runtime.

Referring to scitics_score, as the scritics_score increases by one unit, while keep all other variables constant, the audience_score will increase by 0.44, on average.

Referring to Best_pic_nom, when keep all other variables constant, the audience_scores for the movies which were nominated as best_pic is 9.36 units higher on average than those movies which were not nominated as best_pic.

However, referring to the best_actress_win, when keep all other variables constant, the audience_scores for the movies whose main actress won the Oscar were 2.27 units lower on average than those movies whose main actress didn’t won the Oscar.