Setup

Load packages

```
library(ggplot2)
library(tidyverse)
library(statsr)
library(BAS)
library(MASS)
library(broom)
```

Load data

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

Part 1: Data

The data set is comprised of 651 randomly sampled movies produced and released before 2016.Due to random sampling, the research result would be able to generalize to all movies that are within the data range of this data set (or, when no extrapolation is made).Due to lack of random assignment, no causality conclustion could be formed.

Part 2: Data manipulation

```
# filter through NAs in movies and create new variable feature_film
nmovies<-na.omit(movies)%>%
  mutate(feature_film = ifelse(title_type == "Feature Film", "Yes", "No"))
```

```
# Create new variable drama
nmovies<-nmovies%>%
  mutate(drama = ifelse(genre == "Drama", "Yes", "No"))
```

```
# Create new variable mpaa_rating_R
nmovies<-nmovies%>%
  mutate(mpaa_rating_R = ifelse(mpaa_rating == "R", "Yes", "No"))
```

```
# Create two new variable
nmovies<-nmovies%>%
  mutate(oscar_season = ifelse(thtr_rel_month == 10|thtr_rel_month == 11|thtr_rel_mont
h == 12, "Yes", "No"))%>%
  mutate(summer_season = ifelse(thtr_rel_month == 5|thtr_rel_month == 6|thtr_rel_mont
h == 7|thtr_rel_month == 8, "Yes", "No"))
```

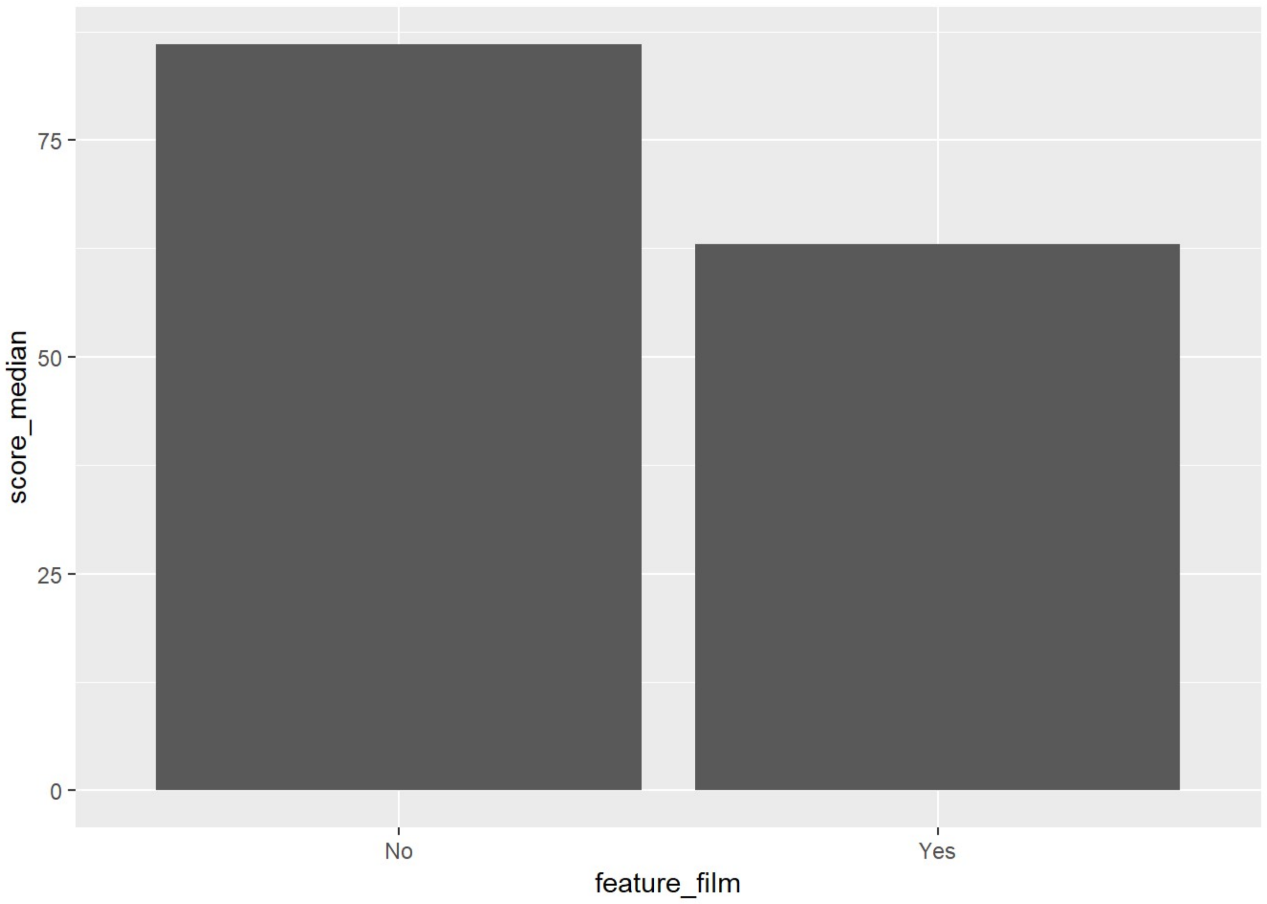
or can do the following:

movies$oscar_season = ifelse(movies$ thtr_rel_month %in% c(10,11,12), “yes”, “no”) movies $summer_season = ifelse(movies$ thtr_rel_month %in% c(5,6,7), “yes”,“no”)

```
# Create new variable season_special
nmovies<-nmovies%>%
  mutate(season_special = ifelse(oscar_season == "Yes", "season_oscar", ifelse(summer_
season == "Yes", "season_summer", "season_none")))
```

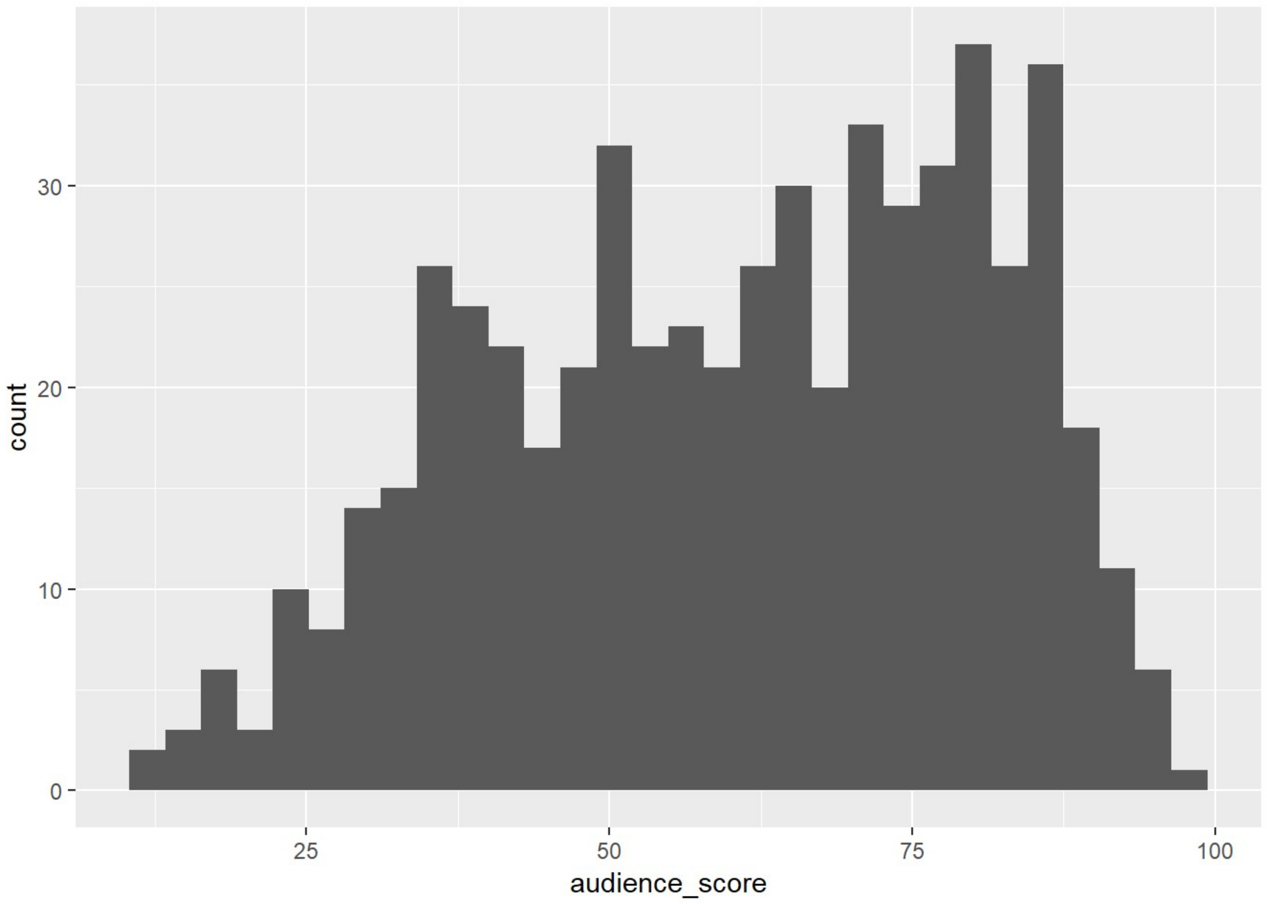
Part 3: Exploratory data analysis

```
nmovies%>%
  group_by(feature_film)%>%
  summarise(score_median = median(audience_score))%>%
  ggplot(aes(x = feature_film, y = score_median))+
  geom_bar(stat = "identity")
```



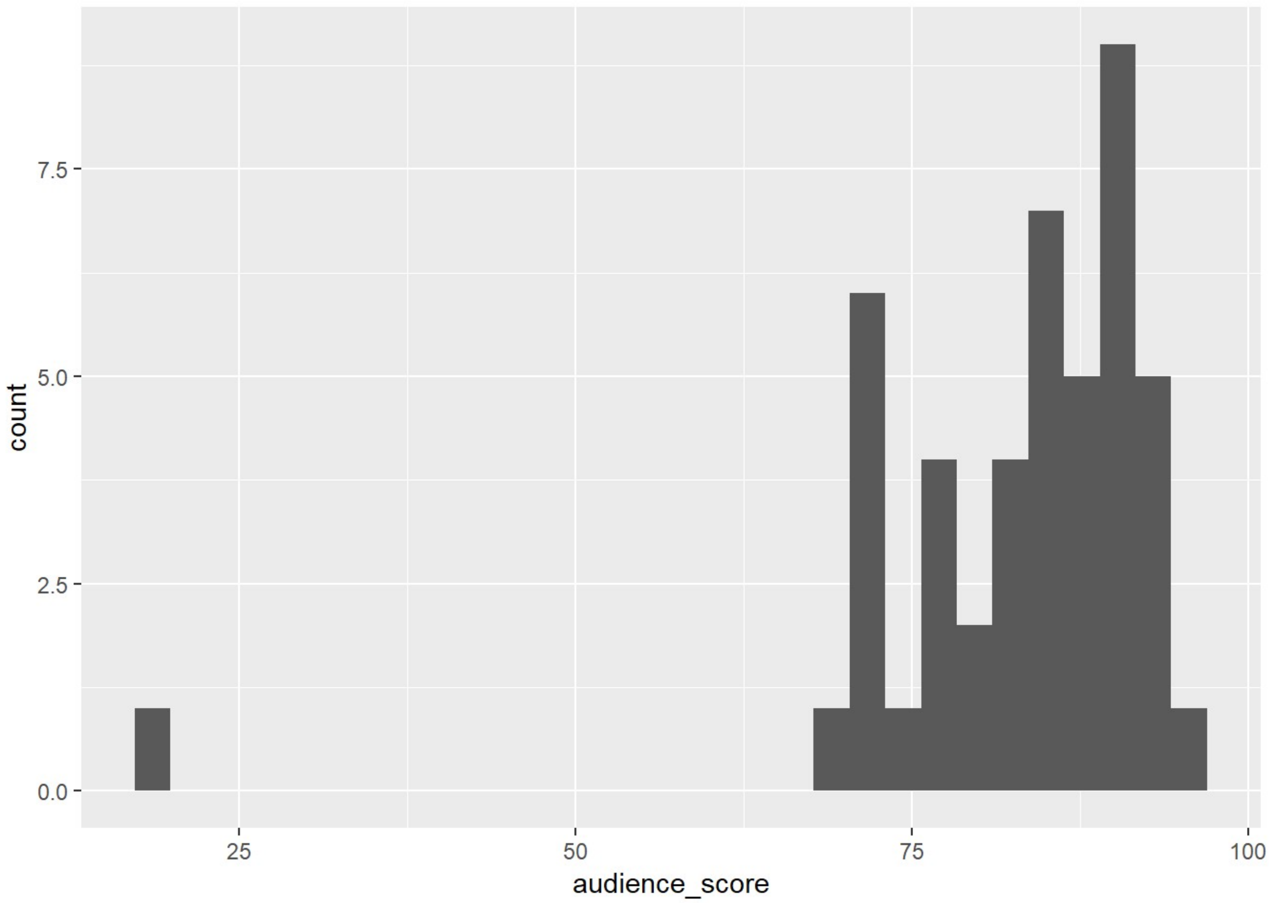
```
nmovies%>%
  filter(feature_film == "Yes")%>%
  ggplot(aes(x = audience_score))+
  geom_histogram()
```

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



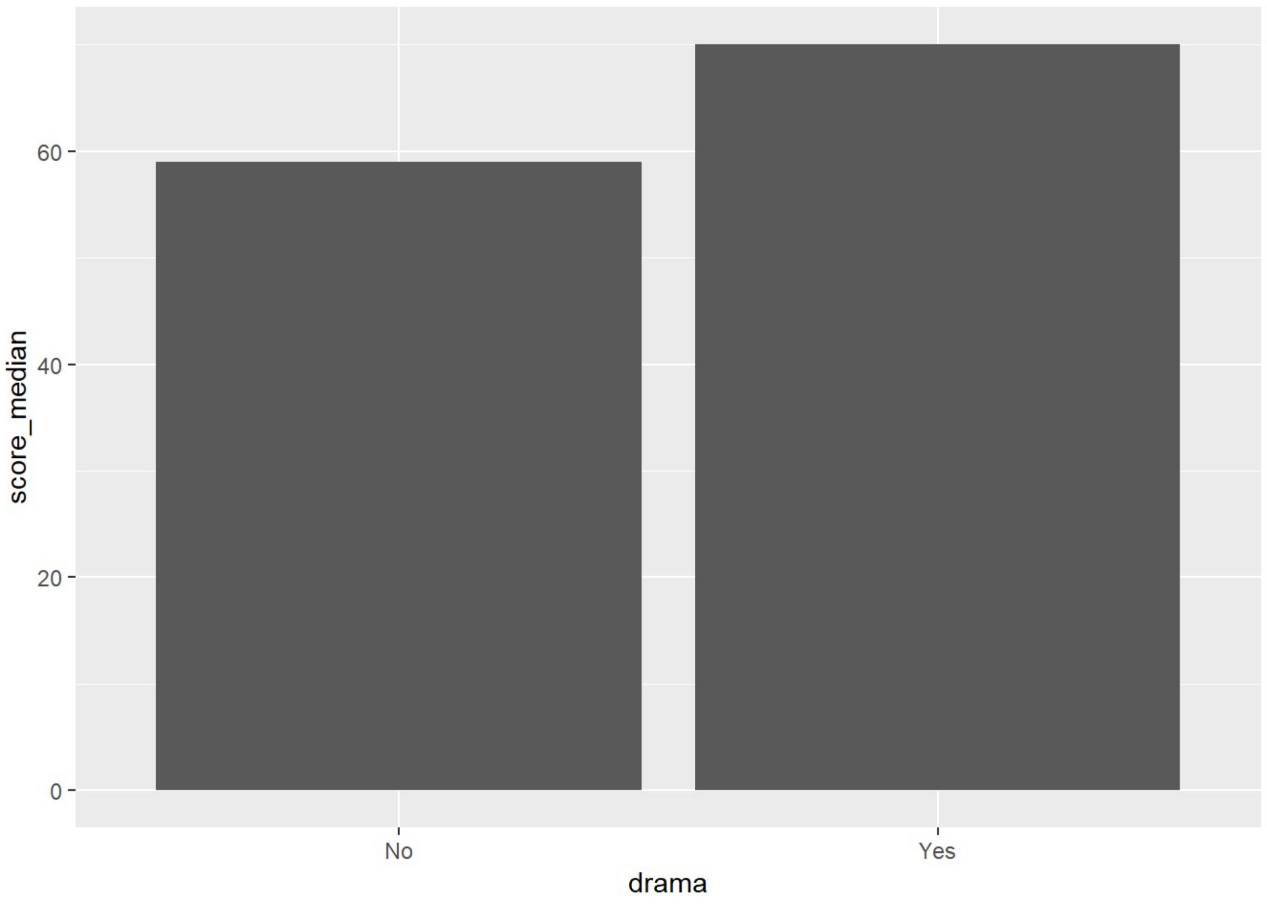
```
nmovies%>%
  filter(feature_film == "No")%>%
  ggplot(aes(x = audience_score))+
  geom_histogram()
```

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



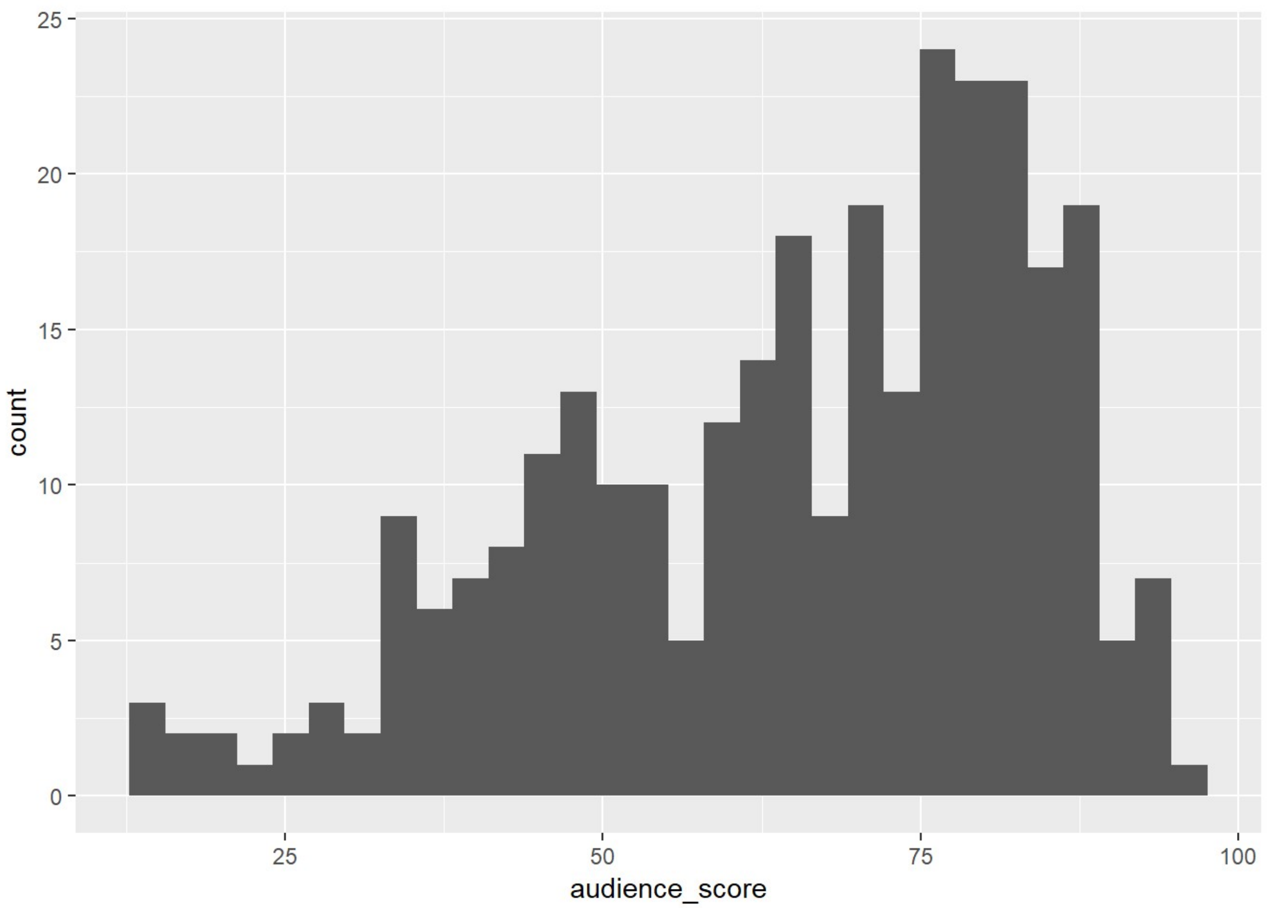
Per the above three plots, we can see that non-feature film has a higher median, and regardless of the one outlier case, both feature and non-feature films have a slightly left skewed distribution.

```
nmovies%>%
  group_by(drama)%>%
  summarise(score_median = median(audience_score))%>%
  ggplot(aes(x = drama, y = score_median))+
  geom_bar(stat = "identity")
```



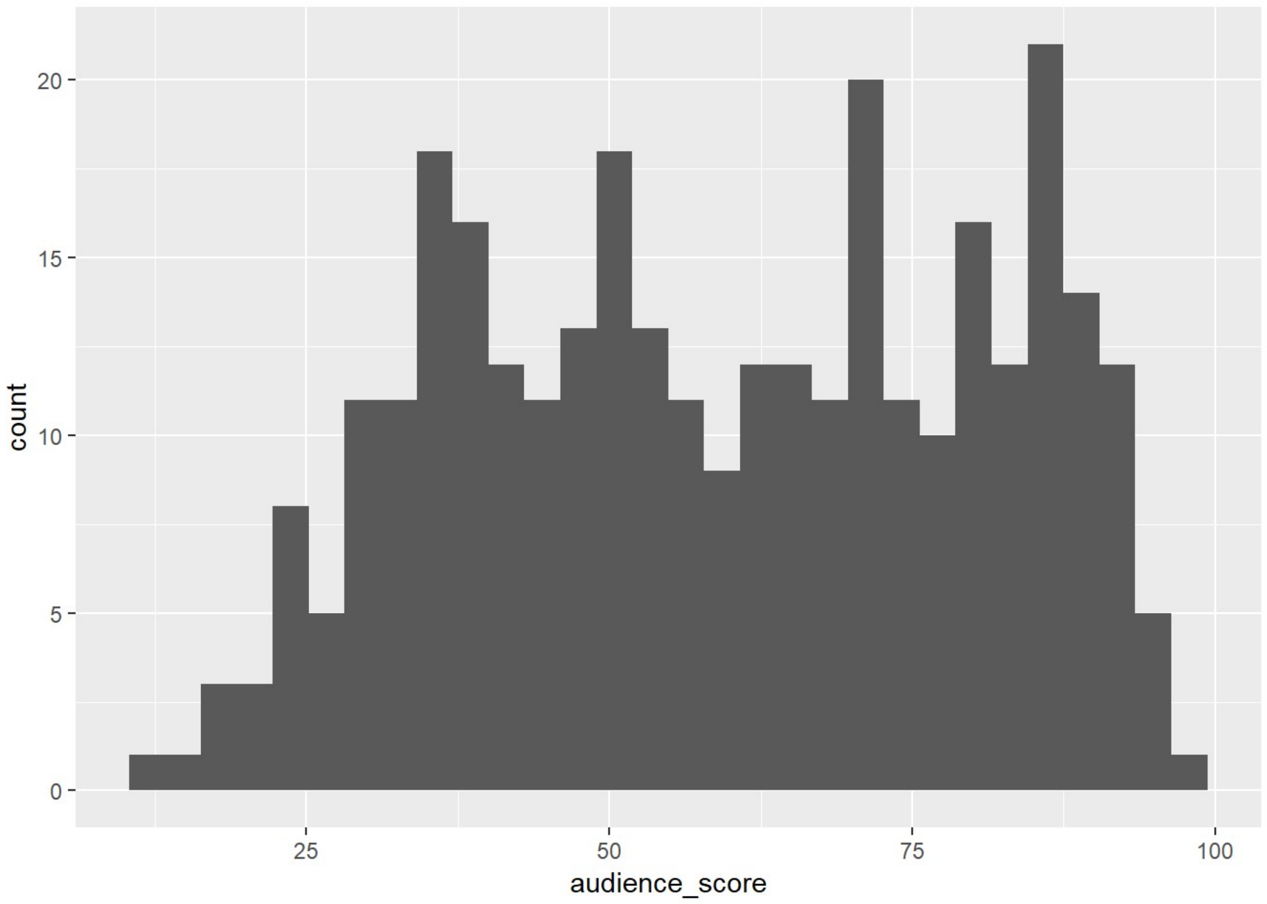
```
nmovies%>%
  filter(drama == "Yes")%>%
  ggplot(aes(x = audience_score))+
  geom_histogram()
```

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



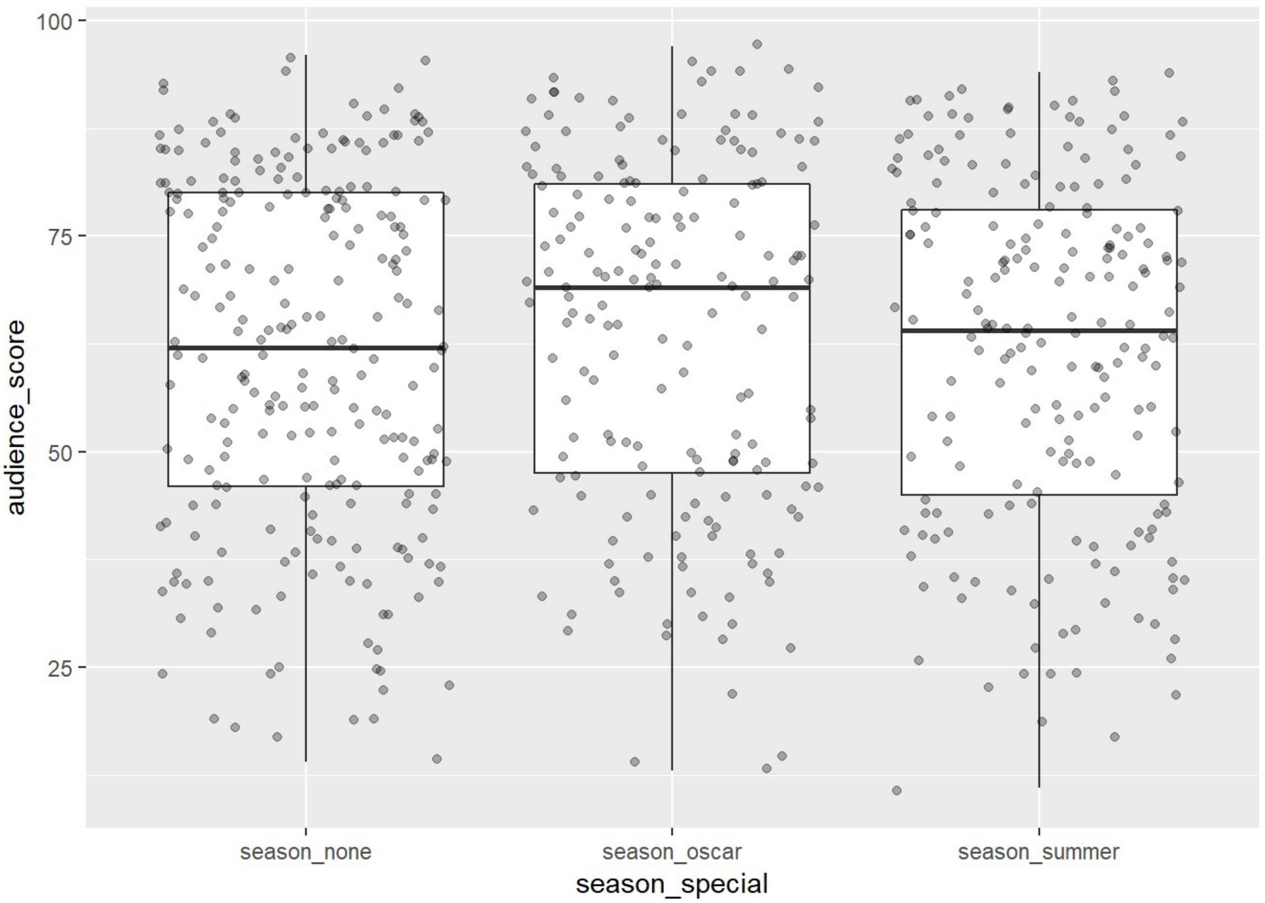
```
nmovies%>%
  filter(drama == "No")%>%
  ggplot(aes(x = audience_score))+
  geom_histogram()
```

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



Per the above three plots, we can see that drama film has a higher median, and both feature and non-feature films have a slightly left skewed distribution.

```
nmovies%>%
  ggplot(aes(x = season_special, y = audience_score))+
  geom_boxplot()+
  geom_jitter(alpha = 0.3)
```

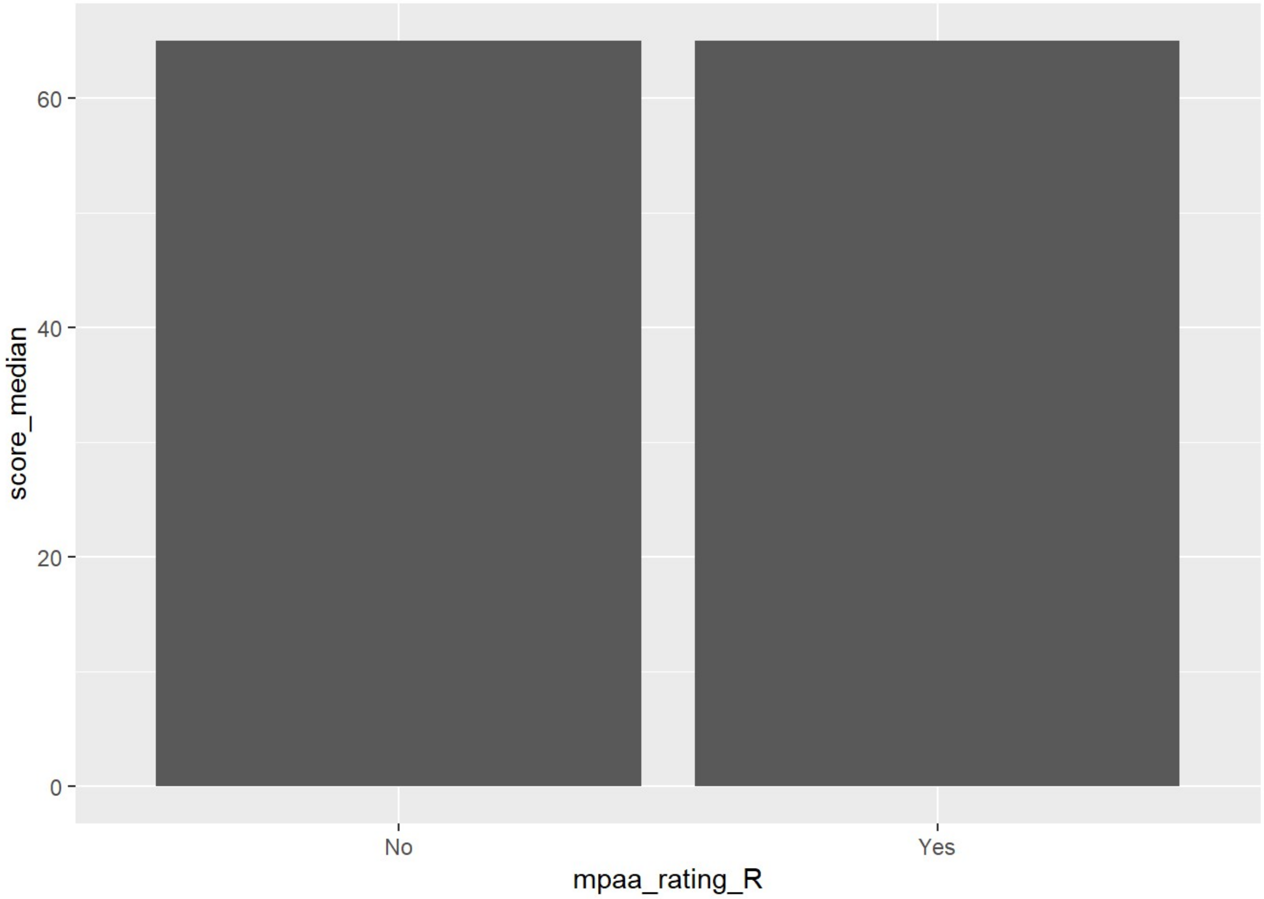


```
nmovies%>%
  group_by(season_special)%>%
  summarise(score_median=median(audience_score), std=sd(audience_score), score_iqr=IQR
(audience_score))
```

```
## # A tibble: 3 x 4
##   season_special score_median   std score_iqr
##   <chr>                 <dbl> <dbl>     <dbl>
## 1 season_none              62  20.3      34  
## 2 season_oscar             69  20.3      33.5
## 3 season_summer            64  19.9      33  
```

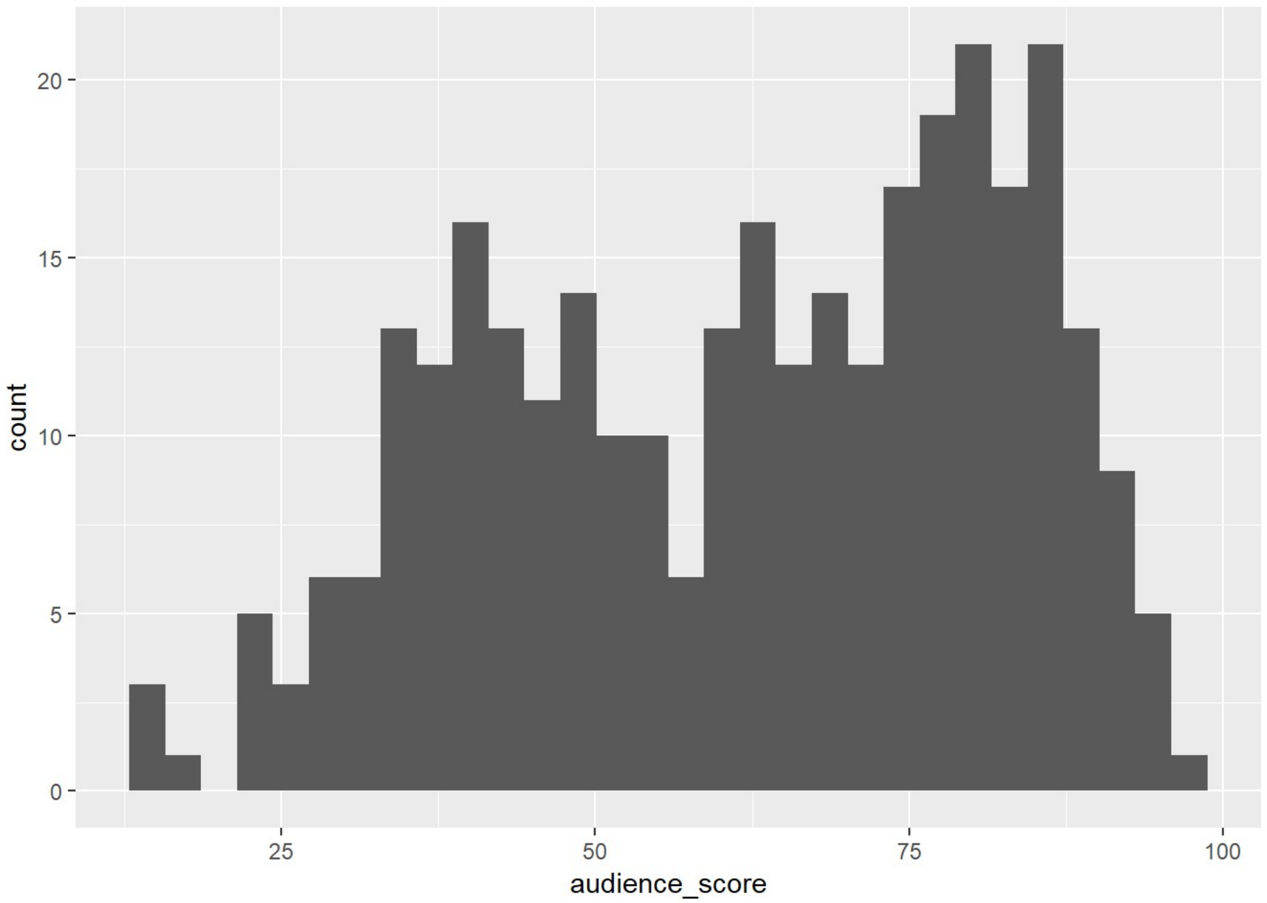
We can tell from the box plot and summary table, Summer seasoned films and non-special seasoned films have very similar IQR and median, only Oscar seasoned films stand out from the pool. Therefore, we will only include the variable oscar_season in our model, and leaving out season_special and summer_season.

```
nmovies%>%
  group_by(mpaa_rating_R)%>%
  summarise(score_median = median(audience_score))%>%
  ggplot(aes(x = mpaa_rating_R, y = score_median))+
  geom_bar(stat = "identity")
```



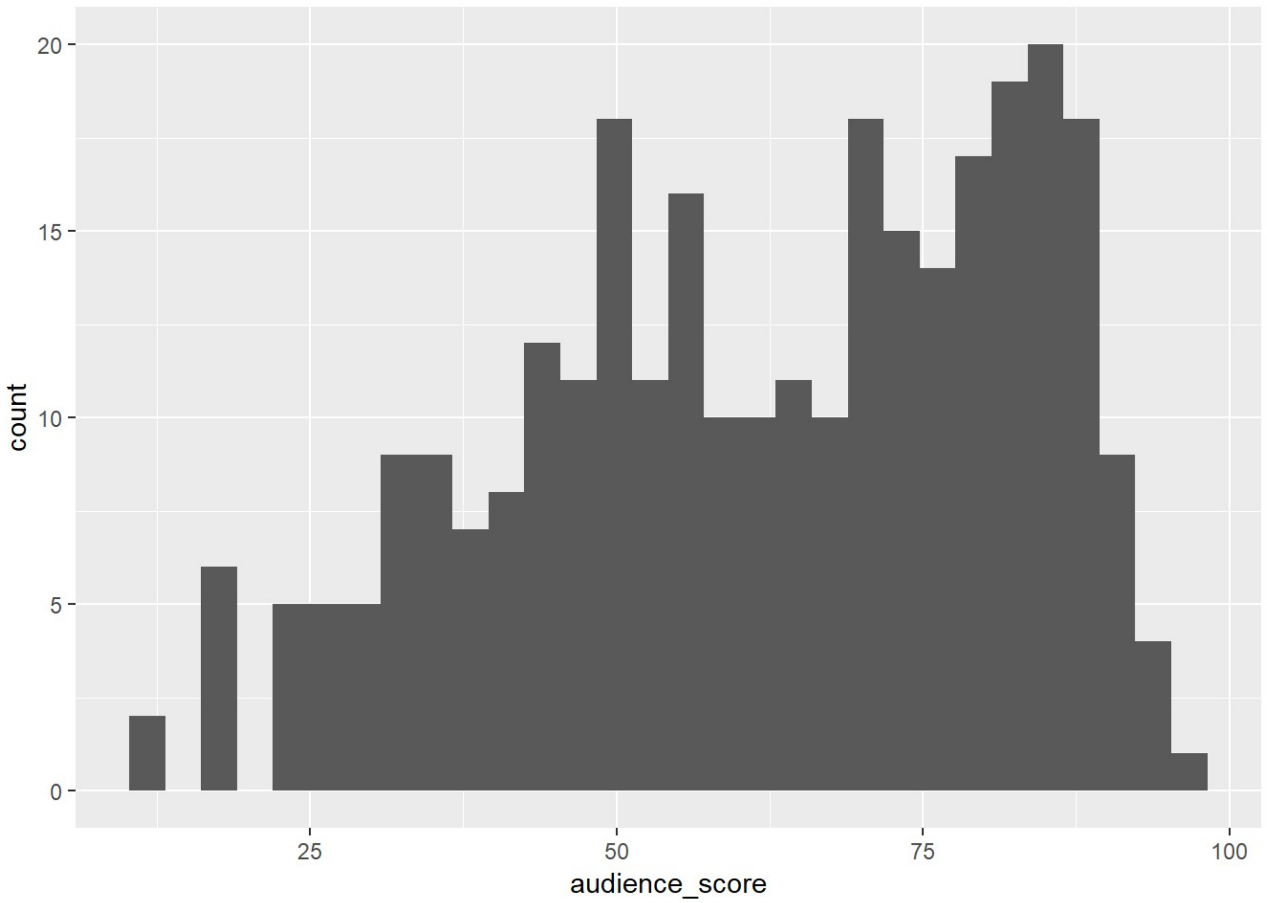
```
nmovies%>%
  filter(mpaa_rating_R == "Yes")%>%
  ggplot(aes(x = audience_score))+
  geom_histogram()
```

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



```
nmovies%>%
  filter(mpaa_rating_R == "No")%>%
  ggplot(aes(x = audience_score))+
  geom_histogram()
```

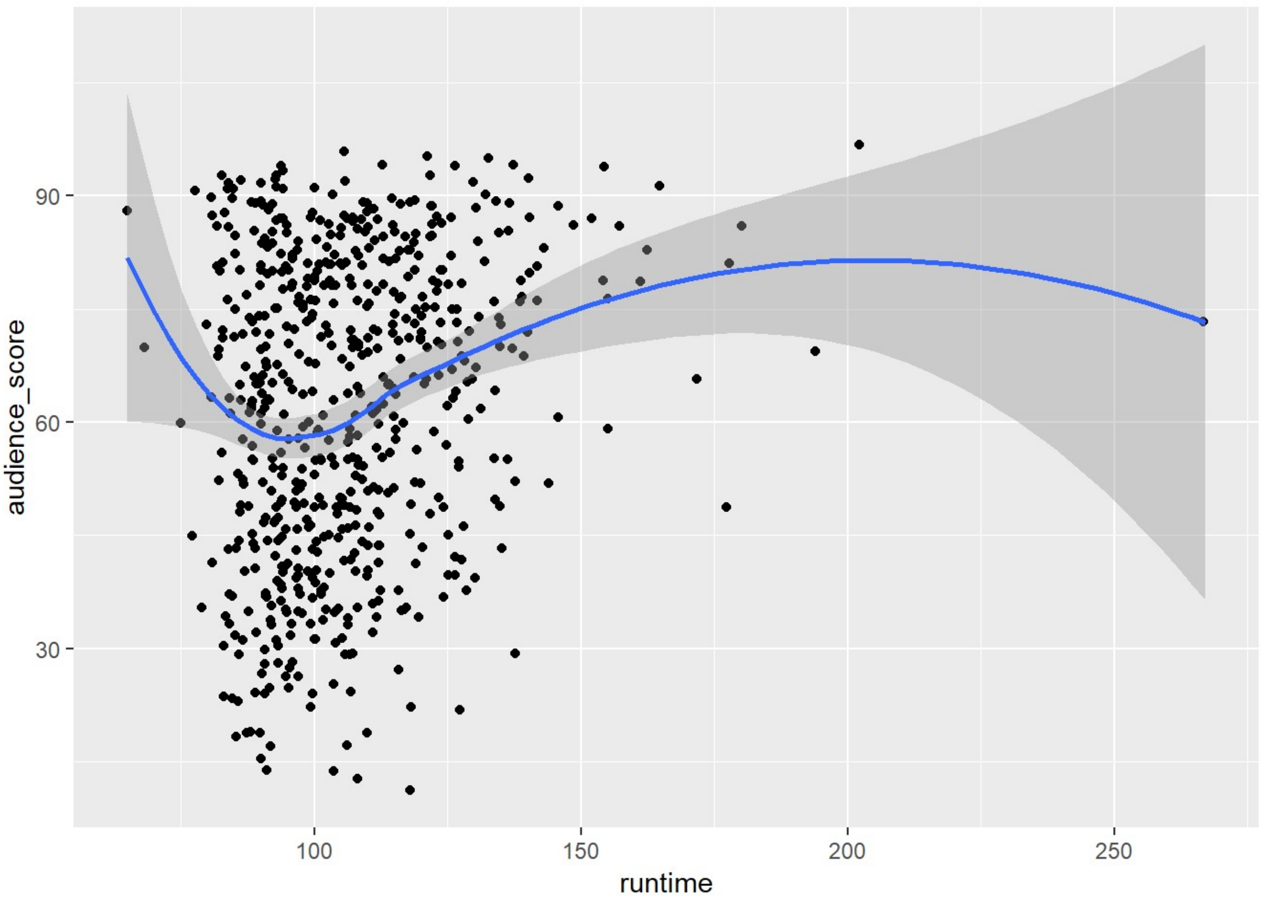
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Per the above three plots, we can see that r rated and non-r rated films seem to have very similar distribution and median.We will keep this variable in the following model, though, as the binominal variable is easy to process and would not save us much resource to exclude.

```
ggplot(data = nmovies, aes(x=runtime, y=audience_score))+
  geom_jitter()+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Clearly, runtime is not linear to audience scores. Therefore, it will be excluded from our model.

Part 4: Modeling

Develop a Bayesian regression model to predict `audience_score` from the following explanatory variables. Note that some of these variables are in the original dataset provided, and others are new variables you constructed earlier:

- `feature_film`
- `drama`
- `runtime`
- `mpaa_rating_R`
- `thtr_rel_year`
- `oscar_season`
- `summer_season`
- `imdb_rating`
- `imdb_num_votes`
- `critics_score`
- `best_pic_nom`
- `best_pic_win`
- `best_actor_win`

- `best_actress_win`
- `best_dir_win`
- `top200_box`

Complete Bayesian model selection and report the final model.

- Carrying out the model selection correctly (5 pts)
- Model diagnostics (5 pts)
- Interpretation of model coefficients (5 pts)

excluding certain variables:

1. overlapping variables such as title type, genre, thtr_rel_month, mpaa_rating, oscar_season, summer_season.

2. none linear relationship varibles such as runtime.

3. meaningless variables such as, thtr_rel_year, dvd_rel_year, dvd_rel_month, dvd_rel_day, imdb_num_votes, imdb_rating, critics_rating, critics_score. These variables become availble at the same time of or are reactions to the audience_score, or are not going to help predict future events, so that it is meaningless to include these variables in our prediction model.

4. variables that creates too many layers, such as director. However, studio is included in the model as it is intuitionally important to ratings and is not reflected in other variables. Director is somehow reflected by the variable best_dir_win.

Model construction:

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

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

```r
nnmovies<- nmovies[, c(1, 9, 18:24, 33:35, 36)]
```

```r
# prob of a case being an outlier:
# being below or above 3 standard deviations from 0
(prob_outlier <- pnorm(-3) + pnorm(3, lower.tail = FALSE))
```

```
## [1] 0.002699796
```

```
# probability of a signle case not being an outler is therefore the complement
(prob_not_outlier <- 1 - prob_outlier)
```

```
## [1] 0.9973002
```

```
# probability of no outliers in the sample of n assuming errors are independent a prio
ri
n <- nrow(nnmovies)
(prob_no_outliers <- prob_not_outlier^n)
```

```
## [1] 0.1876018
```

```
# probability of at least one outlier in the sample is the complement of the
# probability of no outliers in the sample of n
1 - prob_no_outliers
```

```
## [1] 0.8123982
```

```
# solve a new k to keep probability of outliers below 5%
(prob_obs_not_outlier <- 0.95^(1/n))
```

```
## [1] 0.9999171
```

```
(newk <- qnorm(0.5 + 0.5 * prob_obs_not_outlier))
```

```
## [1] 3.93597
```

```
# model with all variables
m_score_full<-lm(audience_score~. -title, data = nnmovies)
```

```
# use stepAIC to solve a model that minimize BIC
m_score_minbic<-stepAIC(m_score_full, scale = 0,
                        direction = "backward", trace = FALSE, keep = NULL, k = newk)
BIC(m_score_minbic)
```

```
## [1] 5389.125
```

```
summary(m_score_minbic)
```

```
##
## Call:
## lm(formula = audience_score ~ best_pic_nom + best_dir_win + top200_box +
##     feature_film + drama, data = nnmovies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -70.988 -13.756   1.091  13.761  40.244
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       81.835      2.690  30.421  < 2e-16 ***
## best_pic_nomyes   21.340      4.040   5.283 1.77e-07 ***
## best_dir_winyes    6.442      2.924   2.203  0.02798 *
## top200_boxyes     12.812      4.790   2.675  0.00768 **
## feature_filmYes  -27.078      2.875  -9.417  < 2e-16 ***
## dramaYes           8.153      1.514   5.385 1.03e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.22 on 613 degrees of freedom
## Multiple R-squared:  0.191,  Adjusted R-squared:  0.1844
## F-statistic: 28.94 on 5 and 613 DF,  p-value: < 2.2e-16
```

```
tidy(m_score_minbic)
```

```
## # A tibble: 6 x 5
##   term            estimate std.error statistic   p.value
##   <chr>              <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)        81.8       2.69     30.4  1.36e-124
## 2 best_pic_nomyes    21.3       4.04      5.28 1.77e-  7
## 3 best_dir_winyes     6.44      2.92      2.20 2.80e-  2
## 4 top200_boxyes      12.8       4.79      2.67 7.68e-  3
## 5 feature_filmYes   -27.1       2.88     -9.42 9.29e- 20
## 6 dramaYes            8.15      1.51      5.39 1.03e-  7
```

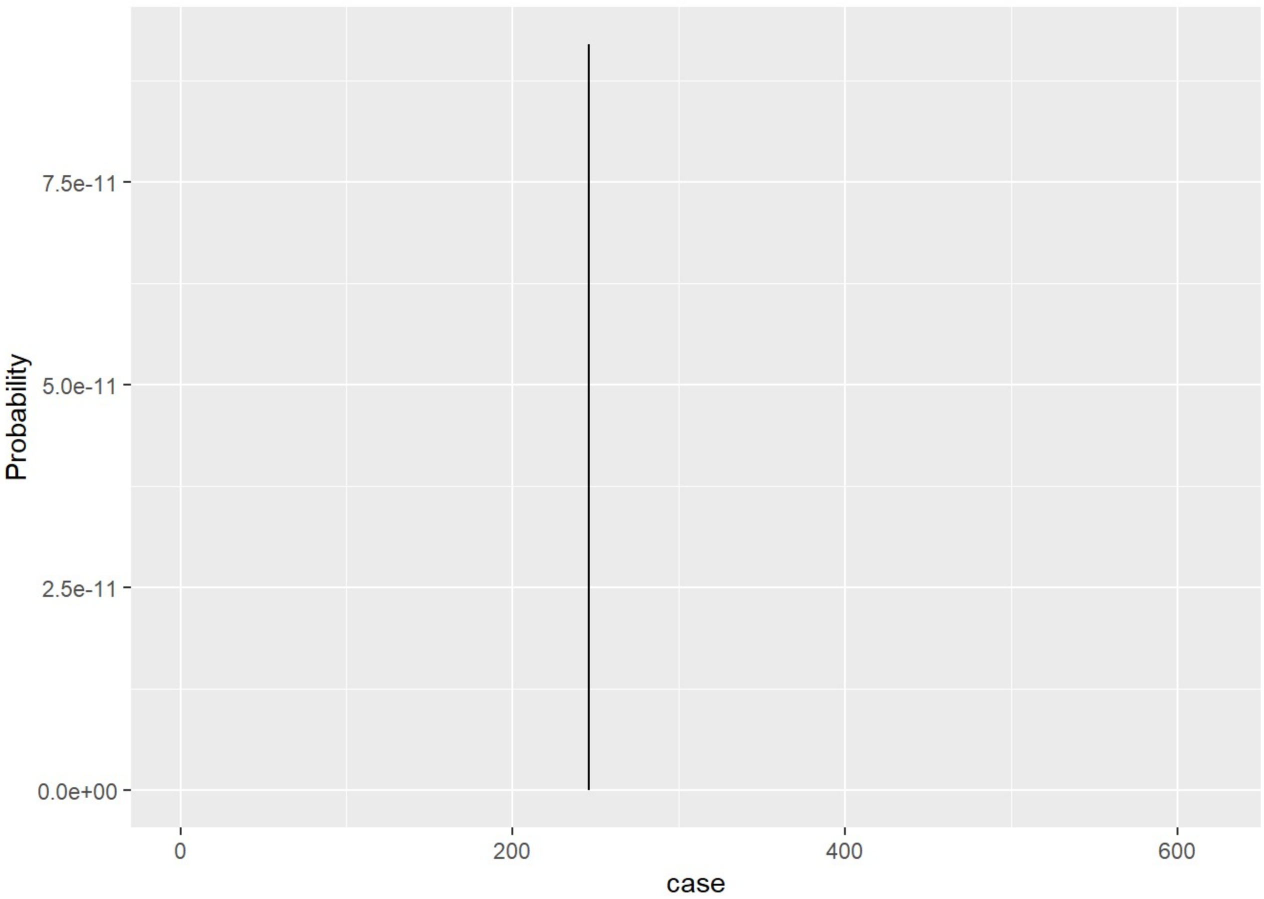
```
confint(m_score_minbic)
```

```
##                       2.5 %    97.5 %
## (Intercept)      76.551713  87.11731
## best_pic_nomyes  13.406919  29.27369
## best_dir_winyes   0.698751  12.18441
## top200_boxyes     3.404488  22.21867
## feature_filmYes -32.725274 -21.43123
## dramaYes          5.179909  11.12632
```

We can see that the min BIC model has 5 variables, all of them are significant as Pr|t| is smaller than 0.05. All other variables have positive relationship with audience scores as their coefficients are positive, except feature film that has a -27 coefficient as point coefficent, or (-33, -21) as 95% credible interval. It is 95% likely that a feature film will have -33 to -21 scores lower on average than non-feature films.

Model diagnostics:

```
# use newk to calculate the posterior probability of each observation being an outlier
outliers<-Bayes.outlier(m_score_minbic, prior.prob = 0.95)

outliers_df <- data.frame(probability = outliers$prob.outlier,
                          case = 1:length(outliers$prob.outlier))

ggplot(outliers_df, aes(ymax = probability, x = case)) +
  geom_linerange(ymin = 0) +
  labs(y = "Probability")
```



```
outliers_df %>%
  filter(probability > 0.95)
```

```
## [1] probability case
## <0 rows> (or 0-length row.names)
```

After refitting new k, one case has significantly higher probability than other cases to be an outlier, but the chance is still almost 0. Therefore, there is no need to exclude any outliers from our model fitting data.

```
# Fit the model using Bayesian linear regression, `bas.lm` function in the `BAS` packa
ge
bma_score <- bas.lm(audience_score~. -title, data = nnmovies,
                   prior = "BIC",
                   modelprior = uniform())

# Print out the marginal posterior inclusion probabilities for each variabl
e
bma_score
```

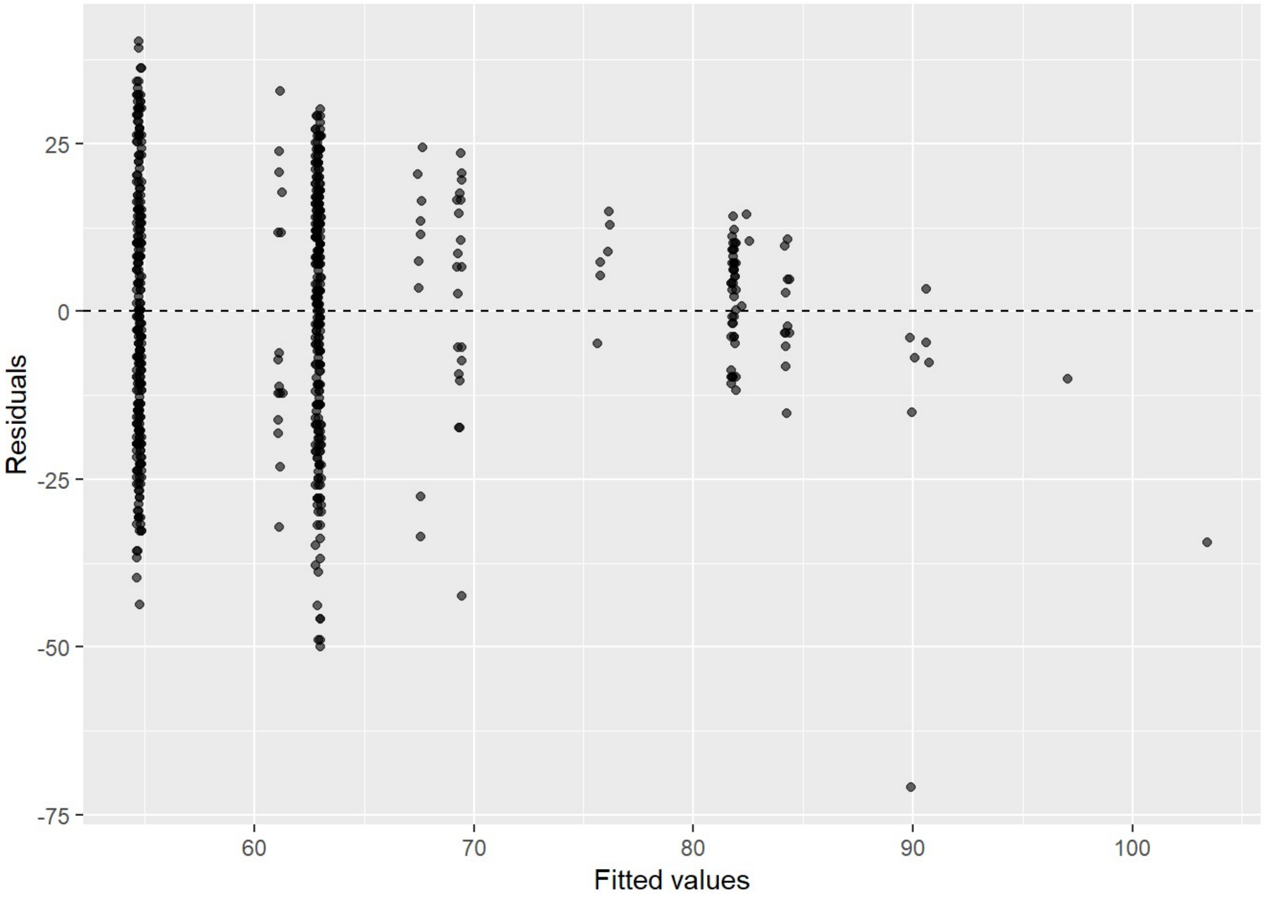
```
##
## Call:
## bas.lm(formula = audience_score ~ . - title, data = nnmovies,
##     prior = "BIC", modelprior = uniform())
##
##
##  Marginal Posterior Inclusion Probabilities:
##           Intercept         thtr_rel_day      best_pic_nomyes
##             1.00000              0.04146              0.99998
##     best_pic_winyes    best_actor_winyes  best_actress_winyes
##             0.04171              0.03952              0.05198
##     best_dir_winyes        top200_boxyes      feature_filmYes
##             0.32484              0.62008              1.00000
##            dramaYes     mpaa_rating_RYes      oscar_seasonYes
##             0.99998              0.07778              0.04041
```

```
# Top 5 most probably models
summary(bma_score)
```

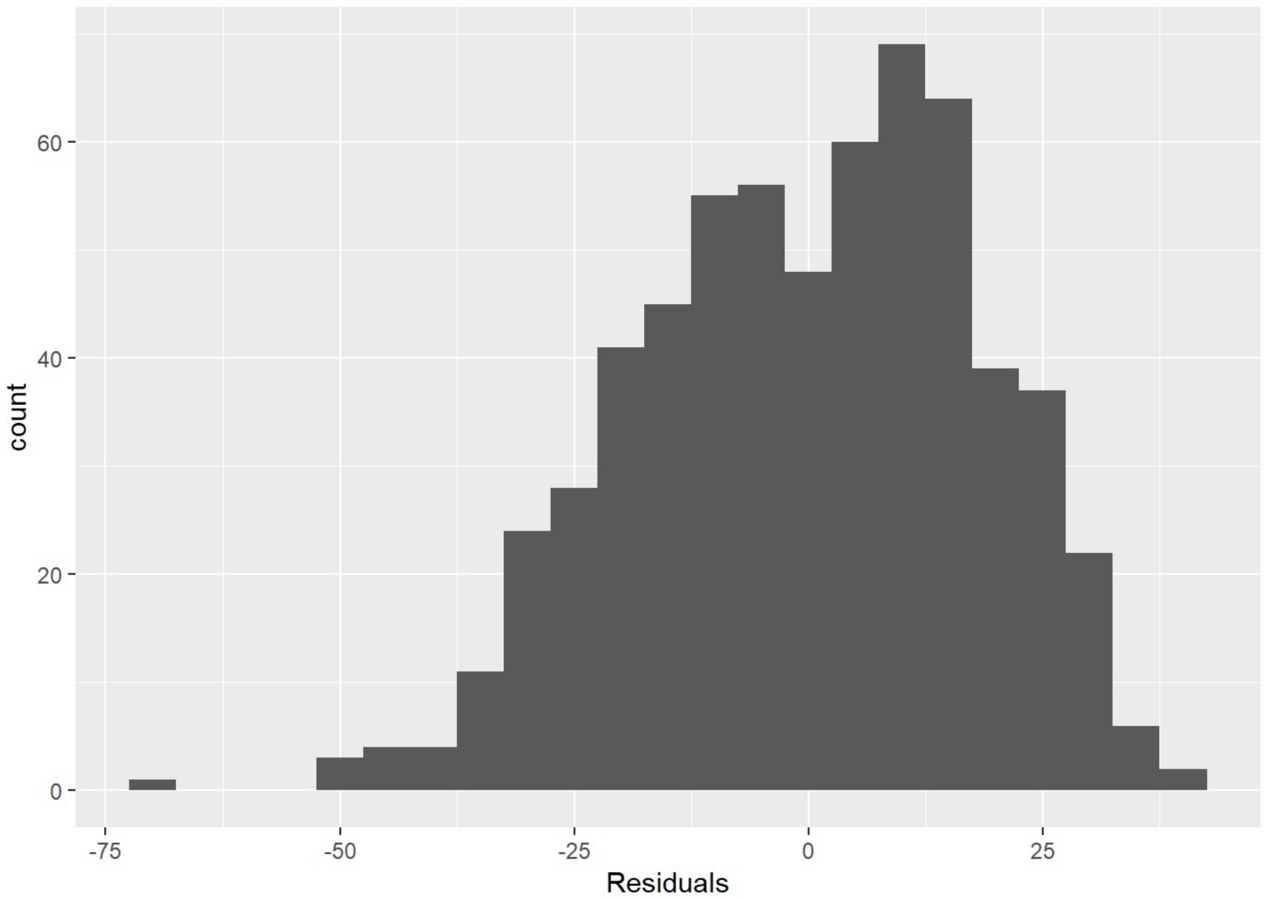
```
##                     P(B != 0 | Y)    model 1      model 2       model 3
## Intercept              1.00000000     1.0000     1.000000     1.0000000
## thtr_rel_day           0.04145787     0.0000     0.000000     0.0000000
## best_pic_nomyes        0.99998272     1.0000     1.000000     1.0000000
## best_pic_winyes        0.04171316     0.0000     0.000000     0.0000000
## best_actor_winyes      0.03951889     0.0000     0.000000     0.0000000
## best_actress_winyes    0.05197721     0.0000     0.000000     0.0000000
## best_dir_winyes        0.32483965     0.0000     0.000000     1.0000000
## top200_boxyes          0.62007574     1.0000     0.000000     1.0000000
## feature_filmYes        1.00000000     1.0000     1.000000     1.0000000
## dramaYes               0.99997876     1.0000     1.000000     1.0000000
## mpaa_rating_RYes       0.07778280     0.0000     0.000000     0.0000000
## oscar_seasonYes        0.04040533     0.0000     0.000000     0.0000000
## BF                             NA     1.0000     0.605389     0.4612518
## PostProbs                      NA     0.3106     0.188000     0.1433000
## R2                             NA     0.1846     0.174700     0.1910000
## dim                            NA     5.0000     4.000000     6.0000000
## logmarg                        NA -3801.7505 -3802.252367 -3802.5242941
##                           model 4       model 5
## Intercept               1.0000000  1.000000e+00
## thtr_rel_day            0.0000000  0.000000e+00
## best_pic_nomyes         1.0000000  1.000000e+00
## best_pic_winyes         0.0000000  0.000000e+00
## best_actor_winyes       0.0000000  0.000000e+00
## best_actress_winyes     0.0000000  0.000000e+00
## best_dir_winyes         1.0000000  0.000000e+00
## top200_boxyes           0.0000000  1.000000e+00
## feature_filmYes         1.0000000  1.000000e+00
## dramaYes                1.0000000  1.000000e+00
## mpaa_rating_RYes        0.0000000  1.000000e+00
## oscar_seasonYes         0.0000000  0.000000e+00
## BF                      0.3164738  9.828143e-02
## PostProbs               0.0983000  3.050000e-02
## R2                      0.1815000  1.869000e-01
## dim                     5.0000000  6.000000e+00
## logmarg             -3802.9009976 -3.804070e+03
```

We can see that the most probable model has one less variable than the min BIC model, best_dir_win, as its posterior probability is 0.32, less than 0.5. However, as 0.32 is still significant, we will stick to the min BIC model.

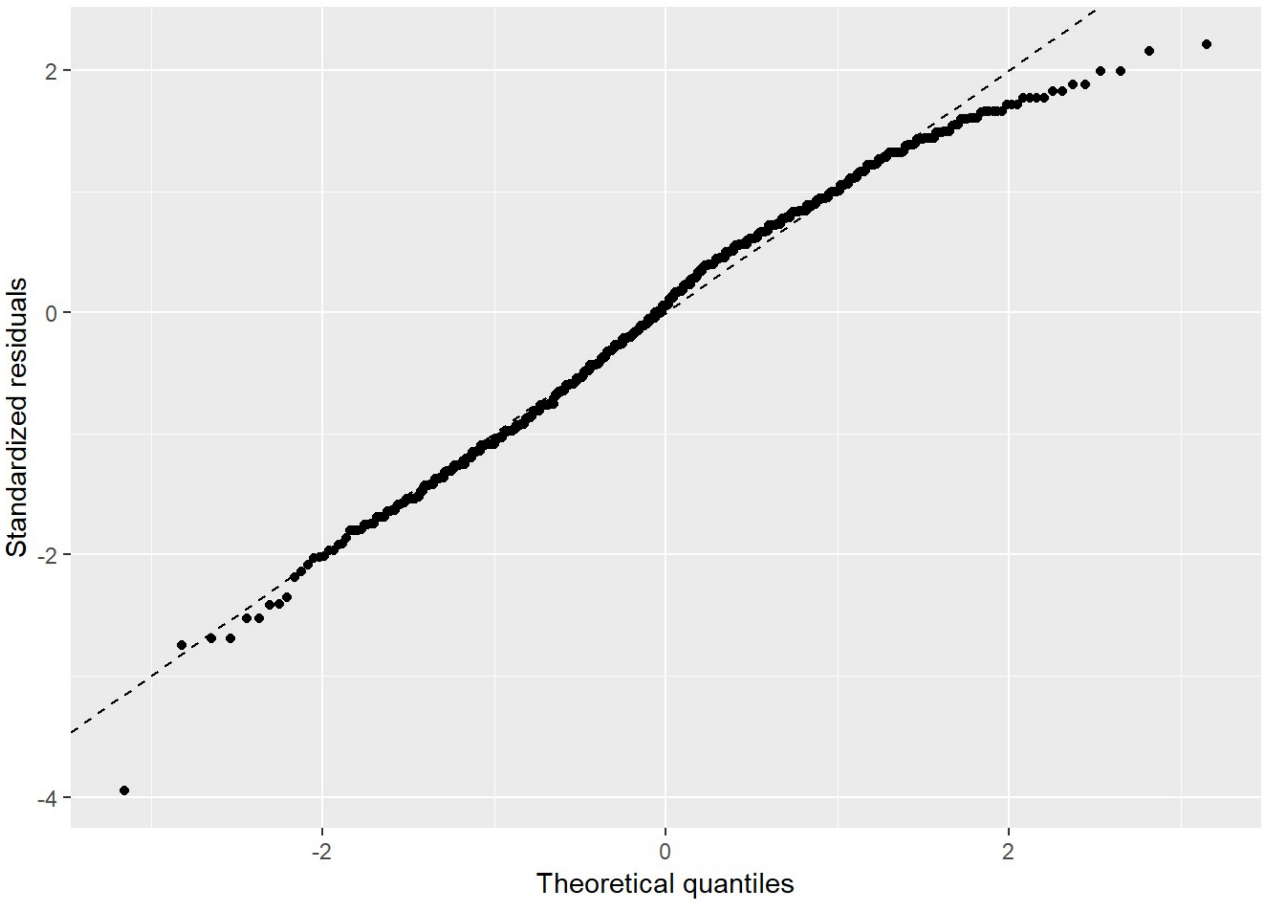
```
m_score_minbic_aug<-augment(m_score_minbic)

ggplot(data = m_score_minbic_aug, aes(x=.fitted, y=.resid))+
  geom_jitter(alpha = 0.6)+
  geom_hline(yintercept = 0, linetype = "dashed")+
  labs(x = "Fitted values", y="Residuals")
```



```
ggplot(data = m_score_minbic_aug, aes(x=.resid))+
  geom_histogram(binwidth = 5)+
  xlab("Residuals")
```



```
ggplot(m_score_minbic_aug)+
  geom_qq(aes(sample=.std.resid))+
  geom_abline(slope = 1, intercept = 0, linetype="dashed")+
  labs(x="Theoretical quantiles", y = "Standardized residuals")
```



Per the residual plot, residuals look randomly distributed around 0, with that one outlier that is far from 0. And regardless of the outlier, residuals are almost normally distributed per the histogram. The theoretical quantiles form almost a stright line, with larger variance towards the end. Therefore, the residuals are normally distributed. All in all, the diagnotics shows the conditions of a solid model have been met. Thus, we will move on to prediction.

Part 5: Prediction

```
# Prediction of Movie Oceans Eight
Oceans_Eight<-data.frame(best_pic_nom = "no",
best_dir_win = "no",
top200_box = "no",
feature_film = "Yes",
drama = "No")

predict(m_score_minbic, Oceans_Eight, interval = "prediction", level = 0.50)
```

```
##        fit      lwr      upr
## 1 54.75626 42.43504 67.07748
```

As 95% credible interval will yield a very wide score range that is not precise and almost includes all possible scores anyway, we will use a 50% CI to greatly increase precision of our prediction.

The movie’s information data comes from Rotten Tomatoes.

The point prediction is about 55 scores, and prediction interval with a 50% confidence level is (42, 67), which is a pretty narrow and precise range. The actual Rotten Tomatoes score is 44, which is fairly close to our predicted point and is included in our interval. Therefore, the model worked out well on this movie.

Part 6: Conclusion

The final model shows that out of the 12 possible variables I fitted in to start with, only 5 of them have significant impact on audience score. And it specifically excluded two varibles that would have fitted in the frequentist model I came out in the last project, best actor win and mpaa_rating, which is interesting, as intuitively, an Oscar winner is more likely to film a high quality movie.

My model comes from the one that minimizes the BIC. Other model selection cretiria could generate a different model (such as the most probable model shown above). And my model excludes studio to start with to simply the modeling. But this variable intuitively could be an important factor to audience scores, as certain studios make better quality films.