Modeling and prediction for movies

Setup

Load packages

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

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

This data set collects information about random sampled movies produced and released before 2016, for example how much audiences and critics like movies. Overall, there are 651 observations and 32 variables. Since random sampling is used, this sample could be generalized to all movies produced and released before 2016. However, this may lead to a bias because they may get easier access to English movies rather than other languages all over the world. Furthermore, this is an observational study without random assignment, so no causality could be conducted.

Part 2: Research question

The popularity of a movie is always an important business concern. Based on this data set, we may want to explore which of those related varibles may be associoated with the audience response to relevant movies

---

Part 3: Exploratory data analysis

At first, we get a quick glance of this data set.

```
names(movies)
```

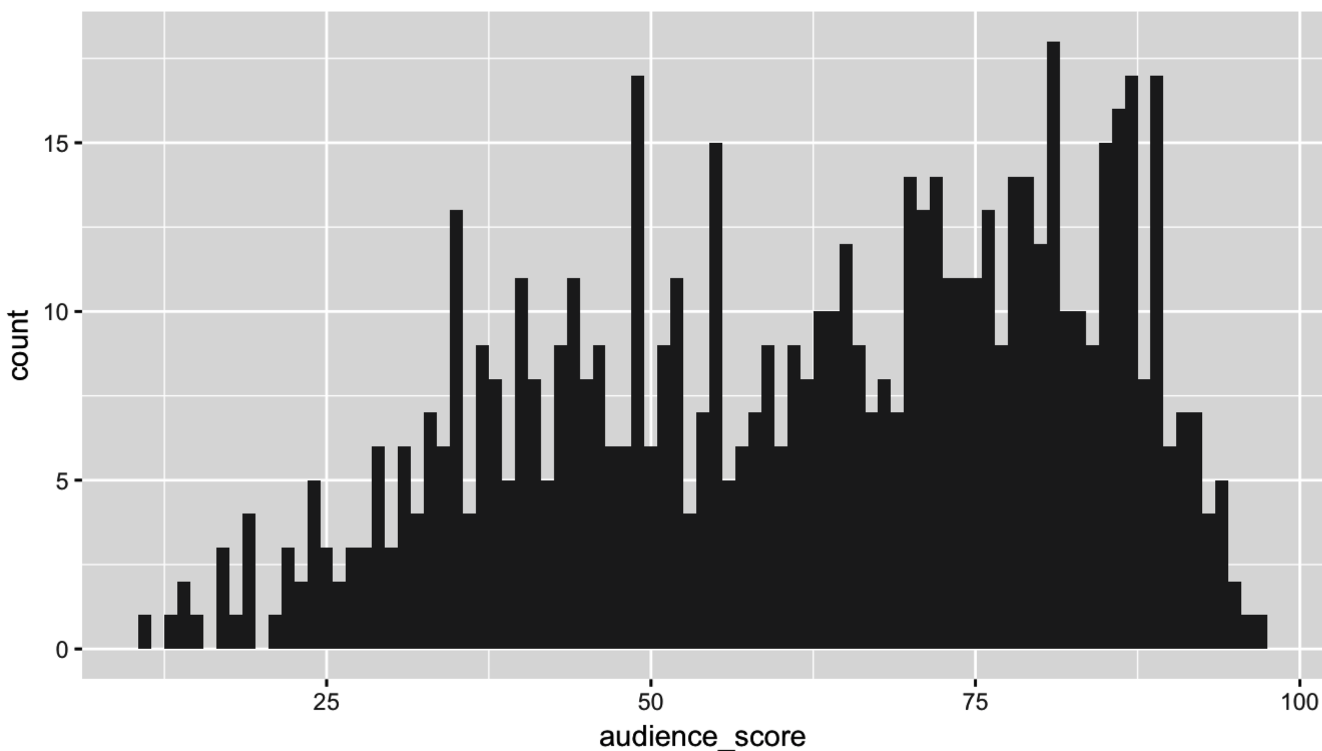
```
##  [1] "title"            "title_type"       "genre"
##  [4] "runtime"          "mpaa_rating"      "studio"
##  [7] "thtr_rel_year"    "thtr_rel_month"   "thtr_rel_day"
## [10] "dvd_rel_year"     "dvd_rel_month"    "dvd_rel_day"
## [13] "imdb_rating"      "imdb_num_votes"   "critics_rating"
## [16] "critics_score"    "audience_rating"  "audience_score"
## [19] "best_pic_nom"     "best_pic_win"     "best_actor_win"
## [22] "best_actress_win" "best_dir_win"     "top200_box"
## [25] "director"         "actor1"           "actor2"
## [28] "actor3"           "actor4"           "actor5"
## [31] "imdb_url"         "rt_url"
```

Here we choose `audience_score` to be our response variable and select some variables of our intrest that maybe associoated with 'audience_score', then we put them together in a new data frame `audience_rep`.

```
audience_rep <- movies %>%
  select(audience_score, genre, runtime, thtr_rel_year, critics_score, best_pic_win, be
st_actor_win, best_actress_win, best_dir_win, top200_box) %>%
  na.omit()
```

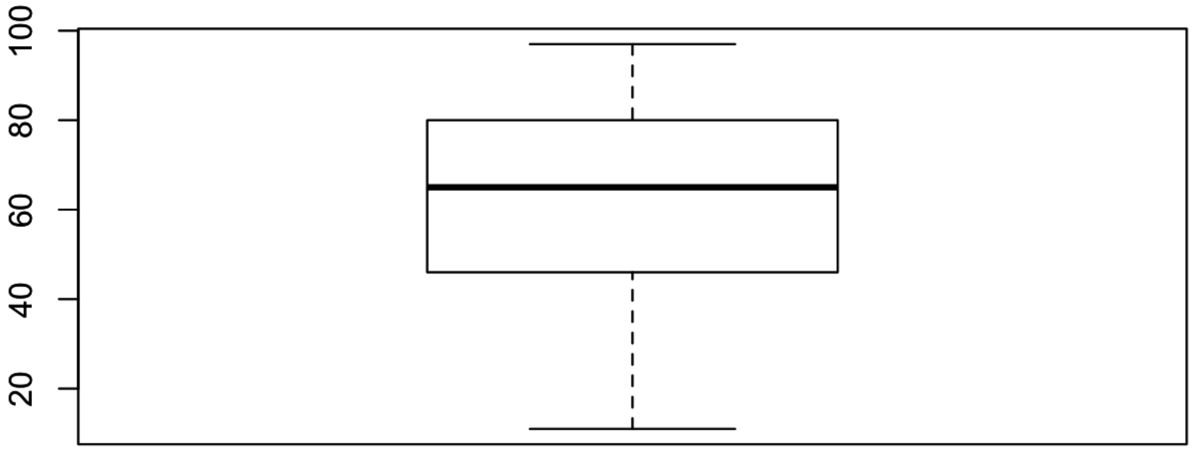
To get familiar with our response varible, we draw a histogram to see the distribution of `audience_score`.

```
ggplot(data = audience_rep, aes(x=audience_score)) +
  geom_histogram(binwidth = 1)
```



We can see that the distribution of `audience_score` is left skewed, we also draw a boxplot and summary its quantile to get further details.

```
boxplot(audience_rep$audience_score)
```



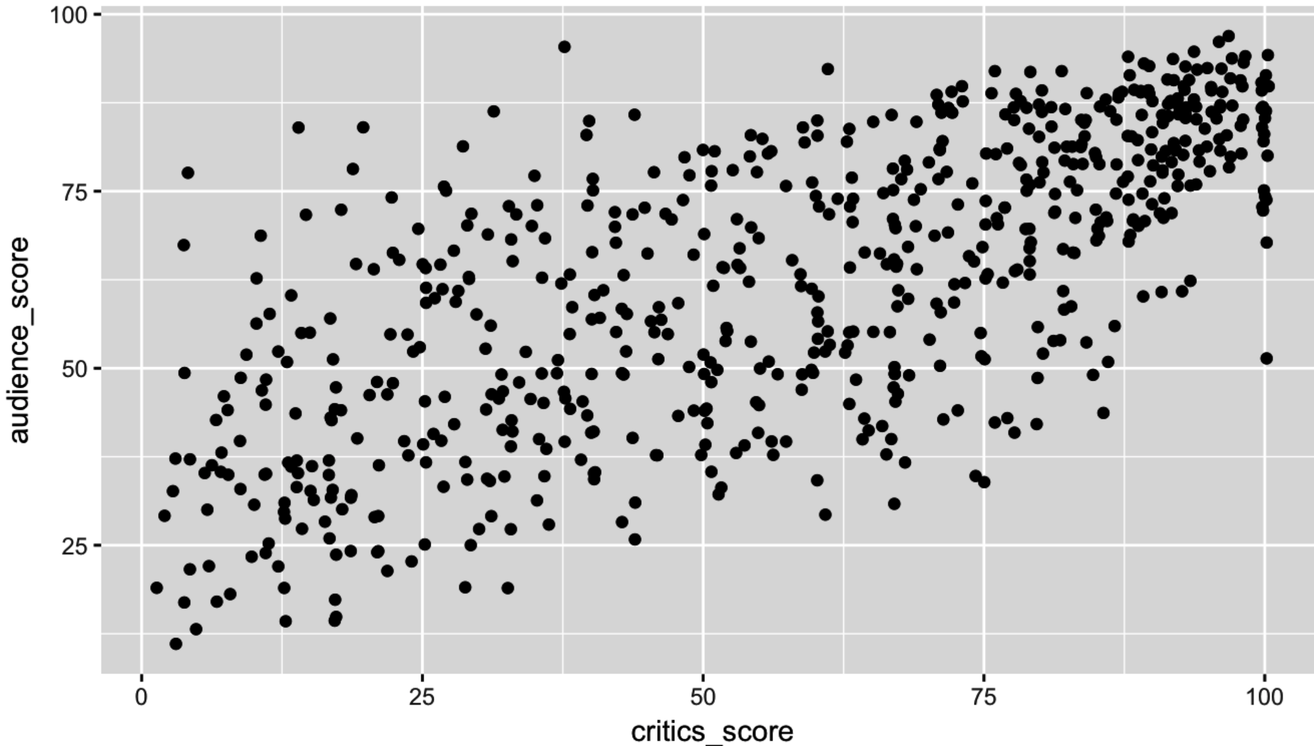

```
quantile(audience_rep$audience_score)
```

```
##   0%  25%  50%  75% 100% 
##   11   46   65   80   97
```

The value of `audience_score` in this sample rangs from 11 to 97 with a median score of 65.

Next, let's pick several variables to explore their relationship with `audience_score` , `critics_score` to be the first one.

```
ggplot(data = audience_rep, aes(x=critics_score, y=audience_score)) +
  geom_jitter()
```



It seems that there is a positive linear association, furthermore we can calculate the correlation coefficent.

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

```
## # A tibble: 1 × 1
##   `cor(audience_score, critics_score)`
##                                  <dbl>
## 1                            0.7041573
```

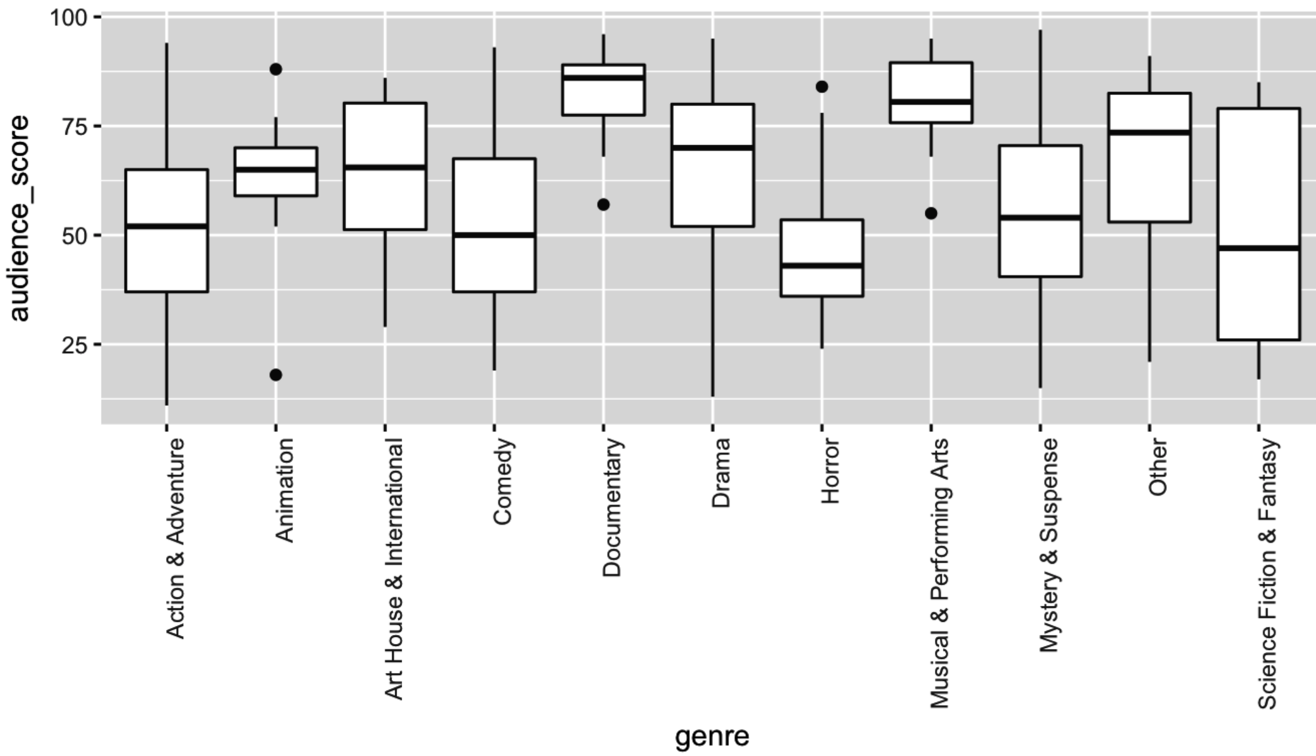
As expected, there is a strong correlation between `audience_score` and `critics_score` . Then we may want to know if the director of a movie who may ever win an Oscar would affect `audience_score` .

```
audience_rep %>%
  group_by(best_dir_win) %>%
  summarise(median=median(audience_score), sd=sd(audience_score))
```

```
## # A tibble: 2 × 3
##   best_dir_win median       sd
##         <fctr>  <dbl>    <dbl>
## 1           no     65 20.24020
## 2          yes     73 18.96535
```

From the summaries above, median scores grouped by `best_dir_win` have an obvious difference which may imply an association between `best_dir_win` and `audience_score` . In addition, genre of a movie may also be affective.

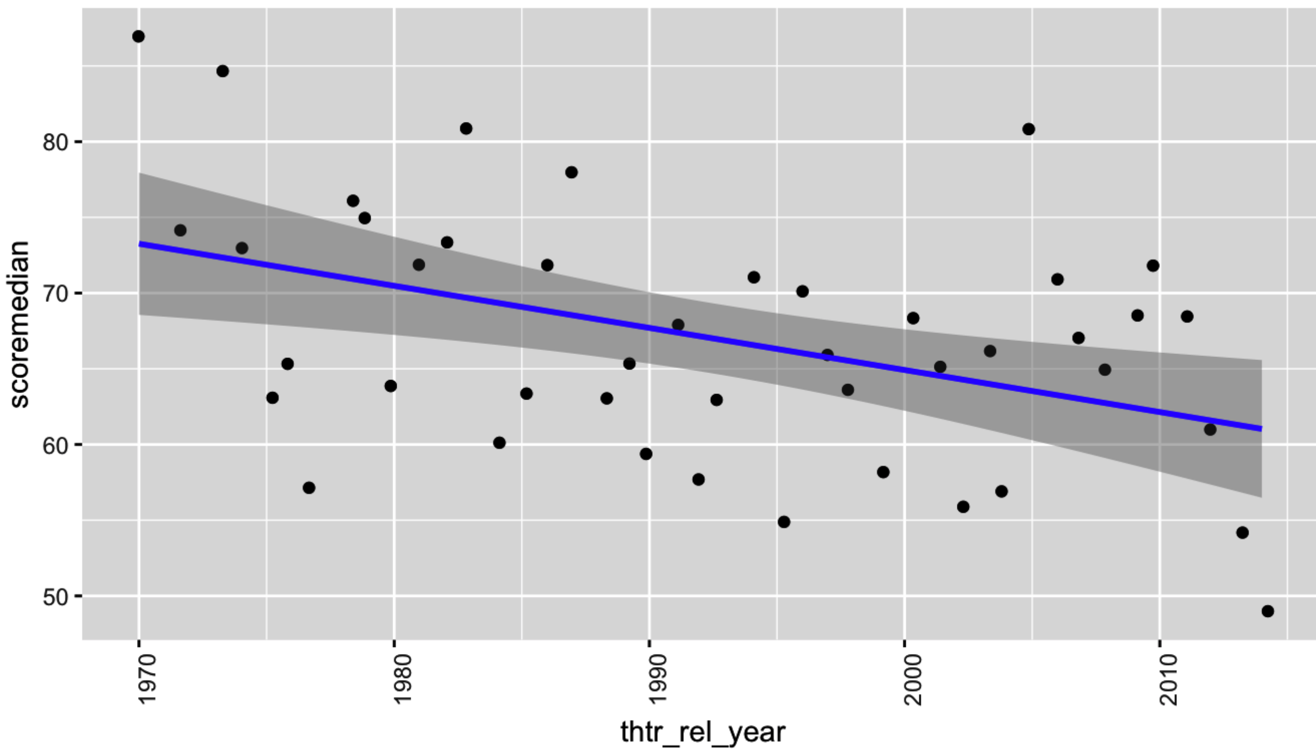
```
ggplot(data = audience_rep, aes(genre, audience_score))+
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



As shown in this side-by-side plot, the distribution of `audience_score` varies a lot among different genres. ‘Documentary’ movies have the highest audience score.

Finally, we could also examine if audience tend to response differently on movies released in different years.

```
medianscore_year <- audience_rep %>%
  group_by(thtr_rel_year) %>%
  summarise(scoremedian = median(audience_score))
ggplot(data = medianscore_year, aes(x = thtr_rel_year, y = scoremedian)) +
  geom_jitter() +
  stat_smooth(method = lm) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



There seems to be an slightly decrease in meadian score from 1970 to 2014.

Overall, we can see some of our selected variable seemed to be associated with `audience_score` , for example `critics_score` , `best_dir_win` , `genre` and so on. However, if we want to predict `audience_score` with the related variables, a multiple linear model is needed.

Part 4: Modeling

In this part we develop a multiple linear regression model to predict `audience_score` which we choose to indicate the popularity of a movie. As `audience_score` to be the response variable, we choose `runtime` , `thtr_rel_year` , `critics_score` , `best_pic_win` , `top200_box` to be our explanatory variables which would be in the full model. In addition, we find that 'Documentary' movies have the higher audience score than other types. We would creat a new variable `documentory` .

```
audience_rep <-audience_rep %>%
  mutate(documentary=ifelse(genre=='Documentary', 'Yes','No'))
```

In full model, our predictors for `audience_score` would be `documentary` , `runtime` , `thtr_rel_year` , `critics_score` , `best_pic_win` and `top200_box` .

However, we excluding some variables like `title` , `director` , `actor1` … `actor5` , `imdb_url` , `rt_url` , because these varibles only supply very detail information about certain movies which is meaningless for prediction. Other variables like `best_actor_win` however has very weak association with `audience_score` as analysed in part 3.

Certain model selection method is needed to search for the best model. Here we would use backwards elimination using adjusted R^2 approach for a more reliable prediction.

We would start the multiple regression with the full model introducted above.

```
m_full <- lm(audience_score ~ documentary + runtime +thtr_rel_year +critics_score +best
_pic_win +top200_box, data = audience_rep)
summary(m_full)
```

```
##
## Call:
## lm(formula = audience_score ~ documentary + runtime + thtr_rel_year +
##     critics_score + best_pic_win + top200_box, data = audience_rep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.611  -9.597   0.661  10.211  41.744
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      52.65656  103.93805   0.507  0.61260
## documentaryYes    8.89519    2.24568   3.961  8.3e-05 ***
## runtime           0.08054    0.03036   2.653  0.00818 **
## thtr_rel_year    -0.01319    0.05186  -0.254  0.79933
## critics_score     0.46325    0.02139  21.658  < 2e-16 ***
## best_pic_winyes   4.08190    5.54325   0.736  0.46177
## top200_boxyes     3.00977    3.76767   0.799  0.42468
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.2 on 643 degrees of freedom
## Multiple R-squared:  0.5121, Adjusted R-squared:  0.5075
## F-statistic: 112.5 on 6 and 643 DF,  p-value: < 2.2e-16
```

```
adjr_full=summary(m_full)$adj.r.squared
```

Next, we try to remove one predictor from the full model and pick the one would yeild biggest adjusted R^2. First creat a new model which drop `documentary` and check the adjusted R^2.

```
m1 <- lm(audience_score ~ runtime +thtr_rel_year +critics_score +best_pic_win +top200_b
ox, data = audience_rep)
adjr_documentary=summary(m1)$adj.r.squared
adjr_documentary
```

```
## [1] 0.4962925
```

Then, try dropping next variable from the full model.(`runtime`)

```
m1 <- lm(audience_score ~ documentary +thtr_rel_year +critics_score +best_pic_win +top2
00_box, data = audience_rep)
adjr_runtime=summary(m1)$adj.r.squared

m1 <- lm(audience_score ~ documentary +runtime +critics_score +best_pic_win +top200_bo
x, data = audience_rep)
adjr_year=summary(m1)$adj.r.squared

m1 <- lm(audience_score ~ documentary +runtime +thtr_rel_year +best_pic_win +top200_bo
x, data = audience_rep)
adjr_critscore=summary(m1)$adj.r.squared

m1 <- lm(audience_score ~ documentary + runtime + thtr_rel_year + critics_score  + top2
00_box, data = audience_rep)
adjr_bestpic=summary(m1)$adj.r.squared

m1 <- lm(audience_score ~ documentary + runtime + thtr_rel_year + critics_score + best_
pic_win, data = audience_rep)
adjr_top200=summary(m1)$adj.r.squared

adjr1=max(adjr_bestpic, adjr_critscore, adjr_documentary, adjr_runtime, adjr_top200, ad
jr_year)
```

After comparing the adjusted R^2 of the models above, we decide to remove `thtr_rel_year`. Then we try to remove another variable remained.

```
m2 <- lm(audience_score ~ runtime +critics_score +best_pic_win +top200_box, data = audi
ence_rep)
adjr_documentary=summary(m2)$adj.r.squared

m2 <- lm(audience_score ~ documentary +critics_score +best_pic_win +top200_box, data =
audience_rep)
adjr_runtime=summary(m2)$adj.r.squared

m2 <- lm(audience_score ~ documentary +runtime +best_pic_win +top200_box, data = audien
ce_rep)
adjr_critscore=summary(m2)$adj.r.squared

m2 <- lm(audience_score ~ documentary + runtime + critics_score  + top200_box, data = a
udience_rep)
adjr_bestpic=summary(m2)$adj.r.squared

m2 <- lm(audience_score ~ documentary + runtime + critics_score + best_pic_win, data =
audience_rep)
adjr_top200=summary(m2)$adj.r.squared

adjr2=max(adjr_bestpic, adjr_critscore, adjr_documentary, adjr_runtime, adjr_top200)
```

Then we find adjusted R^2 of the `best_pic_win` dropped model is biggest among this five models and bigger than `adjr1`. Then, we try to drop a third one.

```r
m3 <- lm(audience_score ~ runtime +critics_score +top200_box, data = audience_rep)
adjr_documentary=summary(m3)$adj.r.squared

m3 <- lm(audience_score ~ documentary +critics_score +top200_box, data = audience_rep)
adjr_runtime=summary(m3)$adj.r.squared

m3 <- lm(audience_score ~ documentary +runtime +top200_box, data = audience_rep)
adjr_critscore=summary(m3)$adj.r.squared

m3 <- lm(audience_score ~ documentary + runtime + critics_score, data = audience_rep)
adjr_top200=summary(m3)$adj.r.squared

adjr3=max(adjr_critscore, adjr_documentary, adjr_runtime, adjr_top200)
```

we find adjusted R^2 of the `top200_box` dropped model is biggest among this five models and bigger than `adjr2` and dropping would continue.

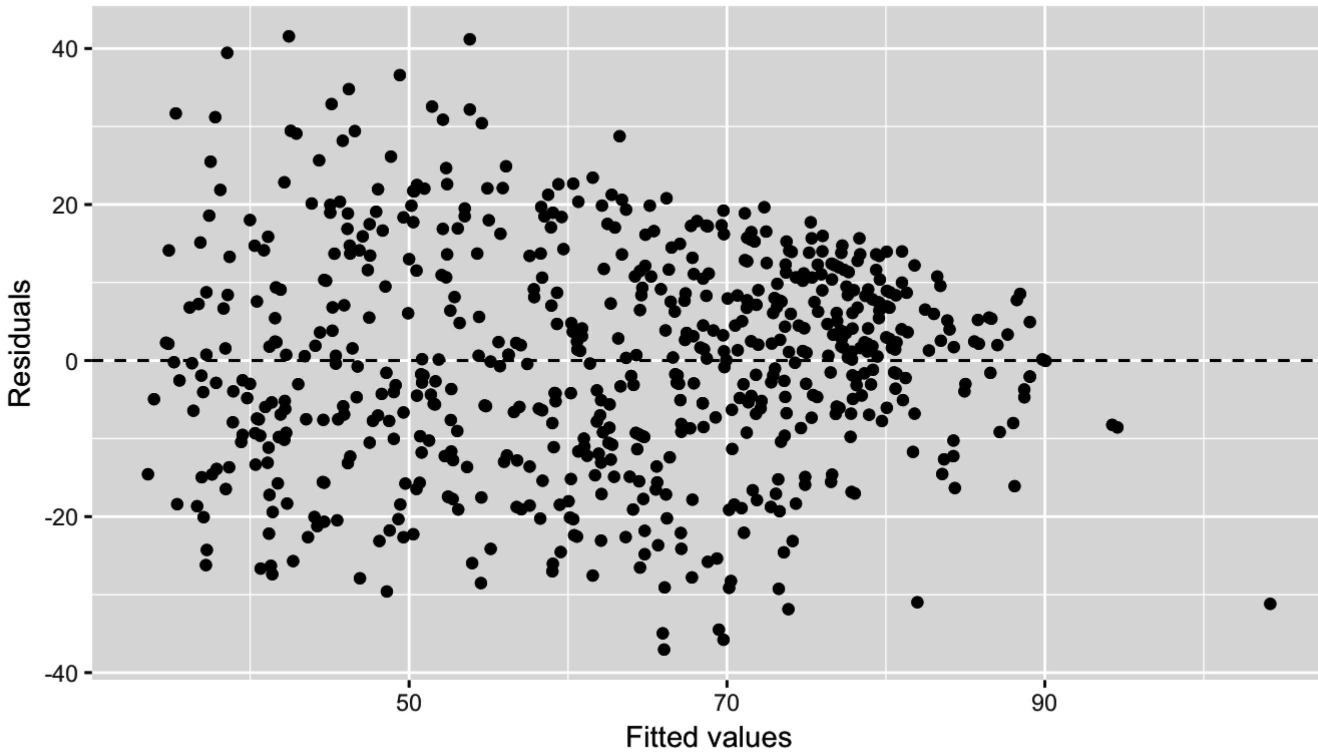
```r
m4 <- lm(audience_score ~ runtime +critics_score, data = audience_rep)
adjr_documentary=summary(m4)$adj.r.squared

m4 <- lm(audience_score ~ documentary +critics_score, data = audience_rep)
adjr_runtime=summary(m4)$adj.r.squared

m4 <- lm(audience_score ~ documentary +runtime, data = audience_rep)
adjr_critscore=summary(m4)$adj.r.squared
```

This time none of the models could yeild a higher adjusted R^2 than `adjr3` , so we would not drop a third variable. Finally, our parsimonious model for predicting `audience_score` will be like:

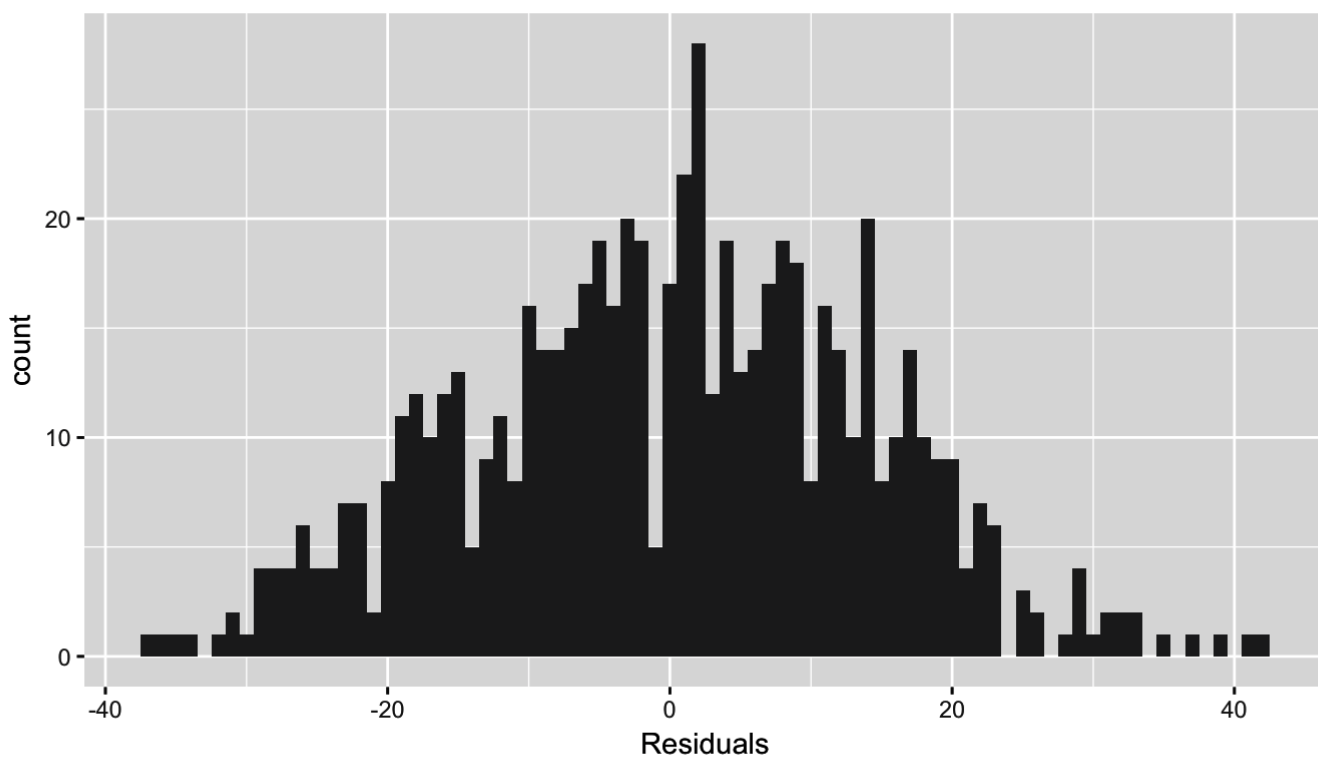
```r
m_final <- lm(audience_score ~ documentary + runtime + critics_score, data = audience_r
ep)
```

To check our model, model diagnostics should be done on this final model.

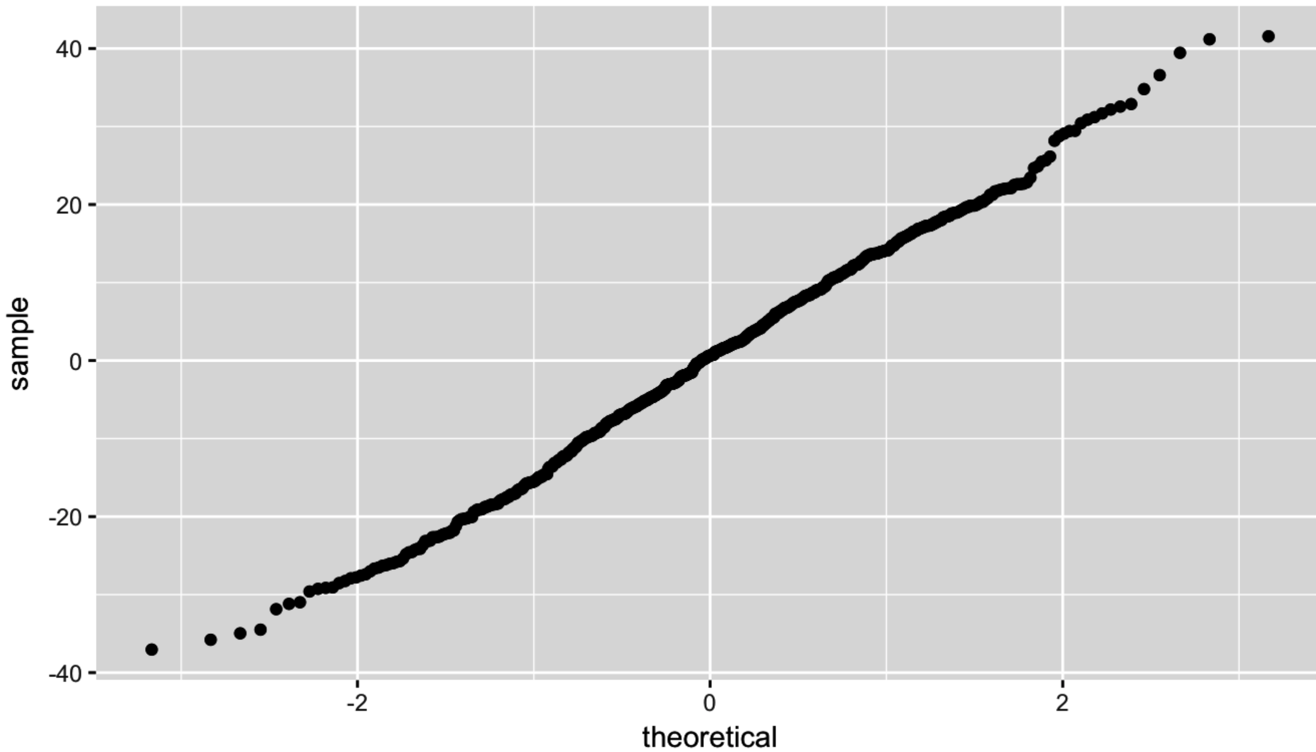
```r
ggplot(data = m_final, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```




```
ggplot(data = m_final, aes(x = .resid)) +
  geom_histogram(binwidth = 1) +
  xlab("Residuals")
```




```
ggplot(data = m_final, aes(sample = .resid)) +
  stat_qq()
```



As shown, despite the residual scatter plot is not very ideal, the histogram plot shows that the residual distribution is fairly normal as well as the qq-plot. So, we can say our model is constructed reasonably.

```r
summary(m_final)
```

```
##
## Call:
## lm(formula = audience_score ~ documentary + runtime + critics_score,
##     data = audience_rep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.044  -9.632   0.623  10.227  41.562
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    25.49954    3.16824   8.048 4.03e-15 ***
## documentaryYes  8.65288    2.21199   3.912 0.000101 ***
## runtime         0.08741    0.02968   2.945 0.003348 **
## critics_score   0.46691    0.02106  22.170  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.18 on 646 degrees of freedom
## Multiple R-squared:  0.5111, Adjusted R-squared:  0.5088
## F-statistic: 225.1 on 3 and 646 DF,  p-value: < 2.2e-16
```

Based on our final model, we could interpret that as below:

All else held constant, each 1 point increase in `critics_score` , the model predicts `audience_score` to be higher on average by 0.47 point.

All else held constant, each 1 minite increase in `runtime`, the model predicts `audience_score` to be higher on average by 0.09 point.

All else held constant, the model predicts that movies whose genre is documentary are expected to has an increase in `audience_score` by 8.65 than other type movies, on average.

Howerver, the intercept here doesn't have real meaning.

Part 5: Prediction

In this part we want to use the model `m_final` to predict `audience_score` for a movies from 2016. First, we need to creat a new data frame for this new movie.

```
newmovie2016 <- data.frame(title="Fantastic Beats and Where to Find Them", documentary
 = "No", runtime = 132, critics_score = 75)
```

Then, we can do prediction using the `predict` function.

```
predict(m_final, newmovie2016)
```

```
##        1
## 72.05618
```

For the measure of prediction uncertainty, we construct a prediction interval.

```
predict(m_final, newmovie2016, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 72.05618 44.14384 99.96851
```

The model predicts, with 95% confidence, that a 132 minites long movie which is not a documentary and scored by critics at 75 point is expected to have a `audience_score` between 44.14 and 99.97.

reference:

URL: https://www.rottentomatoes.com/m/fantastic_beasts_and_where_to_find_them (https://www.rottentomatoes.com/m/fantastic_beasts_and_where_to_find_them)

Part 6: Conclusion

As a conclusion, we use a dataset which collect information about randomly sampled movies from 1970 to 2014 to pick significant predictors for audience scores for relevant movies. At the end of this analysis, we construct a multiple linear model to predict `audience_score` using the selected predictors, `documentary`, `runtime` and `critics_score`. However, there maybe other features that associated with audience scores that not included in this dataset, which would affect model efficiency.