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.

```
setwd("F:/specialization/22-Master Statistics with R (Duke University)/data")
movies <- readRDS("movies01.rds")
```

Part 1: Data

The data set is comprised of 651 randomly sampled movies produced and released before 2016. The data is about how much audiences and critics like movies as well as numerous other variables about the movies. Tt includes information from Rotten Tomatoes and IMDB for a random sample of movies.

We are interested in learning what attributes make a movie popular also interested in learning something new about movies.

Part 2: Research question

Does the data suggest that, critics rating in Rotten Tomatoes has relationship with audience score and does number of awards correlated with both the explantory and response variables?

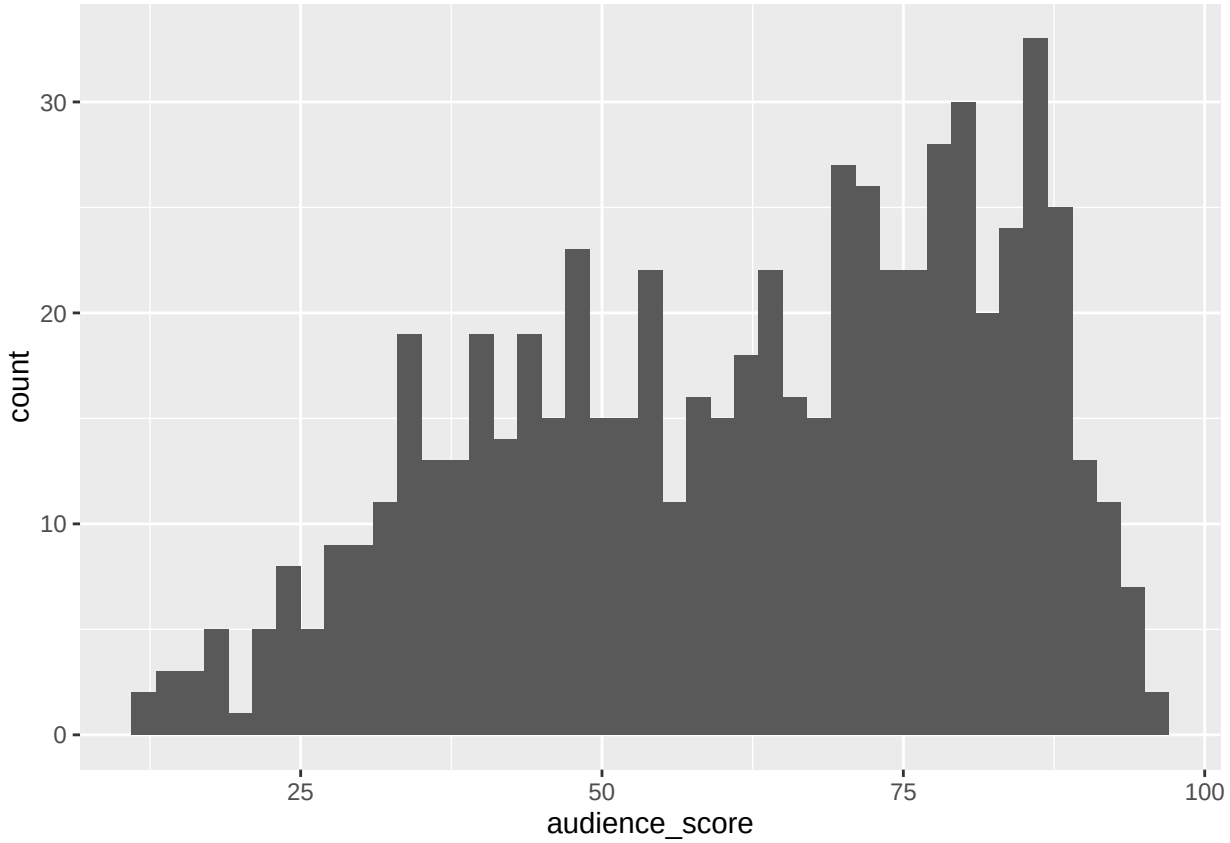
Exploratory variable : Categorical variable for audience rating on Rotten Tomatoes (Spilled, Upright)
– critics_rating
Comfonding variable : Categorical variable for critics rating on Rotten Tomatoes (Certified Fresh, Fresh, Rotten)
– audience_rating
Response variable : Audience score on Rotten Tomatoes
– audience_score

Part 3: Exploratory data analysis

Audience Score

Checking the skewness

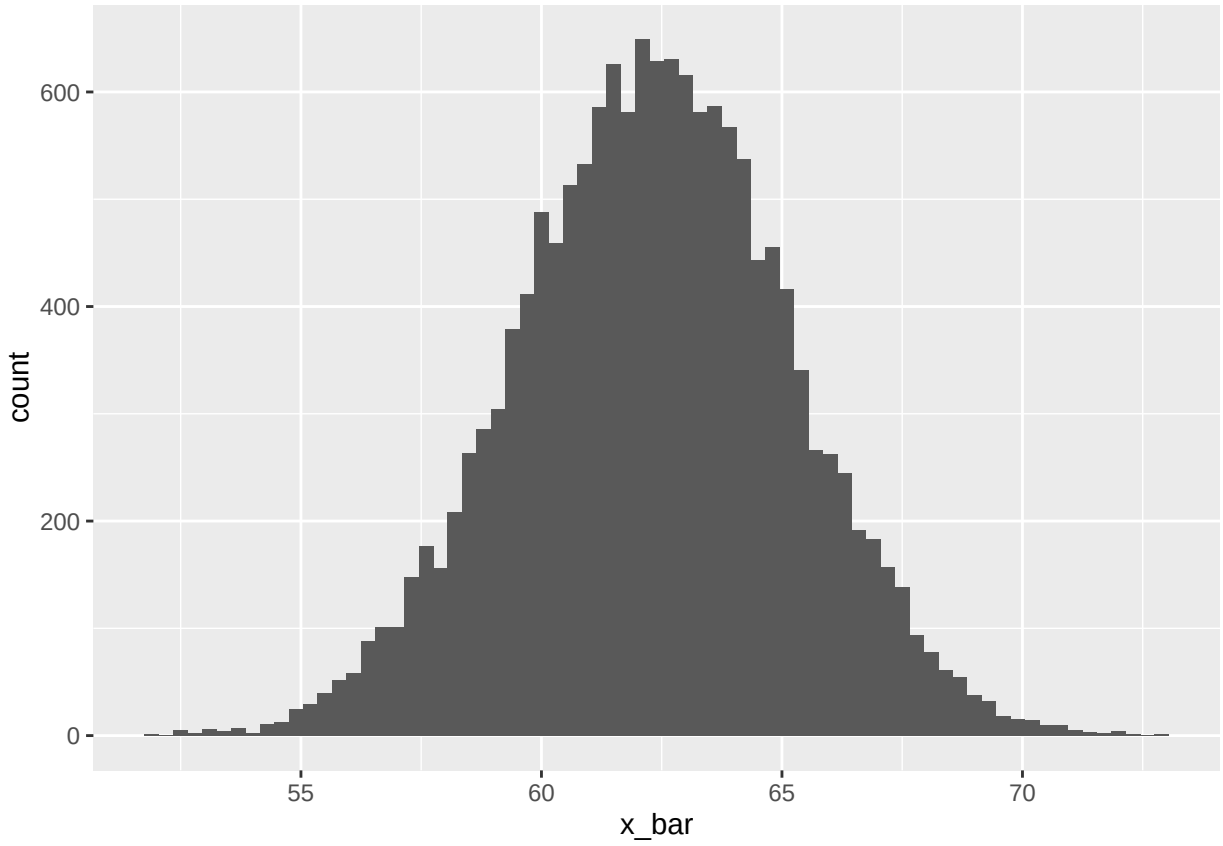
```
ggplot(movies, aes(audience_score)) +
     geom_histogram(binwidth = 2)
```



Okay it has such strong skewness of the acore distribution. but I want nearly perfect normal distribution for further inference.

Apply Sampling Distribution

```
sample_means50 <- movies %>%
     rep_sample_n(size = 50, reps = 15000, replace = TRUE) %>%
     summarise(x_bar = mean(audience_score))
ggplot(data = sample_means50, aes(x = x_bar)) +
  geom_histogram(binwidth = .3)
```



Okay now we can build 95% CI with Sampling Distribution.

Building 95% CI with Sampling Distribution

```
z_star_95 <- qnorm(.975)
z_star_95
```

```
## [1] 1.959964
```

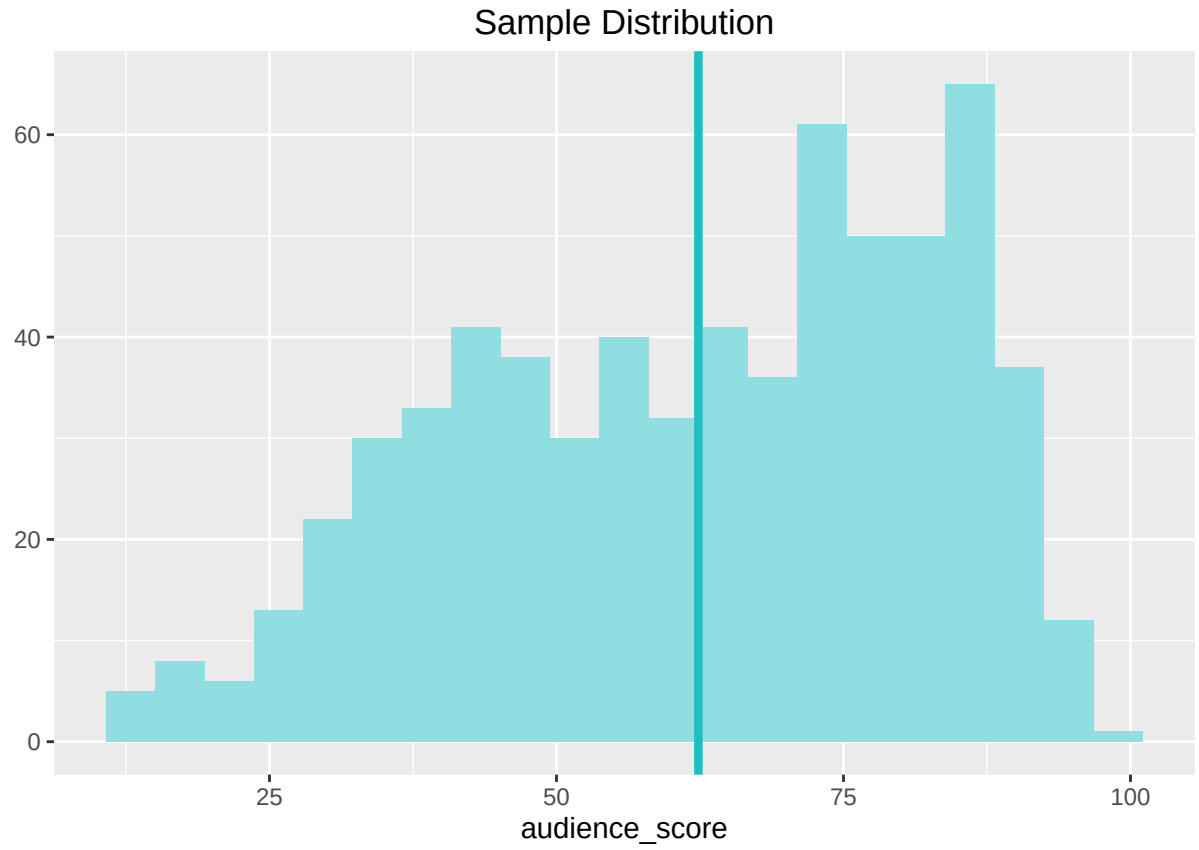
```
n <- 60
ci <- movies %>%
     rep_sample_n(size = 60, reps = 50, replace = TRUE) %>%
     summarise(lower = mean(audience_score) - z_star_95 * (sd(audience_score) / sqrt(n)),
               upper = mean(audience_score) + z_star_95 * (sd(audience_score) / sqrt(n)))
ci %>%
  slice(1:5)
```

```
## Source: local data frame [5 x 3]
##
##   replicate    lower    upper
##       (int)    (dbl)    (dbl)
## 1         1 53.30635 63.72698
## 2         2 60.94195 70.55805
## 3         3 57.48692 68.34642
## 4         4 61.57157 71.46177
## 5         5 59.31783 69.18217
```

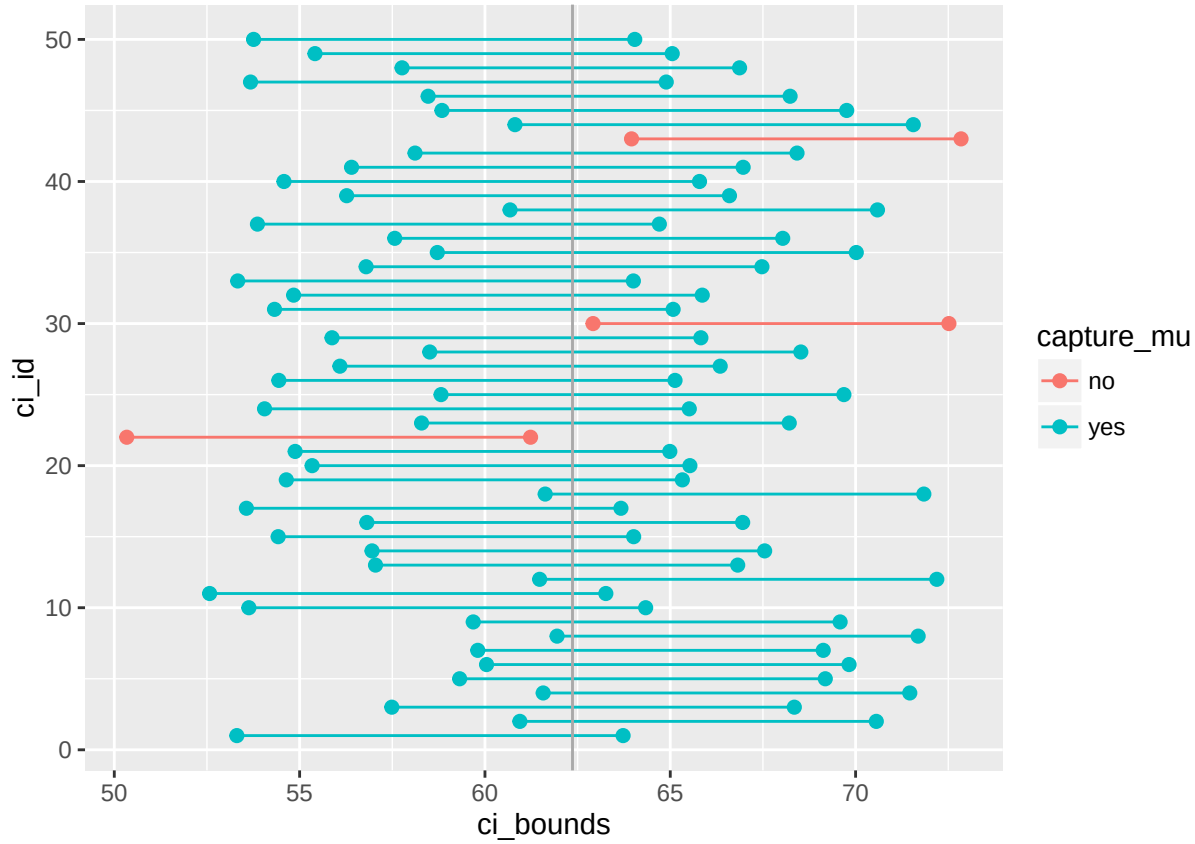
Plotting Confidence Interval

```
inference(y = audience_score, data = movies, statistic = "mean", type = "ci",
          conf_level = 0.95, method = "theoretical", order = c("Yes","No"))
```

```
## Single numerical variable
## n = 651, y-bar = 62.3625, s = 20.2226
## 95% CI: (60.8062 , 63.9189)
```



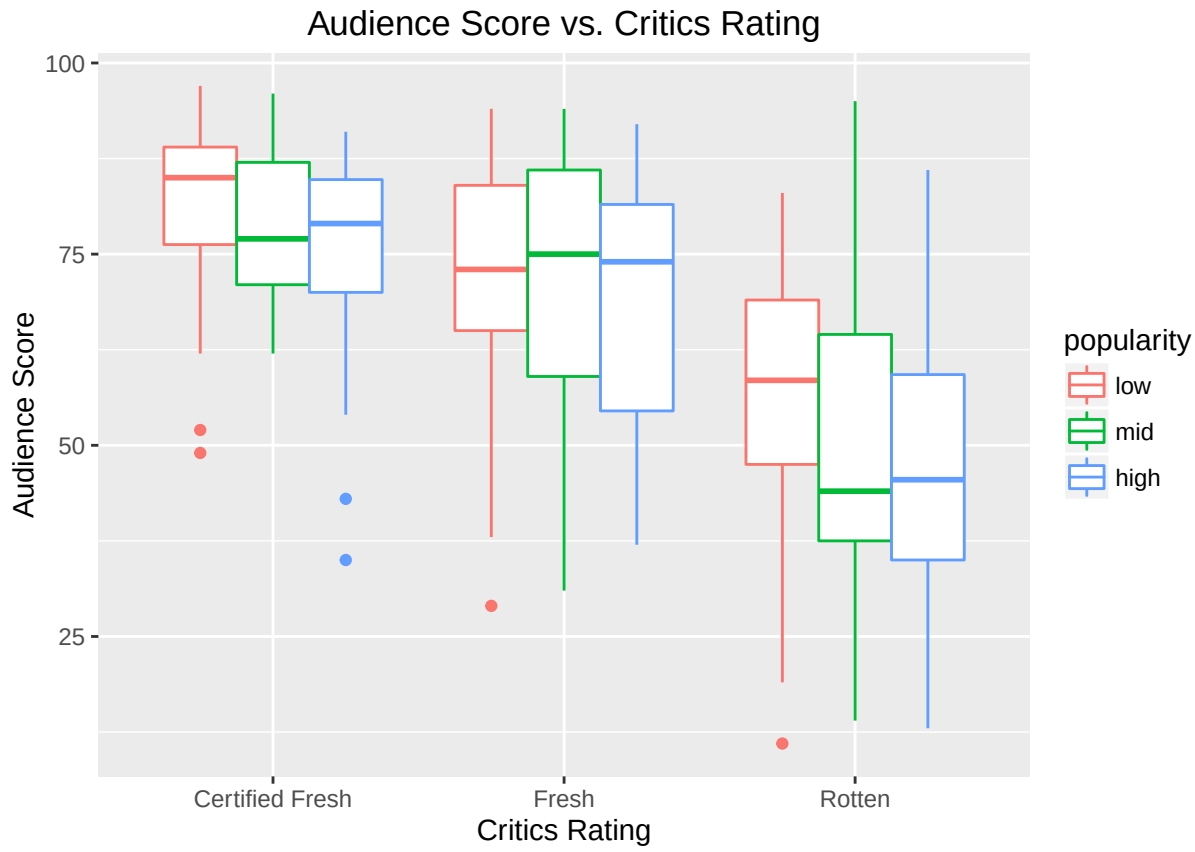

```
# true population mean
params <- movies %>%
     summarise(mu = mean(audience_score))

ci <- ci %>%
     mutate(capture_mu = ifelse(lower < params$mu & upper > params$mu, "yes", "no"))

ci_data <- data.frame(ci_id = c(1:50, 1:50),
                      ci_bounds = c(ci$lower, ci$upper),
                      capture_mu = c(ci$capture, ci$capture))

ggplot(data = ci_data, aes(x = ci_bounds, y = ci_id, group = ci_id, color = capture_mu)) +
     geom_point(size = 2) +  # add points at the ends, size = 2
     geom_line() +           # connect with lines
     geom_vline(xintercept = params$mu, color = "darkgray") # draw vertical line
```



Interpretation: The population mean of audience score lies between these two bounds of 60.8062 , 63.9189, since 95% of the time confidence intervals contain the true mean.

Research quesion:

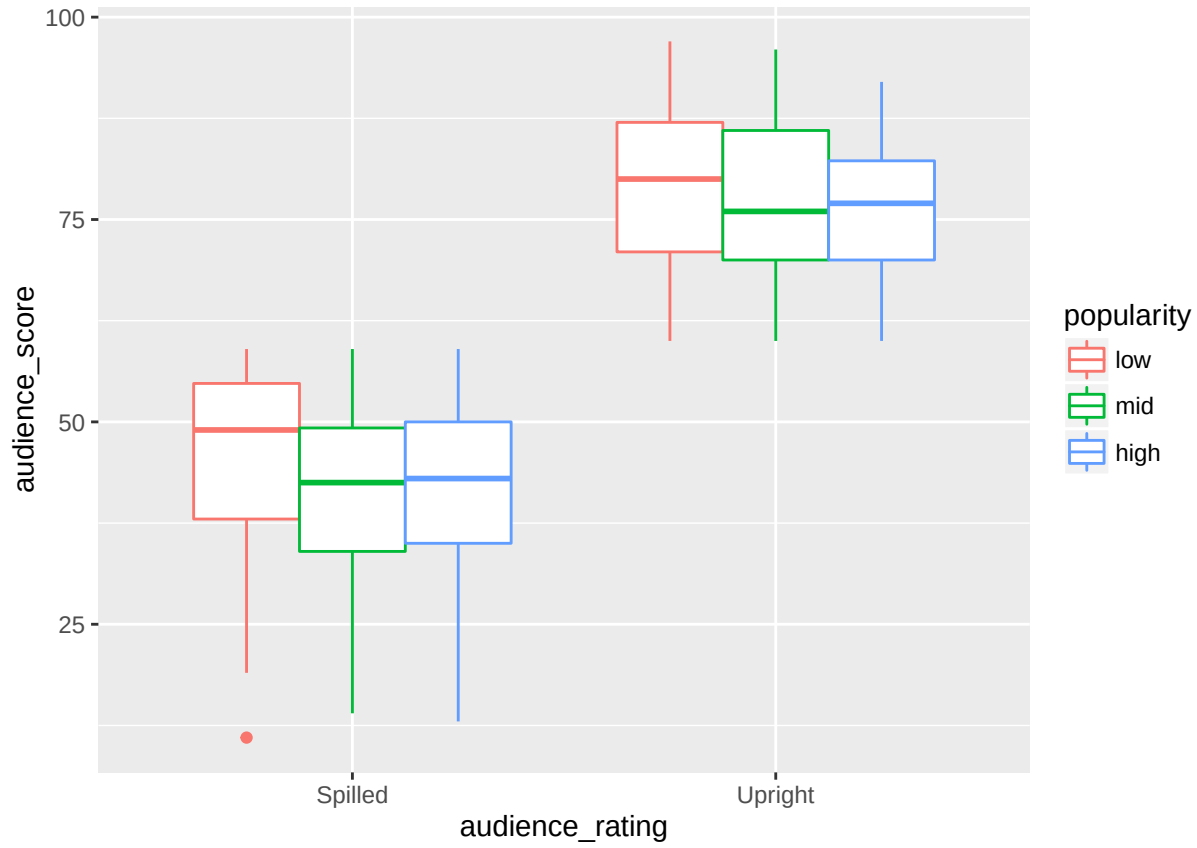
```
ggplot(movies, aes(critics_rating, audience_score)) +
     geom_boxplot(aes(colour=popularity)) +
     labs(title = "Audience Score vs. Critics Rating") +
     xlab("Critics Rating") +
     ylab("Audience Score")
```




```
movies %>%
     group_by(critics_rating) %>%
     summarise(mean(audience_score))
```

```
## Source: local data frame [3 x 2]
##
##    critics_rating mean(audience_score)
##            (fctr)                (dbl)
## 1 Certified Fresh             79.37037
## 2           Fresh             69.97129
## 3          Rotten             49.70358
```

```
ggplot(movies, aes(audience_rating, audience_score)) +
     geom_boxplot(aes(colour=popularity))
```




```
movies %>%
     group_by(audience_rating) %>%
     summarise(mean(audience_score))
```

```
## Source: local data frame [2 x 2]
##
##   audience_rating mean(audience_score)
##            (fctr)                (dbl)
## 1         Spilled             41.93455
## 2         Upright             77.30319
```

Interpretation:
Box plot shows that both critics and audience rating variables have strong relationship with audience score. the mean of Certified Fresh

Audience Score vs Critices Rating

$H_0 : \mu_1 = \mu_2 = \mu_3 ... = \mu_k$; H_A : At least one mean is different

```
model <- aov(audience_score ~ critics_rating, movies)
anova(model)
```

```
## Analysis of Variance Table
##
## Response: audience_score
##                 Df Sum Sq Mean Sq F value    Pr(>F)
## critics_rating   2 100347   50174  196.48 < 2.2e-16 ***
## Residuals      648 165473     255
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
result <- 100347/(100347+165473)
result
```

```
## [1] 0.3774998
```

```
# We need to caluclate new p-value for ANOVA
k <- length(unique(movies$critics_rating))
k <- k*(k-1)/2
alpha <- .05
alpha_adj <- alpha/k
alpha_adj
```

```
## [1] 0.01666667
```

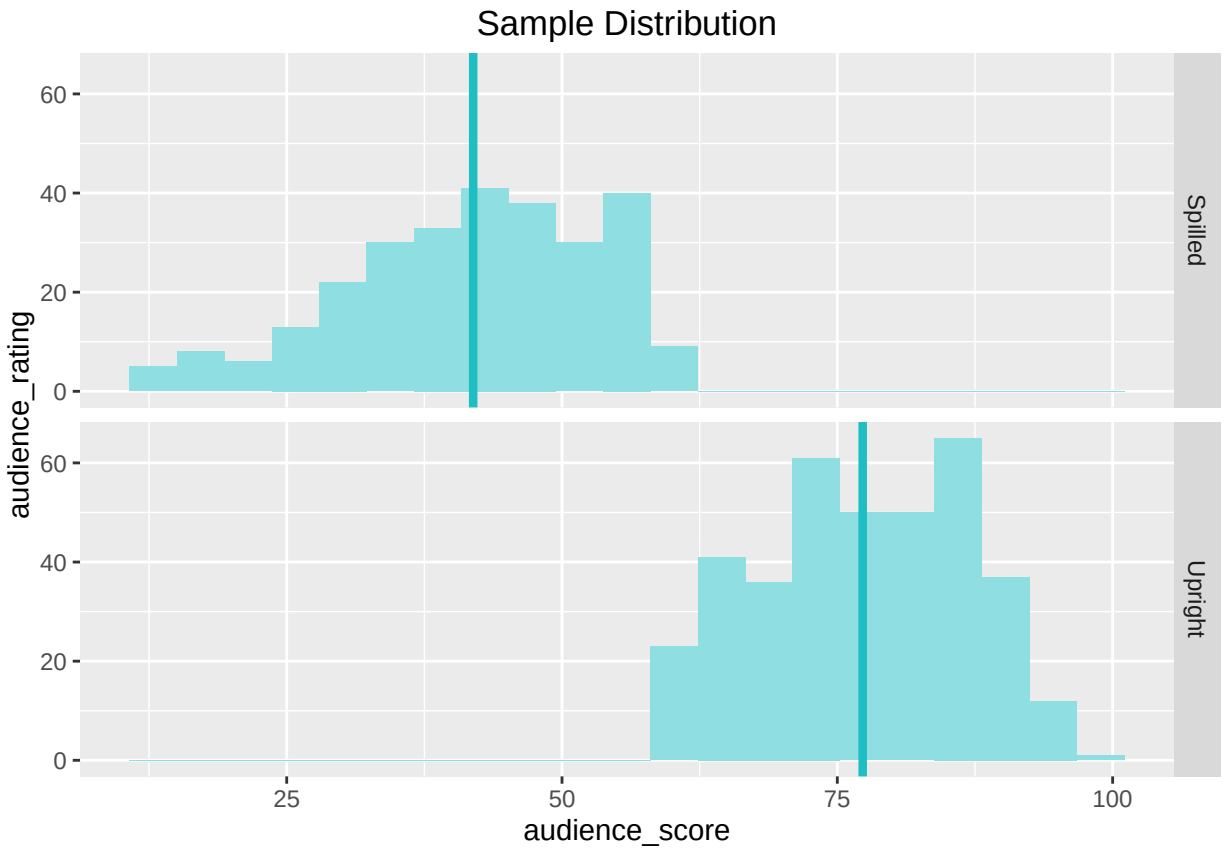
Interpretation:
Over 37% percent of data is explained by this anlysis else are not explained by this variables which indicate that this variables is great explanatory variable.
Since adjust p-value is .017, we reject the null hypothesis because F value is 2.2e-16. Therefore we can conclude that there are at least two group means are significantly different from each other.

Audience Score vs Audience Rating

$H_0 : \mu_{Spilled} = \mu_{Upright}$; $H_A : \mu_{Spilled} \neq \mu_{Upright}$

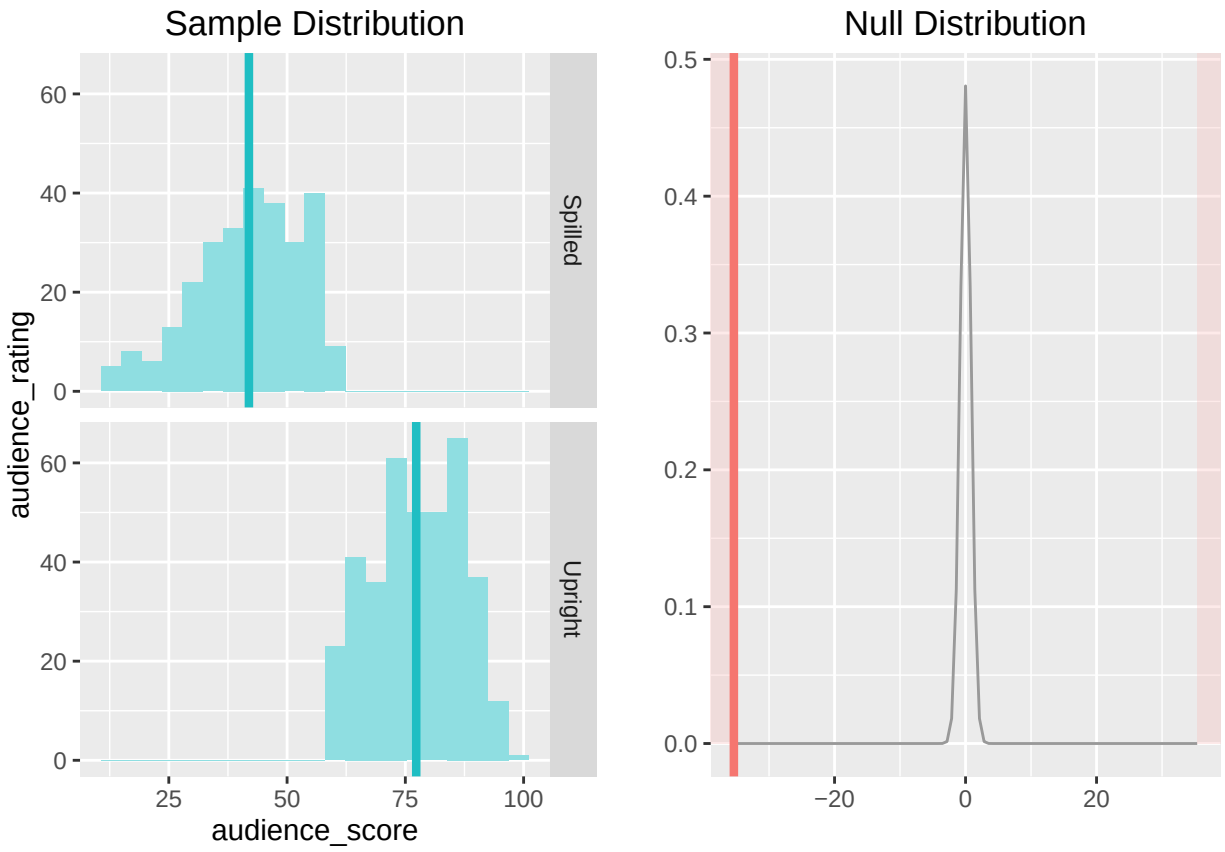
```
# Confidence Interval with function (Mean : Catagorical vs Quantative)
inference(y = audience_score, x = audience_rating, data = movies, statistic = "mean", type = "ci",
          method = "theoretical", order = c("Spilled","Upright"))
```

```
## Response variable: numerical, Explanatory variable: categorical (2 levels)
## n_Spilled = 275, y_bar_Spilled = 41.9345, s_Spilled = 11.217
## n_Upright = 376, y_bar_Upright = 77.3032, s_Upright = 9.3317
## 95% CI (Spilled - Upright): (-37.0029 , -33.7344)
```




```
# Hypothesis Test with functions (Mean : Catagorical vs Quantative)
inference(y = audience_score, x = audience_rating, data = movies, statistic = "mean", type = "ht", null
          alternative = "twosided", method = "theoretical")
```

```
## Response variable: numerical
## Explanatory variable: categorical (2 levels)
## n_Spilled = 275, y_bar_Spilled = 41.9345, s_Spilled = 11.217
## n_Upright = 376, y_bar_Upright = 77.3032, s_Upright = 9.3317
## H0: mu_Spilled =  mu_Upright
## HA: mu_Spilled != mu_Upright
## t = -42.6058, df = 274
## p_value = < 0.0001
```



Interpretation:
The population mean difference of two groups lies btween there two bounds of -37 and -33, since 95% of the time confidence intervals contain the tru means. Since the p-val is less than .05 we will reject the null hypothesis and accept the alternative hypothesis that aduience rating is effecting on aduience score.

```
# Anova
model <- aov(audience_score ~ audience_rating, movies)
anova(model)
```

```
## Analysis of Variance Table
##
## Response: audience_score
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## audience_rating    1 198690  198690  1920.9 < 2.2e-16 ***
## Residuals        649  67130     103
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
result <- 198690/(198690+67130)
result
```

```
## [1] 0.7474607
```

```
# We need to caluclate new p-value for ANOVA
k <- length(unique(movies$critics_rating))
k <- k*(k-1)/2
alpha <- .05
alpha_adj <- alpha/k
```

```
alpha_adj
```

```
## [1] 0.01666667
```

Interpretation:
Over 74% percent of data is explained by this anlysis else are not explained by this variables which indicate that this variables is great explanatory variable.
Since adjust p-value is .017, we reject the null hypothesis because F value is 2.2e-16. Therefore we can conclude that there are at least two group means are significantly different from each other.

Part 4: Modeling

```
# Build base model
mdl_lm0 <- lm(audience_score ~ ., movies)
sum_lm0 <- summary(mdl_lm0)
sum_lm0
```

```
##
## Call:
## lm(formula = audience_score ~ ., data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.8923  -4.4001   0.2849   4.2137  24.1463
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     3.283e+02  1.306e+02   2.514   0.0122 *
## genreAnimation                  3.825e+00  2.757e+00   1.387   0.1658
## genreArt House & International -1.491e+00  2.131e+00  -0.700   0.4842
## genreComedy                     1.794e+00  1.164e+00   1.541   0.1239
## genreDocumentary                1.642e+00  1.721e+00   0.954   0.3403
## genreDrama                     -1.437e-02  1.041e+00  -0.014   0.9890
## genreHorror                    -1.617e+00  1.732e+00  -0.934   0.3509
## genreMusical & Performing Arts  3.968e+00  2.264e+00   1.753   0.0801 .
## genreMystery & Suspense        -2.598e+00  1.306e+00  -1.990   0.0471 *
## genreOther                     -4.055e-01  1.971e+00  -0.206   0.8371
## genreScience Fiction & Fantasy -4.439e-01  2.461e+00  -0.180   0.8569
## mpaa_ratingNC-17               -1.156e+00  5.219e+00  -0.221   0.8248
## mpaa_ratingPG                   1.575e-01  1.903e+00   0.083   0.9341
## mpaa_ratingPG-13               -5.170e-01  2.022e+00  -0.256   0.7982
## mpaa_ratingR                   -7.419e-01  1.937e+00  -0.383   0.7019
## mpaa_ratingUnrated              7.240e-01  2.261e+00   0.320   0.7489
## critics_ratingFresh            -2.318e-01  9.075e-01  -0.255   0.7985
## critics_ratingRotten           -1.018e+00  1.413e+00  -0.720   0.4717
## audience_ratingUpright          1.964e+01  8.014e-01  24.503   <2e-16 ***
## best_pic_nomyes                 4.078e+00  1.822e+00   2.238   0.0256 *
## best_pic_winyes                -3.085e+00  3.191e+00  -0.967   0.3340
## best_actor_winyes              -1.027e-01  8.220e-01  -0.125   0.9006
## best_actress_winyes            -1.473e+00  9.067e-01  -1.624   0.1049
## best_dir_winyes                 6.963e-02  1.195e+00   0.058   0.9535
## top200_boxyes                  -8.892e-01  1.939e+00  -0.459   0.6467
```

```
## popularitymid                 -1.322e+00  1.120e+00  -1.180   0.2385
## popularityhigh                -5.017e-01  8.916e-01  -0.563   0.5738
## runtime                       -2.451e-02  1.757e-02  -1.395   0.1635
## thtr_rel_year                 -1.663e-01  6.495e-02  -2.560   0.0107 *
## thtr_rel_month                -1.735e-01  7.982e-02  -2.174   0.0301 *
## imdb_rating                    9.299e+00  4.917e-01  18.911   <2e-16 ***
## imdb_num_votes                 3.992e-06  3.765e-06   1.060   0.2894
## critics_score                  6.392e-03  2.522e-02   0.253   0.8000
## date_dff                      -5.571e-04  2.512e-04  -2.217   0.0270 *
## mu_all                         2.248e-04  2.042e-04   1.101   0.2713
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.835 on 616 degrees of freedom
## Multiple R-squared:  0.8918, Adjusted R-squared:  0.8858
## F-statistic: 149.3 on 34 and 616 DF,  p-value: < 2.2e-16
```

```
sum_lm0$adj.r.squared
```

```
## [1] 0.8857781
```

Wow we already have close to 1 R squared, this base model is already good. However, we need to explor a bit more and see if we can improve this model.
First I'm going remove genre since it has such high p-value.

```
mdl_lm1 <- lm(audience_score ~ . -genre, movies)
sum_lm1 <- summary(mdl_lm1)
sum_lm1
```

```
##
## Call:
## lm(formula = audience_score ~ . - genre, data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.2442  -4.5552   0.2674   4.3063  25.3521
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.419e+02  1.302e+02   2.625  0.00886 **
## mpaa_ratingNC-17        -2.131e+00  5.182e+00  -0.411  0.68102
## mpaa_ratingPG           -6.304e-01  1.731e+00  -0.364  0.71585
## mpaa_ratingPG-13        -1.593e+00  1.787e+00  -0.891  0.37303
## mpaa_ratingR            -2.372e+00  1.686e+00  -1.407  0.15993
## mpaa_ratingUnrated      -2.704e-01  2.019e+00  -0.134  0.89347
## critics_ratingFresh     -4.340e-01  9.094e-01  -0.477  0.63333
## critics_ratingRotten    -1.064e+00  1.419e+00  -0.750  0.45374
## audience_ratingUpright   2.014e+01  7.899e-01  25.501  < 2e-16 ***
## best_pic_nomyes          3.960e+00  1.830e+00   2.165  0.03080 *
## best_pic_winyes         -2.669e+00  3.200e+00  -0.834  0.40462
## best_actor_winyes       -3.050e-01  8.227e-01  -0.371  0.71098
## best_actress_winyes     -1.603e+00  9.026e-01  -1.776  0.07628 .
## best_dir_winyes          1.316e-01  1.204e+00   0.109  0.91300
## top200_boxyes           -1.146e+00  1.943e+00  -0.590  0.55563
## popularitymid           -5.934e-01  1.058e+00  -0.561  0.57523
## popularityhigh          -1.617e-01  8.806e-01  -0.184  0.85439
```

```
## runtime                -3.042e-02  1.706e-02  -1.783  0.07508 .
## thtr_rel_year          -1.721e-01  6.478e-02  -2.656  0.00810 **
## thtr_rel_month         -1.382e-01  7.993e-02  -1.728  0.08439 .
## imdb_rating             9.205e+00  4.725e-01  19.484  < 2e-16 ***
## imdb_num_votes          3.761e-06  3.729e-06   1.009  0.31356
## critics_score           9.724e-03  2.524e-02   0.385  0.70020
## date_dff               -5.662e-04  2.517e-04  -2.249  0.02485 *
## mu_all                  1.101e-04  2.010e-04   0.548  0.58419
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.895 on 626 degrees of freedom
## Multiple R-squared:  0.888,  Adjusted R-squared:  0.8837
## F-statistic: 206.9 on 24 and 626 DF,  p-value: < 2.2e-16
```

```
sum_lm1$adj.r.squared
```

```
## [1] 0.8837397
```

By removing genre, R squared reduced, so we are going to keep ganre in the model model.
And now I'm going to remove award variables with same reason.

```
mdl_lm3 <- lm(audience_score ~ . -best_pic_nom -best_pic_win -best_actor_win -best_actress_win -best_di
sum_lm3 <- summary(mdl_lm3)
sum_lm3
```

```
##
## Call:
## lm(formula = audience_score ~ . - best_pic_nom - best_pic_win -
##     best_actor_win - best_actress_win - best_dir_win, data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.7176  -4.6261   0.4347   4.1006  24.3021
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     3.103e+02  1.300e+02   2.388   0.0173 *
## genreAnimation                  3.607e+00  2.754e+00   1.310   0.1908
## genreArt House & International -1.640e+00  2.129e+00  -0.770   0.4416
## genreComedy                     1.655e+00  1.158e+00   1.429   0.1534
## genreDocumentary                1.533e+00  1.719e+00   0.892   0.3728
## genreDrama                     -1.403e-01  1.029e+00  -0.136   0.8916
## genreHorror                    -1.602e+00  1.733e+00  -0.925   0.3555
## genreMusical & Performing Arts  3.865e+00  2.267e+00   1.705   0.0887 .
## genreMystery & Suspense        -2.800e+00  1.292e+00  -2.168   0.0305 *
## genreOther                     -1.673e-01  1.962e+00  -0.085   0.9321
## genreScience Fiction & Fantasy -4.422e-01  2.464e+00  -0.179   0.8576
## mpaa_ratingNC-17               -1.147e+00  5.213e+00  -0.220   0.8259
## mpaa_ratingPG                   1.600e-01  1.903e+00   0.084   0.9330
## mpaa_ratingPG-13               -4.459e-01  2.023e+00  -0.220   0.8256
## mpaa_ratingR                   -7.094e-01  1.939e+00  -0.366   0.7145
## mpaa_ratingUnrated              7.041e-01  2.264e+00   0.311   0.7559
## critics_ratingFresh            -3.262e-01  8.975e-01  -0.363   0.7164
## critics_ratingRotten           -1.147e+00  1.413e+00  -0.812   0.4169
## audience_ratingUpright          1.972e+01  7.987e-01  24.685   <2e-16 ***
```

```
## top200_boxyes                  -1.176e+00  1.936e+00  -0.607   0.5439
## popularitymid                  -1.357e+00  1.119e+00  -1.212   0.2260
## popularityhigh                 -5.349e-01  8.913e-01  -0.600   0.5487
## runtime                        -2.546e-02  1.692e-02  -1.504   0.1330
## thtr_rel_year                  -1.574e-01  6.461e-02  -2.436   0.0151 *
## thtr_rel_month                 -1.506e-01  7.926e-02  -1.900   0.0579 .
## imdb_rating                     9.323e+00  4.920e-01  18.947   <2e-16 ***
## imdb_num_votes                  4.375e-06  3.666e-06   1.193   0.2332
## critics_score                   6.317e-03  2.524e-02   0.250   0.8024
## date_dff                       -5.267e-04  2.498e-04  -2.108   0.0354 *
## mu_all                          2.358e-04  2.027e-04   1.163   0.2452
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.846 on 621 degrees of freedom
## Multiple R-squared:  0.8905, Adjusted R-squared:  0.8854
## F-statistic: 174.2 on 29 and 621 DF,  p-value: < 2.2e-16
```

```
sum_lm3$adj.r.squared
```

```
## [1] 0.8853918
```

By removing award variables, we have a bit higher R squared which is good. Now I'm going to remove, mpaa_rating variables.

```
mdl_lm4 <- lm(audience_score ~ . -best_pic_nom -best_pic_win -best_actor_win -best_actress_win -best_di
sum_lm4 <- summary(mdl_lm4)
sum_lm4
```

```
##
## Call:
## lm(formula = audience_score ~ . - best_pic_nom - best_pic_win -
##     best_actor_win - best_actress_win - best_dir_win - mpaa_rating,
##     data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.9776  -4.7754   0.4579   4.1864  24.0204
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.855e+02  1.275e+02   2.239   0.0255 *
## genreAnimation                  3.967e+00  2.482e+00   1.599   0.1104
## genreArt House & International -1.637e+00  2.096e+00  -0.781   0.4350
## genreComedy                     1.584e+00  1.146e+00   1.382   0.1675
## genreDocumentary                2.020e+00  1.634e+00   1.236   0.2168
## genreDrama                     -3.503e-01  1.009e+00  -0.347   0.7285
## genreHorror                    -1.850e+00  1.685e+00  -1.098   0.2727
## genreMusical & Performing Arts  3.835e+00  2.257e+00   1.699   0.0898 .
## genreMystery & Suspense        -3.095e+00  1.264e+00  -2.448   0.0146 *
## genreOther                     -1.386e-01  1.949e+00  -0.071   0.9433
## genreScience Fiction & Fantasy -4.986e-01  2.455e+00  -0.203   0.8391
## critics_ratingFresh            -2.925e-01  8.900e-01  -0.329   0.7426
## critics_ratingRotten           -1.018e+00  1.404e+00  -0.725   0.4686
## audience_ratingUpright          1.975e+01  7.939e-01  24.884   <2e-16 ***
## top200_boxyes                  -9.152e-01  1.913e+00  -0.478   0.6325
```

```
## popularitymid                -1.295e+00  1.099e+00  -1.178   0.2391
## popularityhigh               -4.912e-01  8.835e-01  -0.556   0.5784
## runtime                      -2.455e-02  1.646e-02  -1.492   0.1363
## thtr_rel_year                -1.453e-01  6.338e-02  -2.293   0.0222 *
## thtr_rel_month               -1.499e-01  7.865e-02  -1.905   0.0572 .
## imdb_rating                   9.306e+00  4.896e-01  19.006   <2e-16 ***
## imdb_num_votes                3.965e-06  3.642e-06   1.089   0.2768
## critics_score                 9.653e-03  2.502e-02   0.386   0.6998
## date_dff                     -4.926e-04  2.472e-04  -1.992   0.0468 *
## mu_all                        2.575e-04  1.932e-04   1.333   0.1831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.83 on 626 degrees of freedom
## Multiple R-squared:  0.8902, Adjusted R-squared:  0.8859
## F-statistic: 211.4 on 24 and 626 DF,  p-value: < 2.2e-16
```

```
sum_lm4$adj.r.squared
```

```
## [1] 0.8859477
```

Again we gained a bit more R squared value. We will continue this process until we do not have any more R squared gain.

```
mdl_lm5 <- lm(audience_score ~ . -best_pic_nom -best_pic_win -best_actor_win
              -best_actress_win -best_dir_win -mpaa_rating -top200_box -critics_score, movies)
sum_lm5 <- summary(mdl_lm5)
sum_lm5$adj.r.squared
```

```
## [1] 0.8862442
```

```
mdl_lm6 <- lm(audience_score ~ . -best_pic_nom -best_pic_win -best_actor_win
              -best_actress_win -best_dir_win -mpaa_rating -top200_box -critics_score -popularity, movi
sum_lm6 <- summary(mdl_lm6)
sum_lm6$adj.r.squared
```

```
## [1] 0.8863315
```

Okay we finally found the right combination of predictors with has R squared of .8863 I'm gonig to add the cluster variables to see if it improves a bit

```
setwd("F:/specialization/22-Master Statistics with R (Duke University)/data")
movies <- readRDS("movies04_yesCluster.rds")
mdl_lm7 <- lm(audience_score ~ ., movies)
sum_lm7 <- summary(mdl_lm7)
sum_lm7$adj.r.squared
```

```
## [1] 0.886351
```

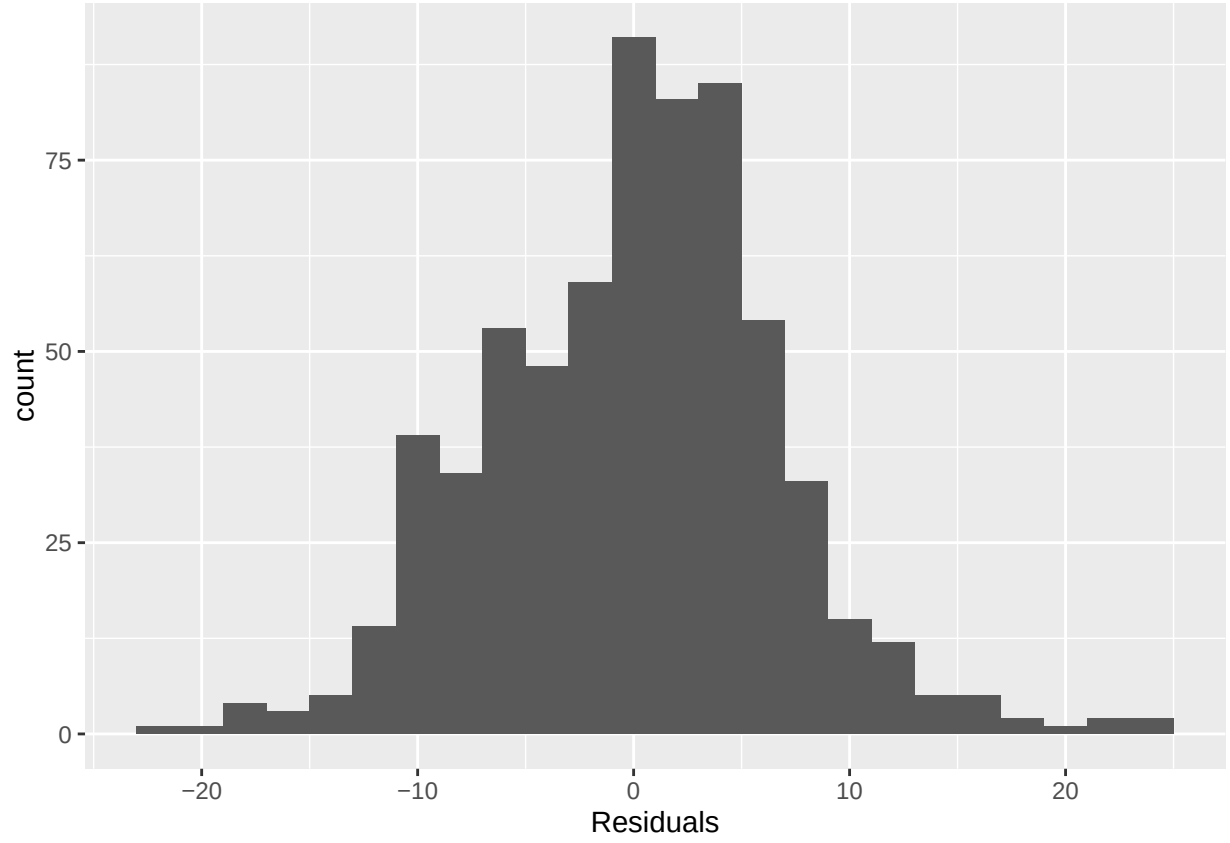
Okay, so cluster variables does not do much, we are going back to previous dataset.

Model Dignostic

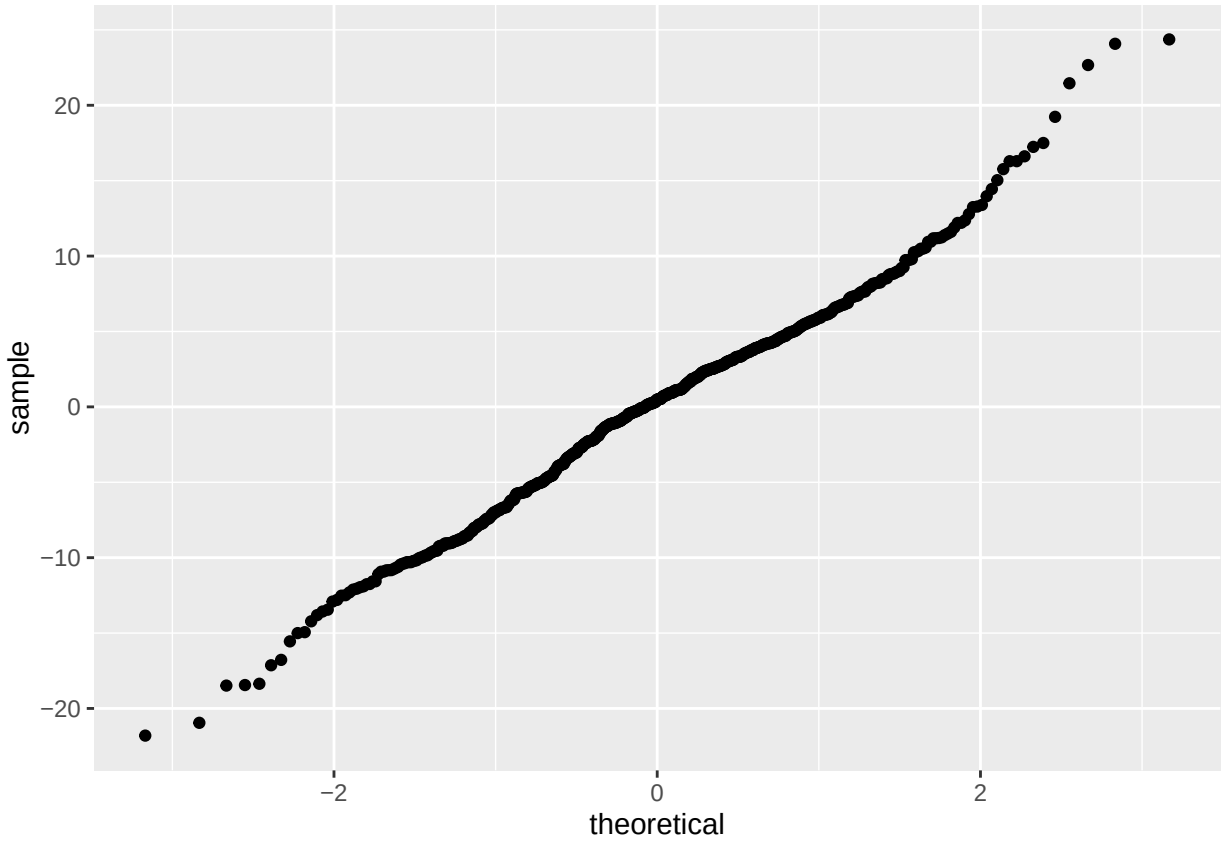
Nearly Normal Residuals with mean 0

```
# Nearly normal residuals: Residuals are right skewed, but the sample size is large, so this may not be
setwd("F:/specialization/22-Master Statistics with R (Duke University)/data")
movies <- readRDS("movies02.rds")
```

```
mdl_lm6 <- lm(audience_score ~ ., movies)

ggplot(mdl_lm6, aes(.resid)) +
     geom_histogram(binwidth = 2) +
     xlab("Residuals")
```

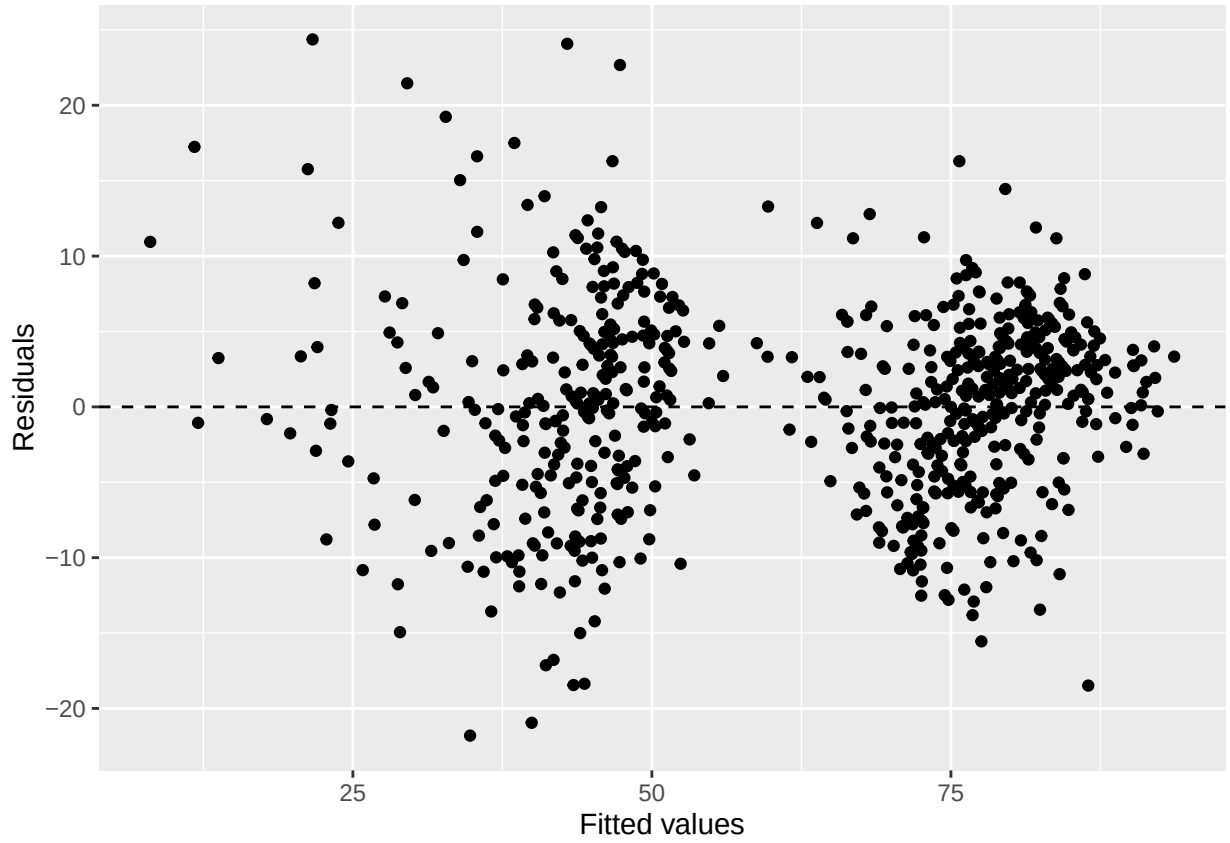



```
ggplot(mdl_lm6, aes(sample = .resid)) +
     stat_qq()
```



Constan Variability of Residuals

```
# Linear association: The residuals plot shows a random scatter.
# Constant variance of residuals: No fan shape in residuals plot.
ggplot(mdl_lm6, aes(.fitted, .resid)) +
     geom_point() +
     geom_hline(yintercept = 0, linetype = "dashed") +
     xlab("Fitted values") +
     ylab("Residuals")
```



The model passes all the diagnostic tests.

Correlation plot

```
library(GGally)
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

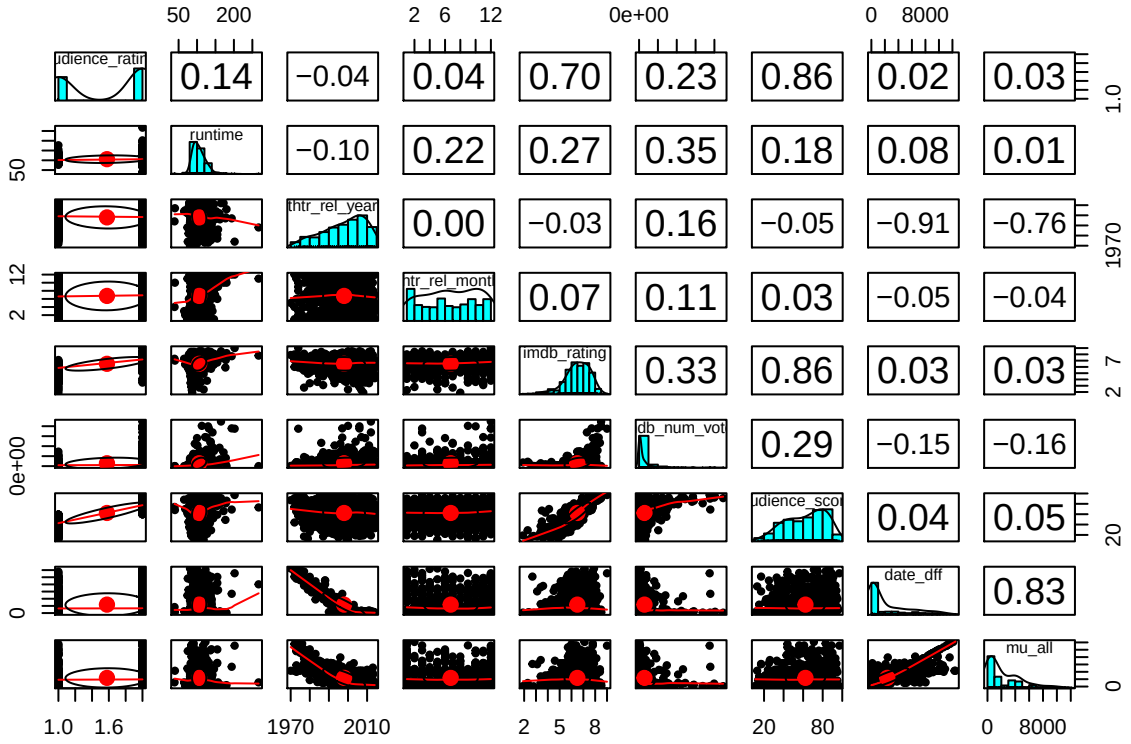
```
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.2.5
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
pairs.panels(movies[,3:ncol(movies)])
```



Correlations between variables seems to be okay.

Part 5: Prediction

```
setwd("F:/specialization/22-Master Statistics with R (Duke University)/data")
movies <- readRDS("movies02.rds")
str(movies)
```

```
## 'data.frame':    651 obs. of  11 variables:
##  $ genre          : Factor w/ 11 levels "Action & Adventure",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ critics_rating : Factor w/ 3 levels "Certified Fresh",..: 1 2 2 3 3 1 2 2 3 3 ...
##  $ audience_rating: Factor w/ 2 levels "Spilled","Upright": 2 1 2 1 2 2 1 1 1 1 ...
##  $ runtime        : num  90 113 118 103 83 120 98 138 99 83 ...
##  $ thtr_rel_year  : num  1992 1977 1979 1996 1973 ...
##  $ thtr_rel_month : num  11 1 10 11 11 6 7 6 1 5 ...
##  $ imdb_rating    : num  8 6.3 7.4 5.6 7.6 7 6 5.7 4.2 3.5 ...
##  $ imdb_num_votes : int  246907 4687 8544 70209 78862 201787 109633 204042 43268 10055 ...
##  $ audience_score : num  92 55 83 40 81 71 49 29 33 30 ...
##  $ date_dff       : num  4346 8592 6538 508 9735 ...
##  $ mu_all         : num  4346 8592 6538 508 9735 ...
```

I'm going to exclude few movies before I build the model so that i can predict the score for that movie which is not included in the sample.

```
rand <- sample(nrow(movies), 5)
test <- movies[rand, ]
train <- movies[-rand, ]

mdl_lm <- lm(audience_score ~ ., train)
summary(mdl_lm)
```

```
##
## Call:
## lm(formula = audience_score ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.7552  -4.6481   0.3952   4.1831  24.3581
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.791e+02  1.278e+02   2.185   0.0293 *
## genreAnimation                  3.869e+00  2.472e+00   1.565   0.1180
## genreArt House & International -2.000e+00  2.060e+00  -0.971   0.3318
## genreComedy                     1.582e+00  1.137e+00   1.391   0.1647
## genreDocumentary                1.432e+00  1.495e+00   0.957   0.3387
## genreDrama                     -4.980e-01  9.866e-01  -0.505   0.6139
## genreHorror                    -1.778e+00  1.682e+00  -1.057   0.2909
## genreMusical & Performing Arts  3.548e+00  2.226e+00   1.594   0.1115
## genreMystery & Suspense        -3.068e+00  1.260e+00  -2.435   0.0152 *
## genreOther                     -1.895e-01  1.941e+00  -0.098   0.9223
## genreScience Fiction & Fantasy -4.654e-01  2.453e+00  -0.190   0.8496
## critics_ratingFresh            -3.480e-01  8.628e-01  -0.403   0.6869
## critics_ratingRotten           -1.398e+00  9.281e-01  -1.506   0.1325
## audience_ratingUpright          1.977e+01  7.918e-01  24.961   <2e-16 ***
## runtime                        -2.347e-02  1.641e-02  -1.431   0.1530
## thtr_rel_year                  -1.426e-01  6.352e-02  -2.245   0.0251 *
## thtr_rel_month                 -1.368e-01  7.866e-02  -1.739   0.0826 .
## imdb_rating                     9.467e+00  4.275e-01  22.147   <2e-16 ***
## imdb_num_votes                  4.787e-06  3.064e-06   1.562   0.1187
## date_dff                       -4.961e-04  2.474e-04  -2.006   0.0453 *
## mu_all                          2.158e-04  1.896e-04   1.139   0.2553
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.834 on 625 degrees of freedom
## Multiple R-squared:  0.8894, Adjusted R-squared:  0.8859
## F-statistic: 251.3 on 20 and 625 DF,  p-value: < 2.2e-16
```

```
y_hat <- round(predict(mdl_lm, test), 0)
y <- test$audience_score
total_r_sq <- sum((y - y_hat)^2)

data.frame("Actual y" = y, "Predicted y" = y_hat)
```

```
##     Actual.y Predicted.y
```

```
## 321       86          81
## 206       74          80
## 303       89          84
## 60        63          59
## 383       84          81
```

```
print(paste0("R Squared : ", total_r_sq))
```

```
## [1] "R Squared : 111"
```

Part 6: Conclusion

I've created new variables called date_dff which is the difference between the thether release date and dvd release date.
I noticed the popular movies tend to wait longer till DVD release wheres failure movies comes out DVD faster. Oviously this variables contributed well in the model. Also I've noticed that variables with nearly zero variance have all most no effects on the model. Also unbalanced categorical variables does not do much in the model for example, 5 yes and 700 no.
Linear model is very simple yet powerful than I thought and it does great job to picking the importatn variables.