PS 6

Ilayda Koca

2022-10-17

Getting Set Up

If you haven't already, create a folder for this course, and then a subfolder within for the second lecture `Topic8_Regression`, and two additional subfolders within `code` and `data`.

Open `RStudio` and create a new RMarkDown file (`.Rmd`) by going to `File -> New File -> R Markdown...`. Change the title to `"DS1000: Problem Set 6"` and the author to your full name. Save this file as `[LAST NAME]_ps6.Rmd` to your `code` folder.

If you haven't already, download the `mv.Rds` file from the course github page (https://github.com/jbisbee1/DS1000-F2022/blob/master/Lectures/Topic8_Regression/data/mv.Rds) and save it to your `data` folder.

The contents of this problem set can be found in the following resources:

- Topic 6 (https://github.com/jbisbee1/DS1000-F2022/tree/master/Lectures/Topic6_UnivariateVisualization) parts 1 & 2 (univariate visualization)
- Topic 7 (https://github.com/jbisbee1/DS1000-F2022/tree/master/Lectures/Topic7_ConditionalVariation) parts 1 & 2 (conditional analysis)
- Topic 8 (https://github.com/jbisbee1/DS1000-F2022/tree/master/Lectures/Topic8_Regression) parts 1 & 2 (regression, RMSE, and cross validation)

Require `tidyverse` and load the `mv.Rds` data to `mv`.

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.4
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.0      ✔ stringr 1.4.1
## ✔ readr   2.1.2      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
mv <- readRDS('../data/mv.Rds')
```

NOTE: Starting with this problem set, we are moving to a system with fewer questions that are each more involved. For EACH of the questions below, you are expected to:

1. Perform uni-variate analysis on each of the two variables separately (determine their class, identifying missing-nes, and plot with the appropriate figure).
2. Plot the conditional relationship between the two variables using the appropriate figure. Transform the variables if needed by logging highly skewed data.

3. Estimate the linear regression model.
4. Evaluate model fit with 1) visual analysis of the residuals and 2) cross validation.

Question 1 [10 points]

- **Research Question:** Are longer movies more expensive to make?
- **Theory:** Longer running movies take more time to make. Time is money.
- **Hypothesis:** The longer the movie, the more expensive it is.
- *Hints:* You are looking at the conditional relationship between `budget` and `runtime`.

```
#look at the data to find missing data.
mv_sum <- mv %>%
  select(budget, runtime) %>%
  summary()

mv_sum
```

```
##      budget              runtime     
##  Min.   :     5172   Min.   : 55.0  
##  1st Qu.: 16865322   1st Qu.: 95.0  
##  Median : 37212044   Median :104.0  
##  Mean   : 57420173   Mean   :107.3  
##  3rd Qu.: 77844746   3rd Qu.:116.0  
##  Max.   :387367903   Max.   :366.0  
##  NA's   :4482        NA's   :4
```

```
#1. Perform uni-variate analysis on each of the two variables separately (determine their class, identifying missing-nes, and plot with the appropriate figure).
g0 <- mv %>%
  select(budget,runtime) %>%
  glimpse()
```

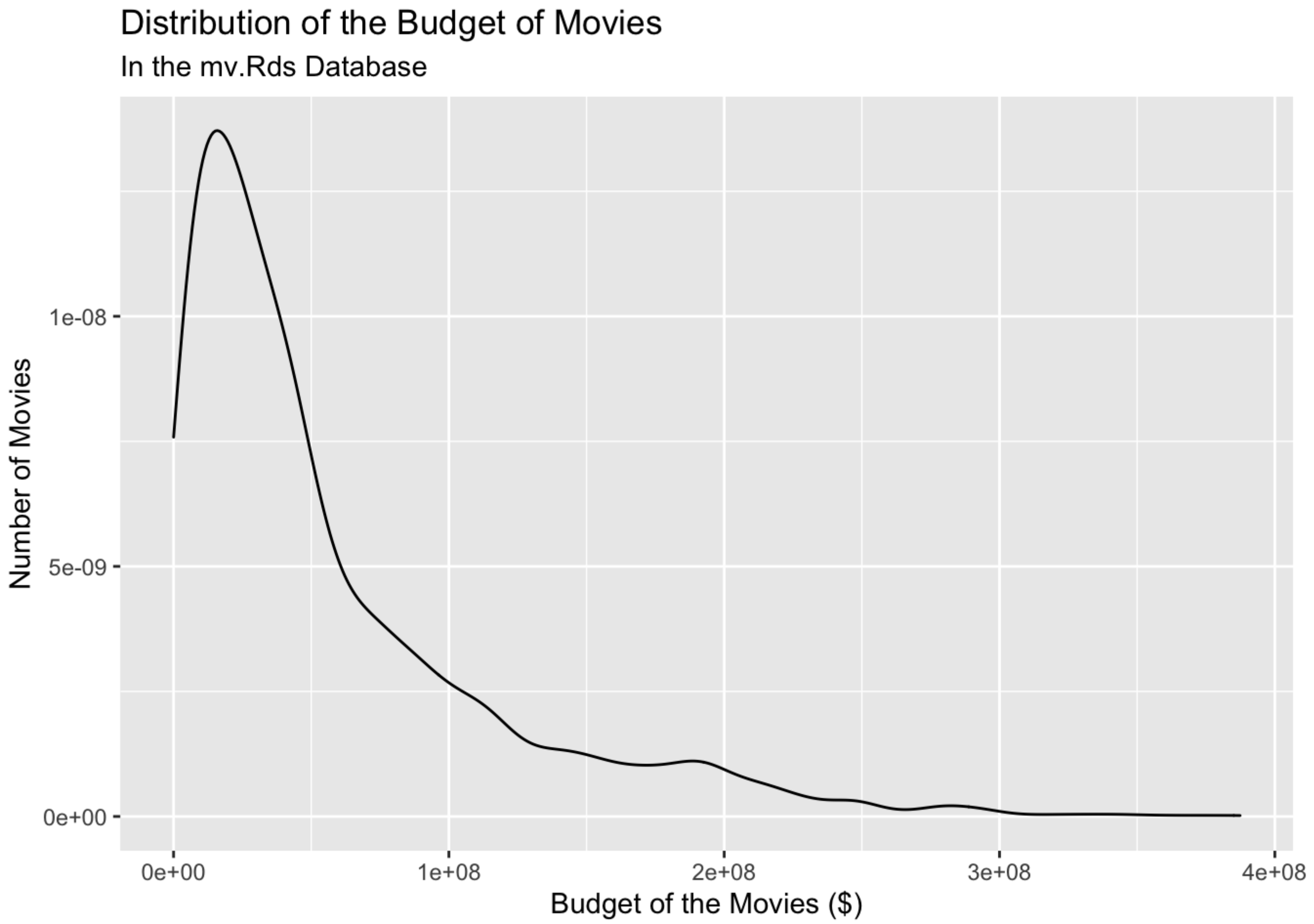
```
## Rows: 7,673
## Columns: 2
## $ budget  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
## $ runtime <dbl> 146, 104, 124, 88, 98, 95, 133, 129, 127, 100, 116, 109, 114, …
```

```
g0
```

```
## # A tibble: 7,673 × 2
##    budget runtime
##     <dbl>   <dbl>
##  1     NA     146
##  2     NA     104
##  3     NA     124
##  4     NA      88
##  5     NA      98
##  6     NA      95
##  7     NA     133
##  8     NA     129
##  9     NA     127
## 10     NA     100
## # … with 7,663 more rows
```

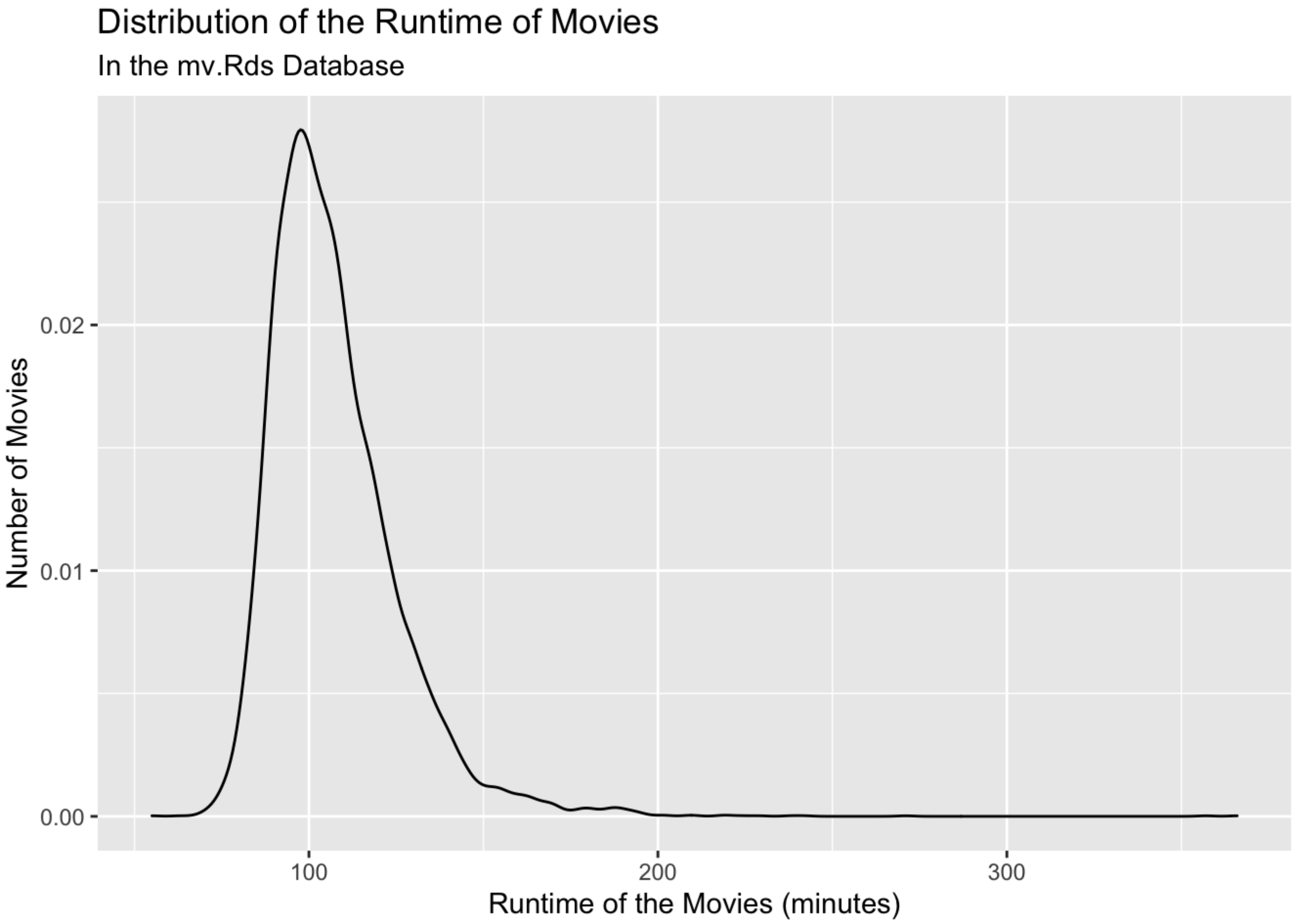
```
g1 <- mv %>%
  ggplot(aes(x = budget, na.rm=T)) +
  geom_density() +
  labs(title = "Distribution of the Budget of Movies",
       subtitle = "In the mv.Rds Database",
       y = "Number of Movies",
       x = "Budget of the Movies ($)")

g1
```

```
## Warning: Removed 4482 rows containing non-finite values (stat_density).
```



```
g2 <- mv %>%
  ggplot(aes(x = runtime, na.rm=T)) +
  geom_density() +
  labs(title = "Distribution of the Runtime of Movies",
       subtitle = "In the mv.Rds Database",
       y = "Number of Movies",
       x = "Runtime of the Movies (minutes)")

g2
```

```
## Warning: Removed 4 rows containing non-finite values (stat_density).
```

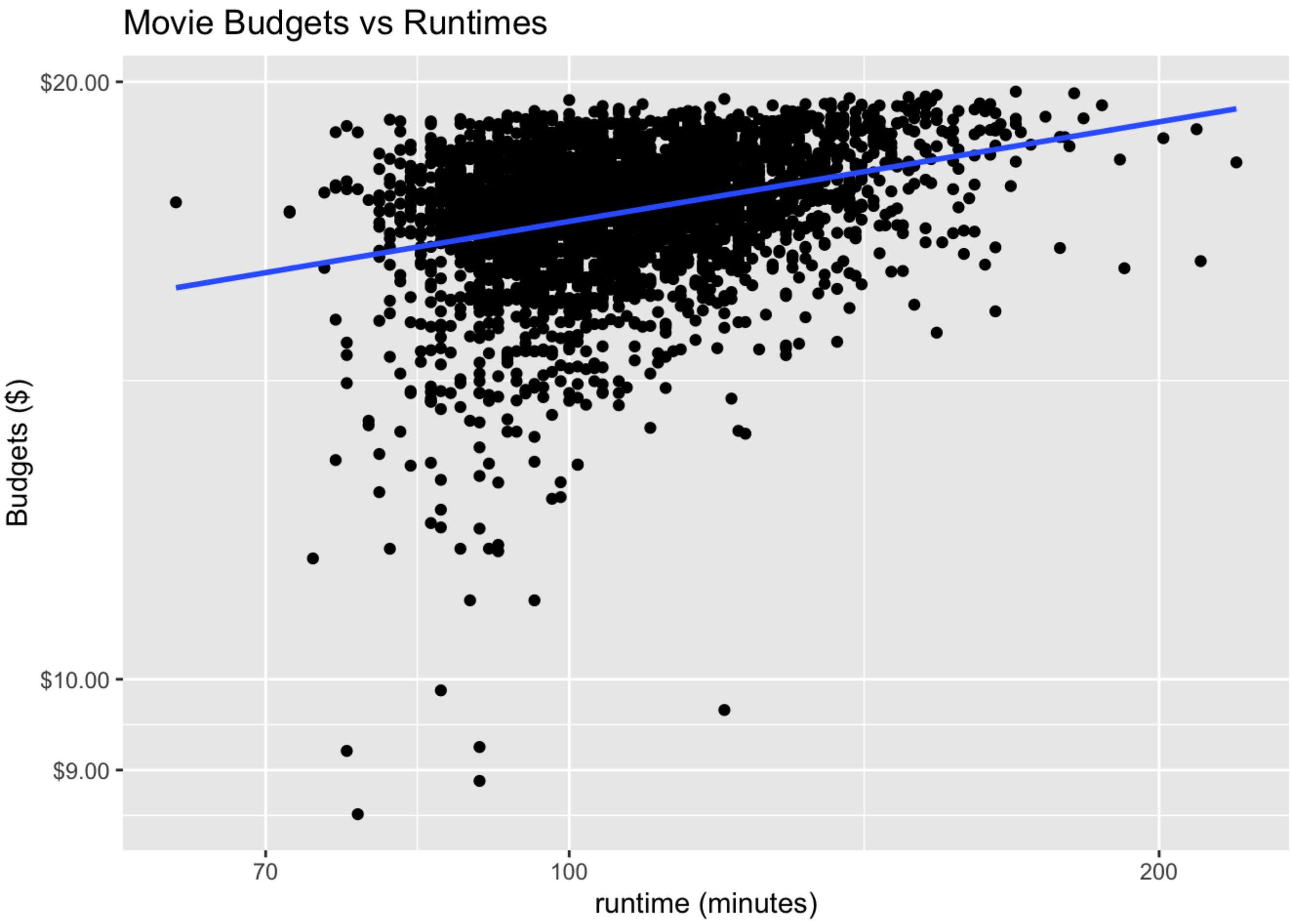


```
# 2. Plot the conditional relationship between the two variables using the appropriat
e figure. Transform the variables if needed by logging highly skewed data.
g3 <- mv %>%
  select(budget,runtime) %>%
  filter(complete.cases(.)) %>%
  glimpse()
```

```
## Rows: 3,189
## Columns: 2
## $ budget  <dbl> 93289619, 10883789, 160147179, 6996721, 13993443, 139934429, 2…
## $ runtime <dbl> 122, 101, 155, 102, 113, 143, 88, 130, 100, 104, 130, 165, 131…
```

```
g5 <- g3 %>%
  ggplot() +
  geom_density(aes(x = log(budget)), color = 'forestgreen',lwd=2) +
  geom_density(aes(x = runtime), color = 'tomato',lwd=2)

g6 <- g3 %>%
  ggplot(aes(x = runtime, y = log(budget))) +
  geom_point() +
  scale_y_log10(labels = scales::dollar) +
  scale_x_log10() +
  geom_smooth(method = 'lm', se = F) +
  labs(title = "Movie Budgets vs Runtimes",
       y = "Budgets ($)",
       x = "runtime (minutes)")

g6
```

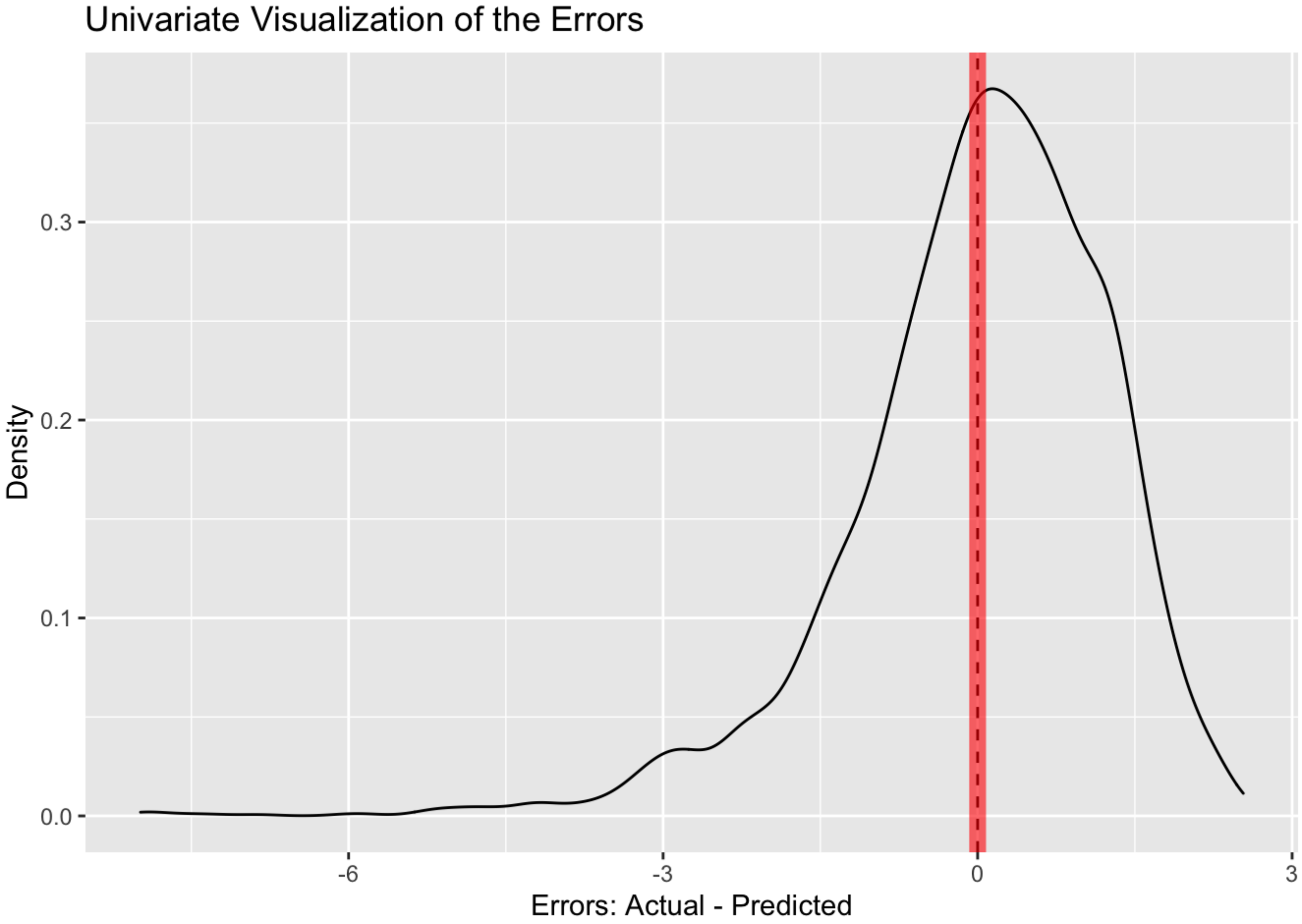
```
## `geom_smooth()` using formula 'y ~ x'
```



```
# 3. Estimate the linear regression model.
#clear the data set of NA values
q1 <- mv %>%
  select(runtime,budget) %>%
  drop_na(runtime,budget)

m1 <- lm(log(budget) ~ runtime, q1)

q1 <- q1 %>%
  mutate(predicted_budget = predict(m1)) %>%
  mutate(errors = log(budget) - predicted_budget)

#RMSE
# E: errors
e1 <- log(q1$budget) - q1$predicted_budget
# SE: squared
se1 <- e1^2
# MSE: mean
mse1 <- mean(se1, na.rm = T)
# RMSE: root
(rmse1 <- sqrt(mse1))
```

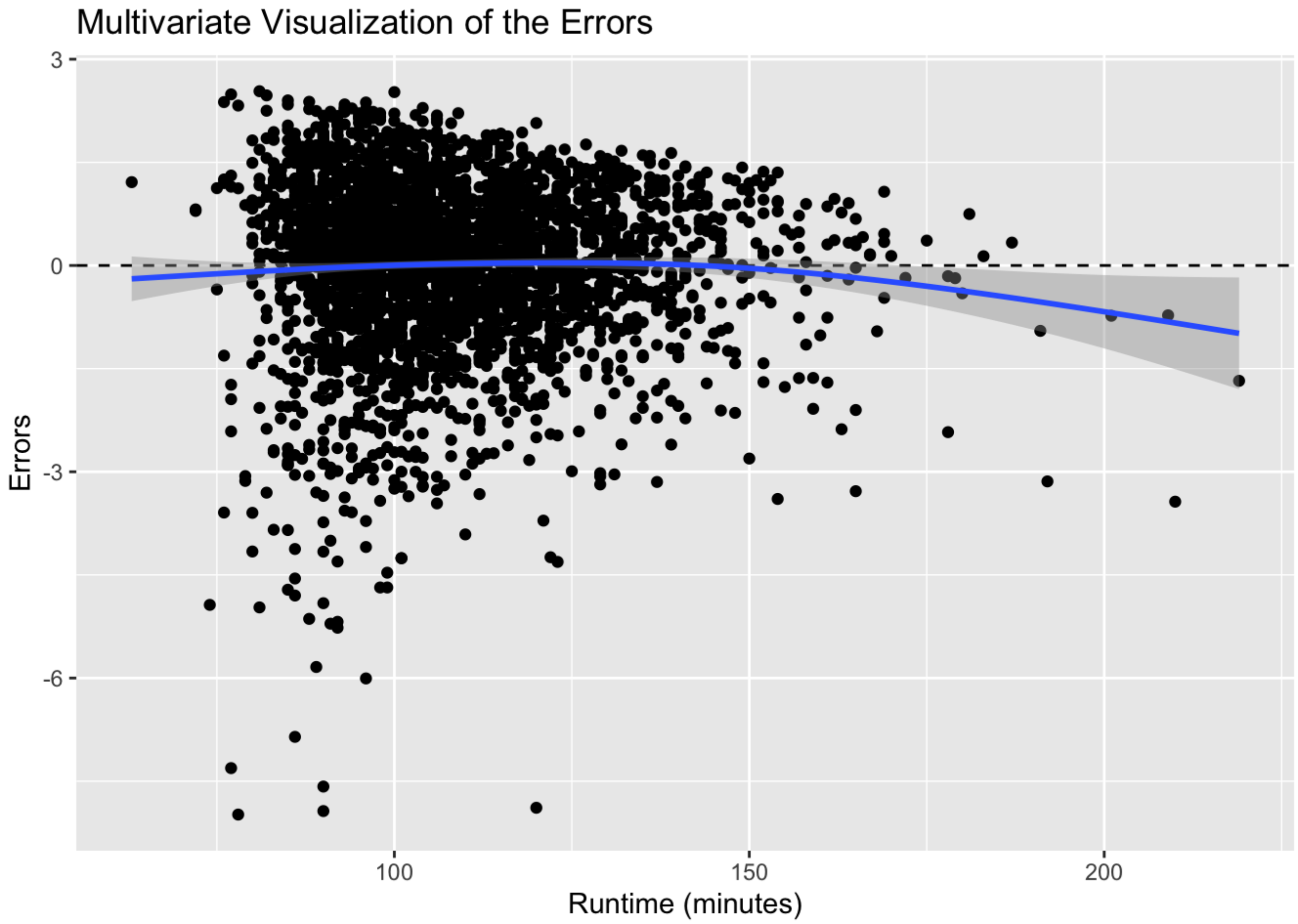
```
## [1] 1.24021
```

```
# 4. Evaluate model fit with 1) visual analysis of the residuals
mvAnalysis <- mv %>%
  select(runtime, budget) %>%
  drop_na()

m3 <- lm(log(budget) ~ runtime,mvAnalysis)

pred_vals <- predict(m3)

errors <- resid(m3)

mvAnalysis$pred_vals <- pred_vals

mvAnalysis$errors <- log(mvAnalysis$budget) - mvAnalysis$pred_vals

mvAnalysis <- mvAnalysis %>%
  mutate(errors = log(budget) - pred_vals)

# Uni-variate visualization of the errors
mvAnalysis %>%
  ggplot(aes(x = errors)) +
  geom_density() +
  geom_vline(xintercept = 0,linetype = 'dashed') +
  geom_vline(xintercept = mean(mvAnalysis$errors),
             color = 'red',size = 3,alpha = .6) +
  labs(title = "Univariate Visualization of the Errors",
       y = "Density",
       x = "Errors: Actual - Predicted")
```



```
# Multivariate visualization of the errors
mvAnalysis %>%
  ggplot(aes(x = runtime,y = errors)) +
  geom_point() +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  geom_smooth() +
  labs(title = "Multivariate Visualization of the Errors",
       y = "Errors",
       x = "Runtime (minutes)")
```

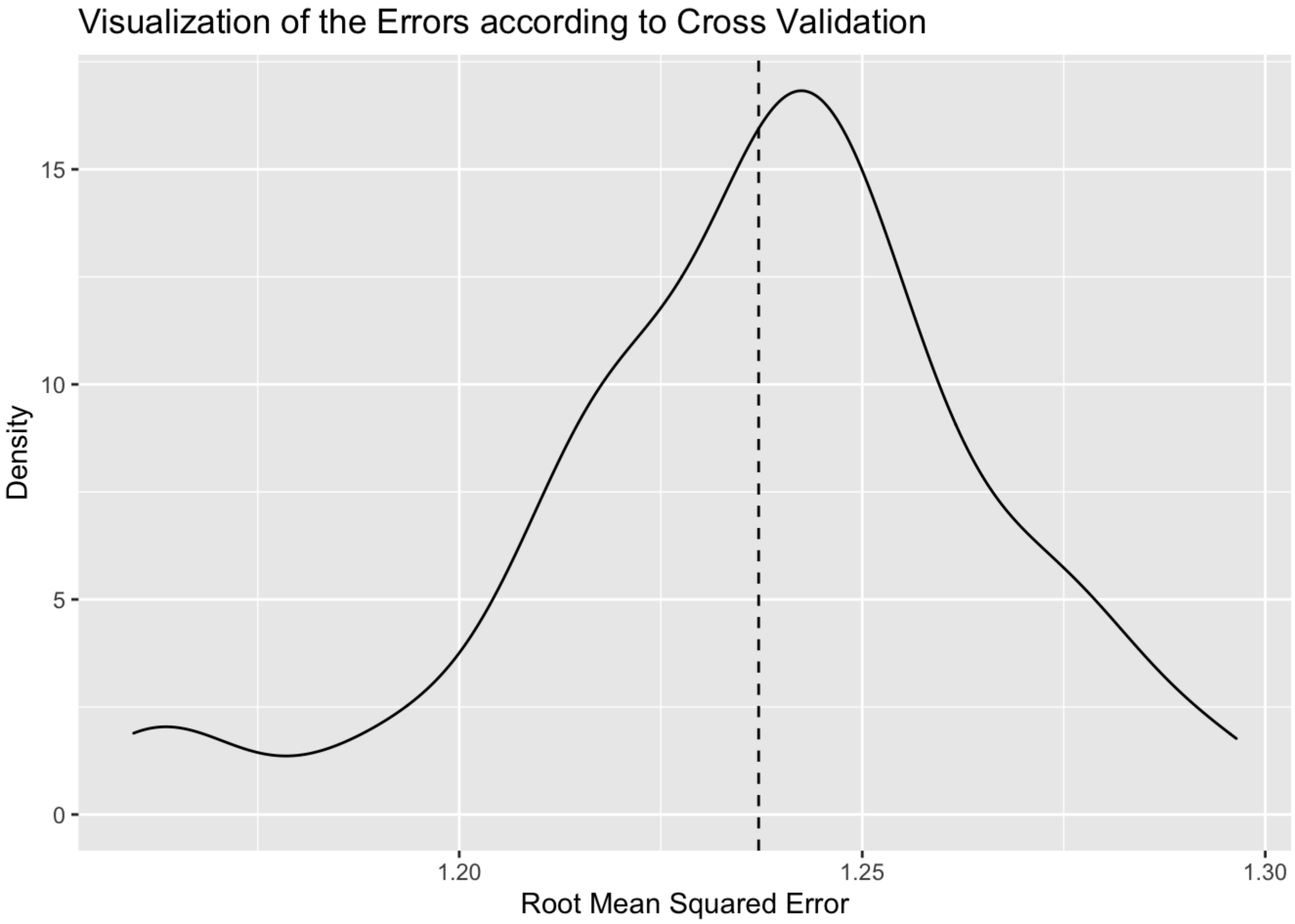
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
# 4. Evaluate model fit with and 2) cross validation.
mvv <- mv %>%
  select(runtime, budget) %>%
  drop_na()

set.seed(123)
bsRes <- NULL
for(i in 1:100) {
  inds <- sample(1:nrow(mvv), size = round(nrow(mvv)/2),replace = F)
  train <- mvv %>% slice(inds)
  test <- mvv %>% slice(-inds)

  mTrain <- lm(log(budget) ~ runtime, train)

  test$preds <- predict(mTrain, newdata = test)
  rmse <- sqrt(mean((log(test$budget) - test$preds)^2,na.rm=T))
  bsRes <- c(bsRes,rmse)
}
mean(rmse)
```

```
## [1] 1.272179
```

```
data.frame(rmseBS = bsRes) %>%
  ggplot(aes(x = rmseBS)) +
  geom_density() +
  geom_vline(xintercept = mean(bsRes), linetype = 'dashed') +
  labs(title = "Visualization of the Errors according to Cross Validation",
       y = "Density",
       x = "Root Mean Squared Error")
```



> - #1 Both run-time and budget variables are stores as double values. They are both continuous variables. Budget has 4482 NA values while run-time variable only has 4 NA values according to the database summary. #2 The scatter-plot and the line of best fit support the hypothesis that the longer running movies take more time to make therefore cost bigger budgets. #3 & 4 According to the uni-variate model fit with visual analysis of the residuals, there is a left skew in the density distribution of the errors, meaning that we are overestimating (at average, the model predicts that the budget is much higher than it should be when it is actually not that high). According to the multivariate visualization of the residuals, our model predicts less accurately for the longer running movies: the model assumes that the longer running movies have more budget at average than they actually do. There is a slight downward sloping best fit line for the run time between 160 and 200 minutes, meaning that the model overestimates the predicted budget of the movies more when run time increases. When I introduced cross validation and conducted a bootstrap to avoid over-fitting, I got a MSRE of 1.27 while the MSRE was 1.24 for the initial calculation without the test and trained data. This is in line with out assumption that we cross validation would result in less over-fitting of the data.

Question 2 [10 points]

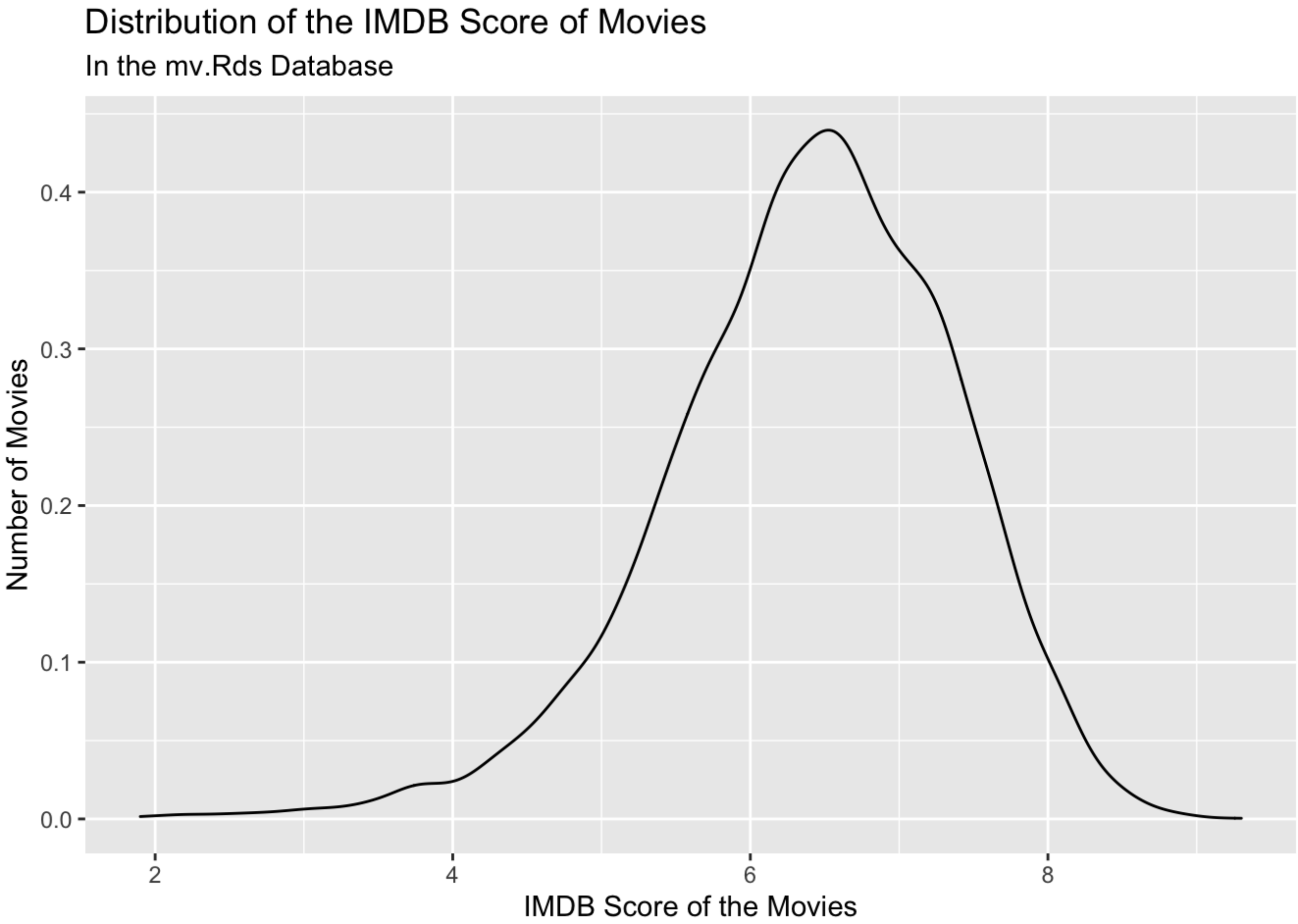
- **Research Question:** Are movies getting worse?
- **Theory:** Changes in the economy have resulted in Hollywood taking fewer and fewer artistic risks, prioritizing cinematic universes like Marvel and reducing the overall quality of movies.
- **Hypothesis:** The more recent the movie, the worse it is.
- *Hints:* You are looking at the conditional relationship between `score` and `year`.

```
#1. Perform uni-variate analysis on each of the two variables separately (determine t
heir class, identifying missing-nes, and plot with the appropriate figure).

#look at the data to find missing data.
mv_sum <- mv %>%
  select(score, year) %>%
  summary()

mv_sum
```

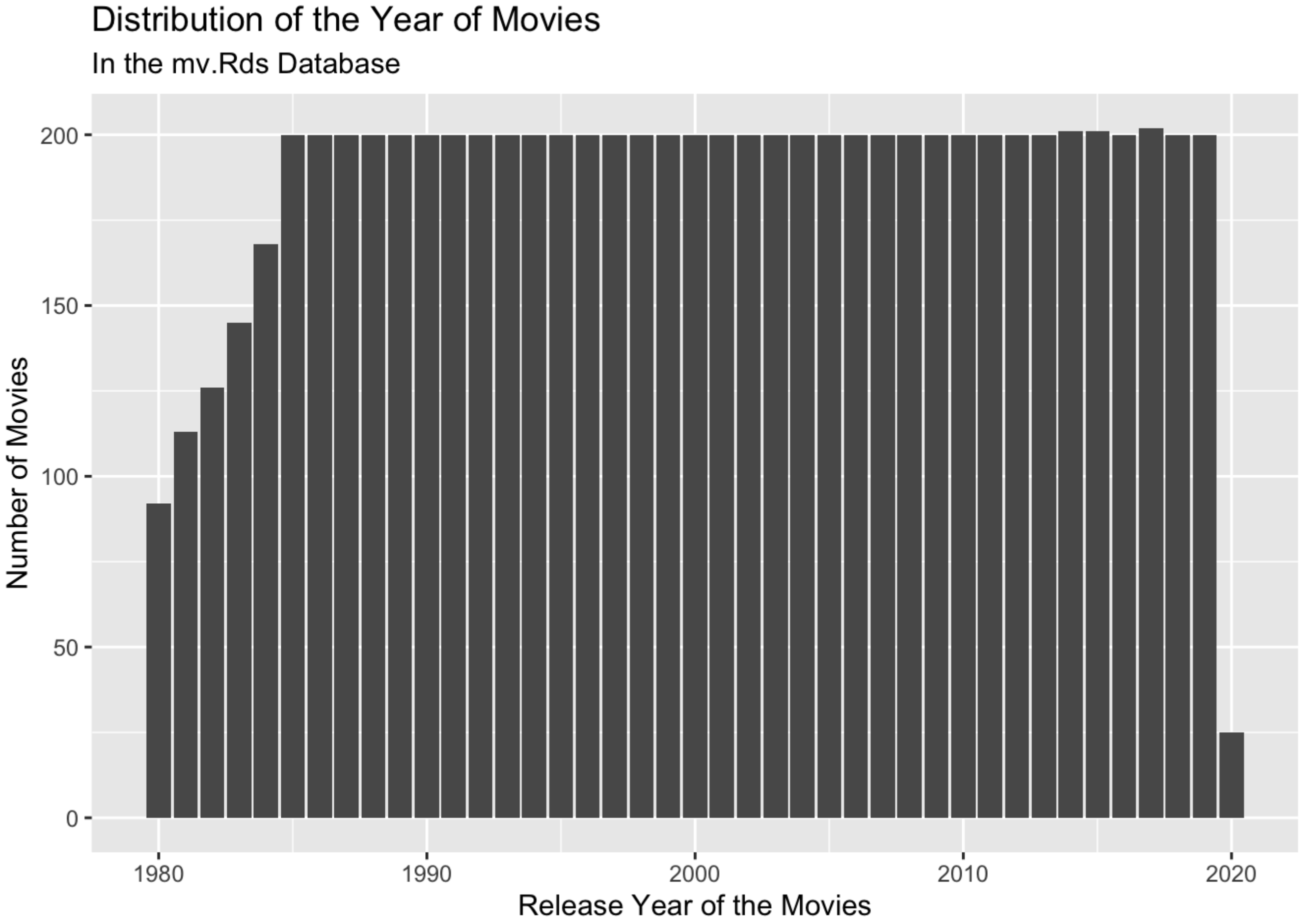
```
##      score            year     
##  Min.   :1.900   Min.   :1980  
##  1st Qu.:5.800   1st Qu.:1991  
##  Median :6.500   Median :2000  
##  Mean   :6.391   Mean   :2000  
##  3rd Qu.:7.100   3rd Qu.:2010  
##  Max.   :9.300   Max.   :2020  
##  NA's   :3
```

```
var1 <- mv %>%
  ggplot(aes(x = score, na.rm=T)) +
  geom_density() +
  labs(title = "Distribution of the IMDB Score of Movies",
       subtitle = "In the mv.Rds Database",
       y = "Number of Movies",
       x = "IMDB Score of the Movies")

var1
```

```
## Warning: Removed 3 rows containing non-finite values (stat_density).
```

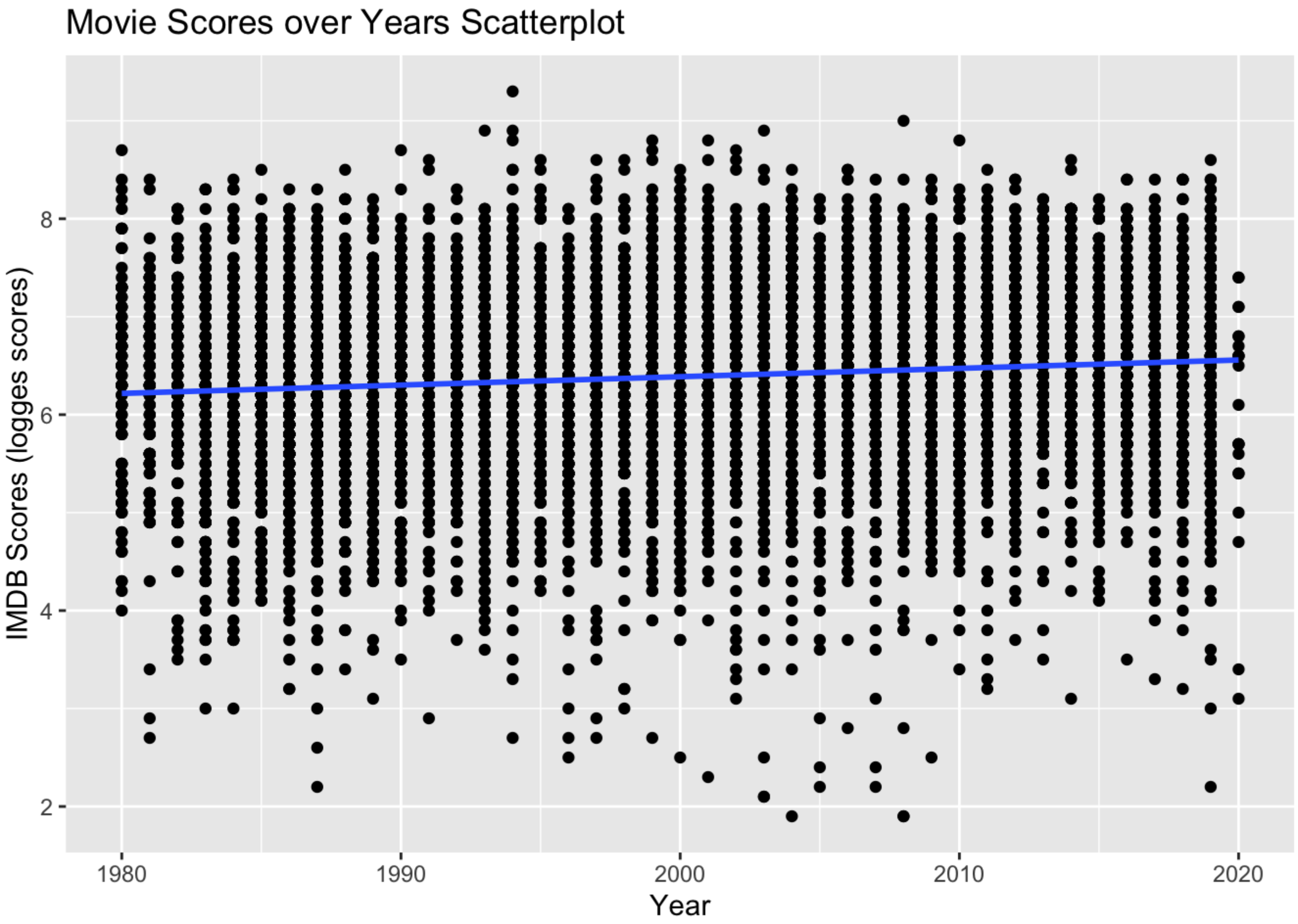


```
var2 <- mv %>%
  ggplot(aes(x = year, na.rm=T)) +
  geom_bar() +
  labs(title = "Distribution of the Year of Movies",
       subtitle = "In the mv.Rds Database",
       y = "Number of Movies",
       x = "Release Year of the Movies")

var2
```



```
# 2. Plot the conditional relationship between the two variables using the appropriat
e figure. Transform the variables if needed by logging highly skewed data.
p2 <- mv %>%
  select(score,year) %>%
  filter(complete.cases(.))

p5 <- p2 %>%
  ggplot(aes(x = year, y = score)) +
  geom_point() +
  geom_smooth(method = 'lm', se = F) +
  labs(title = "Movie Scores over Years Scatterplot",
       y = "IMDB Scores (logges scores)",
       x = "Year")

p5
```

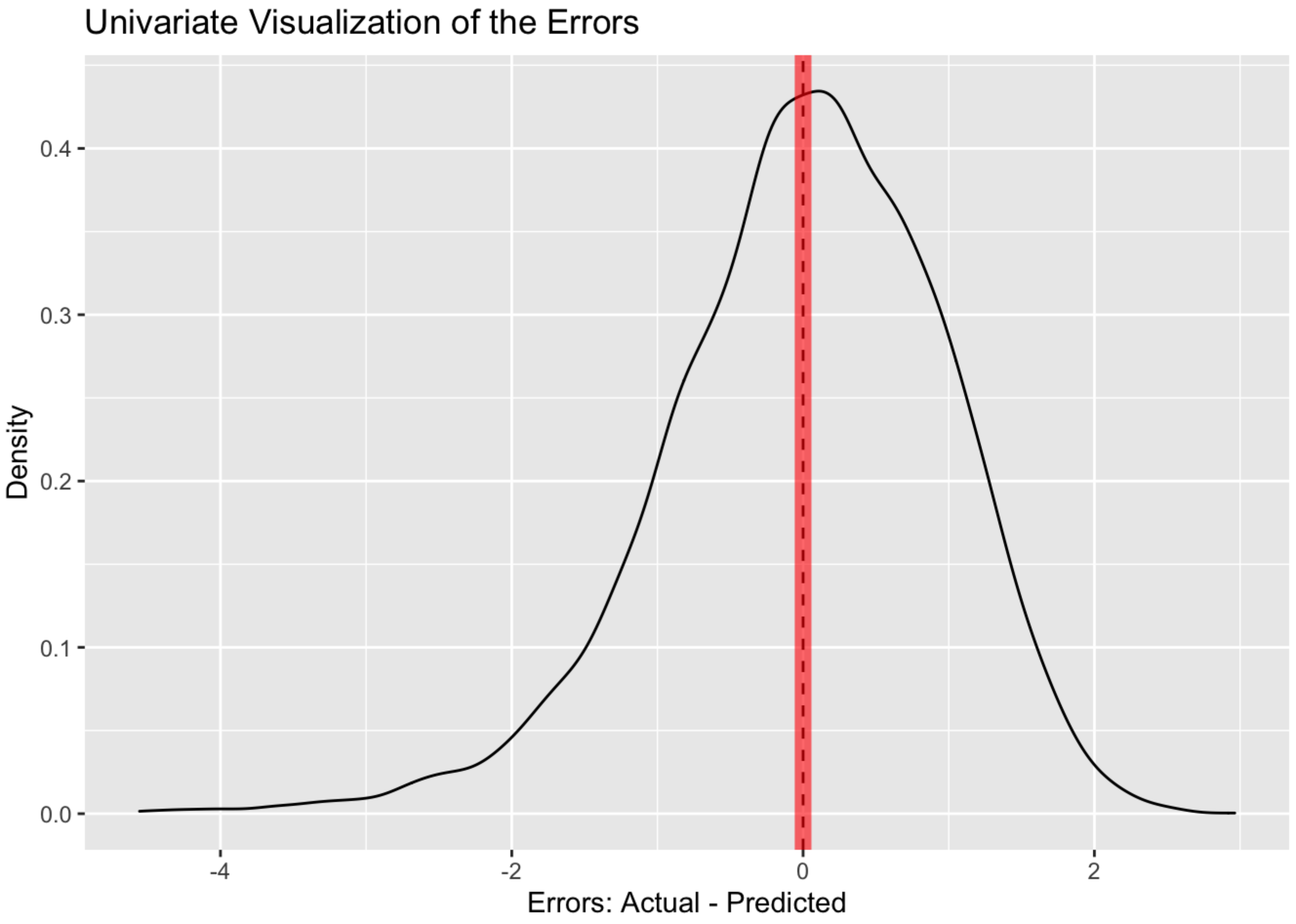
```
## `geom_smooth()` using formula 'y ~ x'
```



```
# 3. Estimate the linear regression model.
#clear the data set of NA values
q2 <- mv %>%
  drop_na(year,score)

m2 <- lm(formula = score ~ year, data = q2)

q2 <- q2 %>%
  mutate(predicted_score = predict(m2)) %>%
  mutate(errors = score - predicted_score)

#RMSE
# E: errors
e2 <- q2$score - q2$predicted_score
# SE: squared
se2 <- e2^2
# MSE: mean
mse2 <- mean(se2, na.rm = T)
# RMSE: root
(rmse2 <- sqrt(mse2))
```

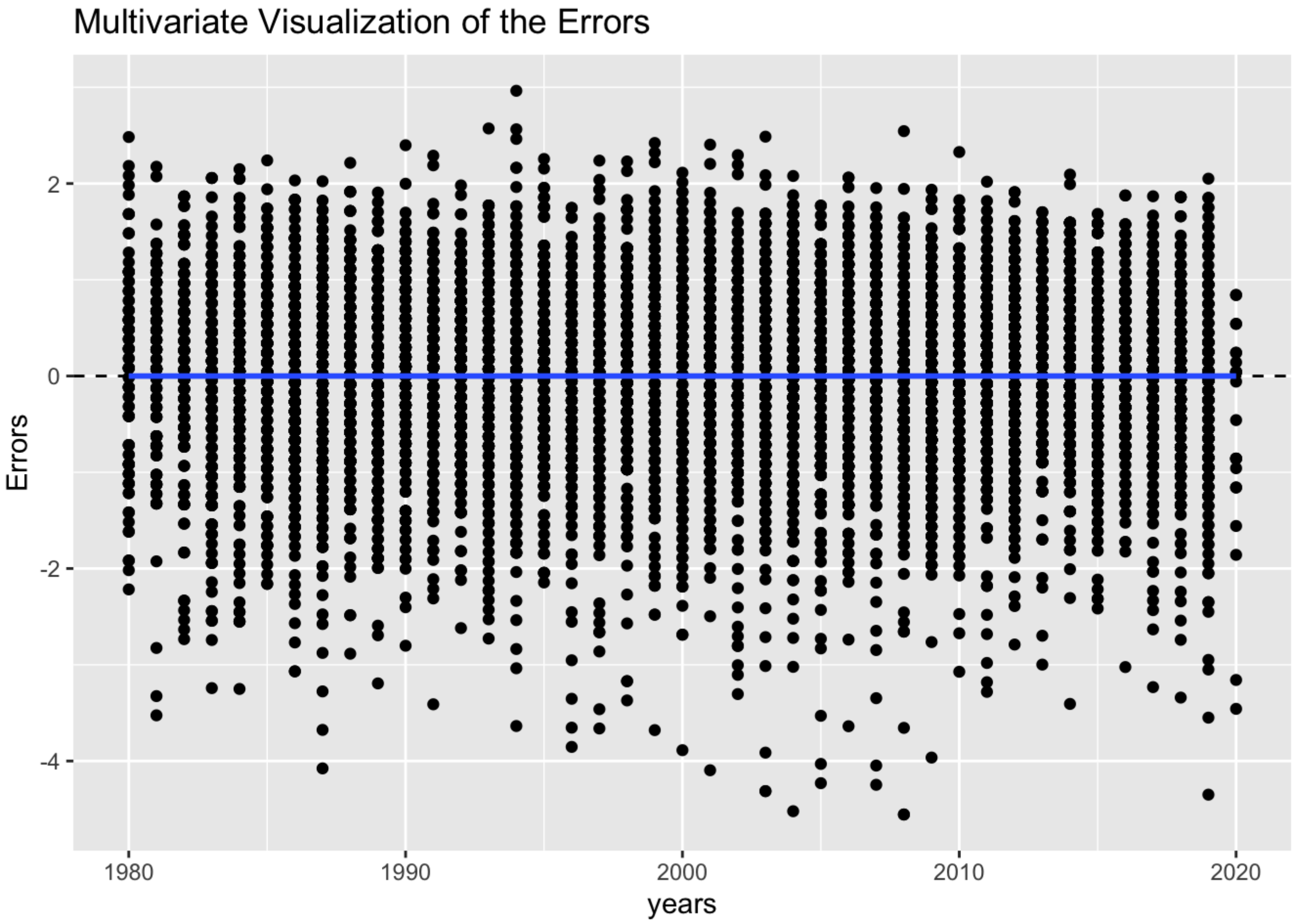
```
## [1] 0.9640419
```

```
# 4. Evaluate model fit with 1) visual analysis of the residuals
mvAnalysis2 <- mv %>%
  select(score, year) %>%
  drop_na()

m_3 <- lm(score ~ year,mvAnalysis2)

pred_vals <- predict(m_3)

errors <- resid(m_3)

mvAnalysis2$pred_vals <- pred_vals

mvAnalysis2$errors <- log(mvAnalysis2$score) - mvAnalysis2$pred_vals

mvAnalysis2 <- mvAnalysis2 %>%
  mutate(errors =  score - pred_vals)

# Uni-variate visualization of the errors
mvAnalysis2 %>%
  ggplot(aes(x = errors)) +
  geom_density() +
  geom_vline(xintercept = 0,linetype = 'dashed') +
  geom_vline(xintercept = mean(mvAnalysis2$errors),
             color = 'red',size = 3,alpha = .6) +
  labs(title = "Univariate Visualization of the Errors",
       y = "Density",
       x = "Errors: Actual - Predicted")
```



```
# Multivariate visualization of the errors
mvAnalysis2 %>%
  ggplot(aes(x = year,y = errors)) +
  geom_point() +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  geom_smooth() +
  labs(title = "Multivariate Visualization of the Errors",
       y = "Errors",
       x = "years")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



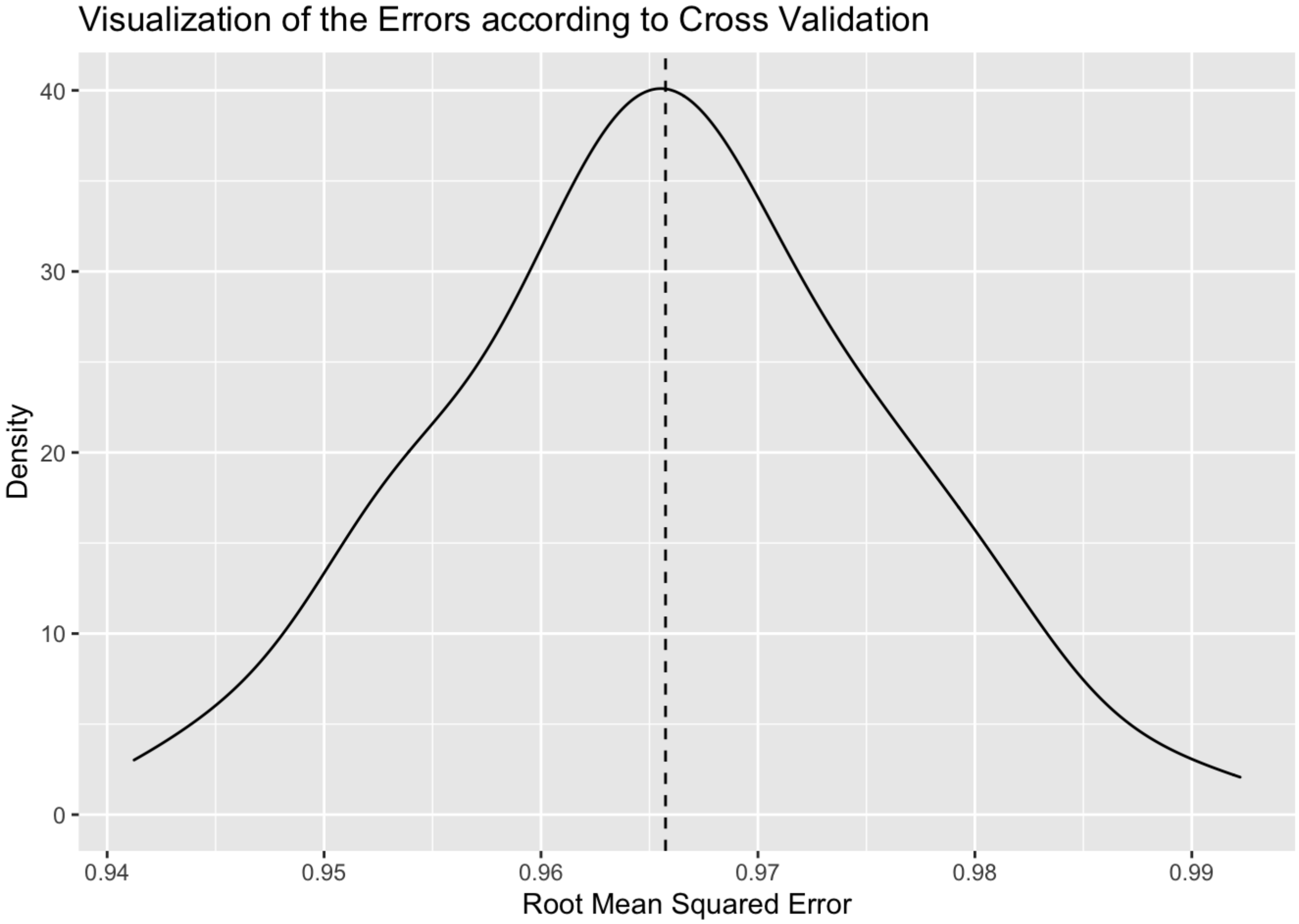
```
# 4. Evaluate model fit with and 2) cross validation.
mv %>%
  select(score, year) %>%
  drop_na()
```

```
## # A tibble: 7,670 × 2
##    score  year
##    <dbl> <dbl>
##  1   8.4  1980
##  2   5.8  1980
##  3   8.7  1980
##  4   7.7  1980
##  5   7.3  1980
##  6   6.4  1980
##  7   7.9  1980
##  8   8.2  1980
##  9   6.8  1980
## 10   7    1980
## # … with 7,660 more rows
```

```
set.seed(123)
bsRes <- NULL
for(i in 1:100) {
  inds <- sample(1:nrow(mv), size = round(nrow(mv)/2),replace = F)
  train <- mv %>% slice(inds)
  test <- mv %>% slice(-inds)

  mTrain <- lm(score ~ year, train)

  test$preds <- predict(mTrain, newdata = test)
  rmse <- sqrt(mean((test$score - test$preds)^2,na.rm=T))
  bsRes <- c(bsRes,rmse)
}
mean(rmse)
```

```
## [1] 0.9605156
```

```
data.frame(rmseBS = bsRes) %>%
  ggplot(aes(x = rmseBS)) +
  geom_density() +
  geom_vline(xintercept = mean(bsRes), linetype = 'dashed') +
  labs(title = "Visualization of the Errors according to Cross Validation",
       y = "Density",
       x = "Root Mean Squared Error")
```


Visualization of the Errors according to Cross Validation
Density
40
30
20
10
0
0.94
0.95
0.96
0.97
0.98
0.99
Root Mean Squared Error

> - #1 The budget is stored as a continuous variable of class double and the year is also stored as a double class. However, it is important to note that treating year as a categorical variable will allow for better data visualization using bar plots, but for the sake of calculating the RMSE and performing cross validation, I will treat year as a continuous variable. The score variable has 3 missing data while the year variable does not have any missing data according to the selected summary. #2 The above graph titled “Movie Scores over Years Scatter plot” refutes the hypothesis that the more recent the movie, the worse it is. According to the line of best fit to the scatter-plot above, there is a very slight upward trend in the IMDB scores of recent Hollywood movies as opposed to significant negative correlation. #3 & 4 According to the uni-variate model fit with visual analysis of the residuals, there is a slight left skew in the density distribution of the errors, meaning that we are overestimating (at average, the model predicts that the score is higher than it should be when it is actually not that high). According to the multivariate visualization of the residuals, our model’s prediction ability stays constant over the years as the best fit line lies on the horizontal y=0 line. There is no more overestimating or no more underestimating done in one year compared to other years. The model estimates scores for years consistently. When I introduced cross validation and conducted a bootstrap to avoid over-fitting, I got a MSRE of 0.96 which is essentially the same as the initial calculation without the test and trained data.

Question 3 [5 EXTRA CREDIT points]

Create your own research question, theory, and hypothesis using the movie data, and answer it.

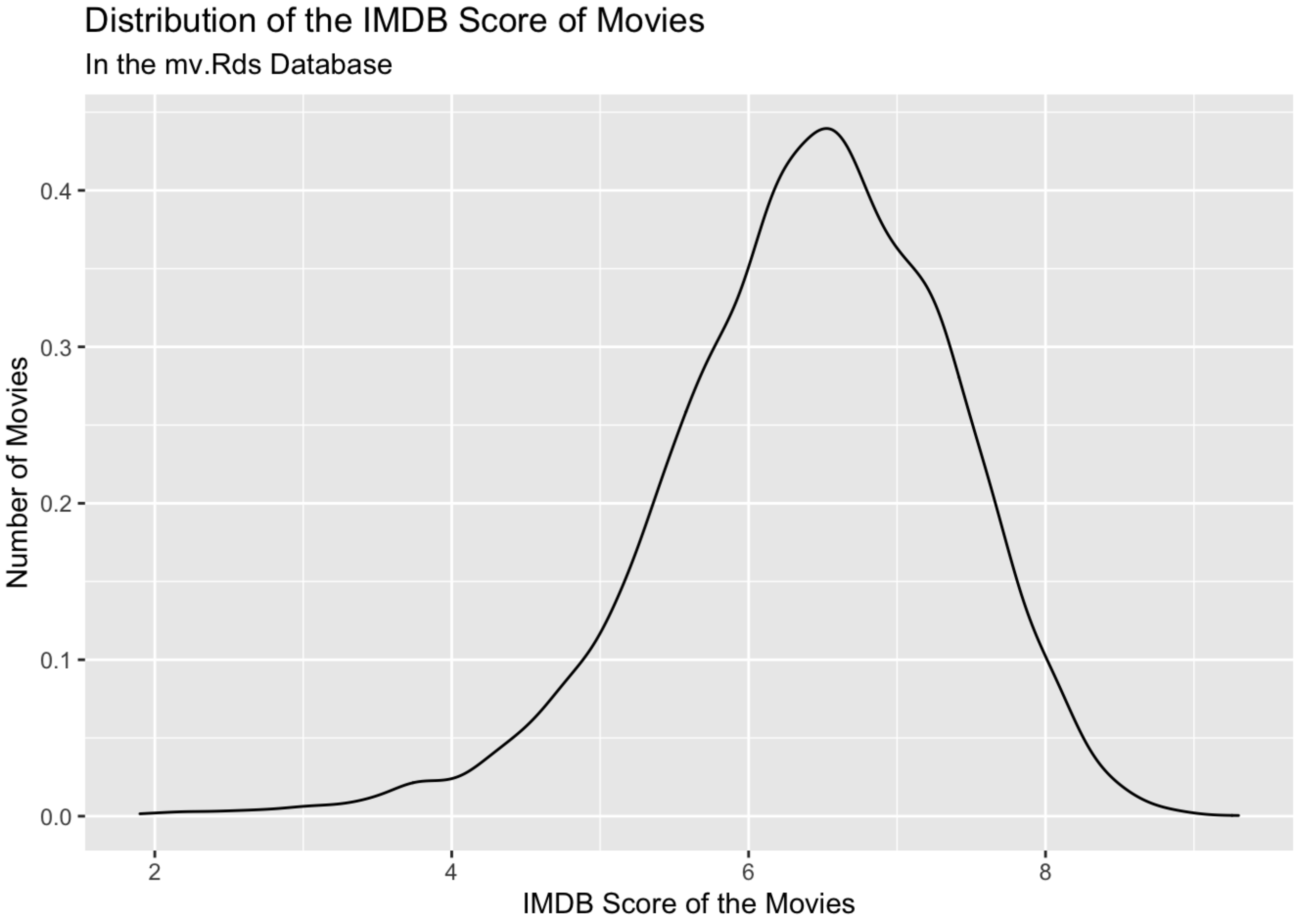
- **Research Question:** Does American movies score better than the rest of the movies?
- **Theory:** American culture has influence on many international countries, therefore people like to watch American movies.
- **Hypothesis:** American movies score better than the rest of the movies.

```
# 1. Uni-variate analysis on each of the two variables separately (determine their cl
ass, identifying missing data, and plot with the appropriate figure).
sum <- mv %>%
  select(score, country) %>%
  summary()

sum
```

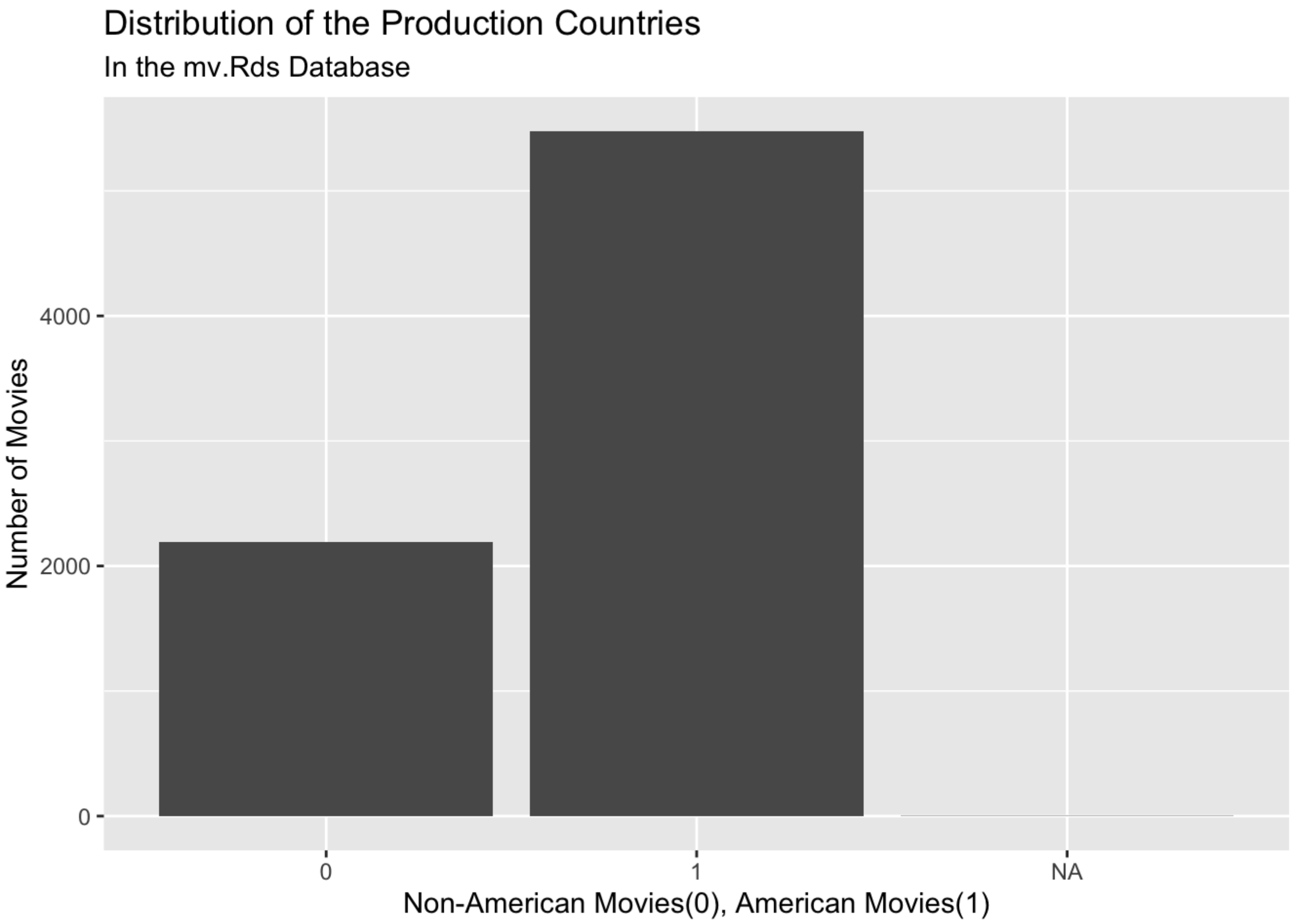
```
##      score          country         
##  Min.   :1.900   Length:7673       
##  1st Qu.:5.800   Class :character  
##  Median :6.500   Mode  :character  
##  Mean   :6.391                     
##  3rd Qu.:7.100                     
##  Max.   :9.300                     
##  NA's   :3
```

```
var3 <- mv %>%
  ggplot(aes(x = score, na.rm=T)) +
  geom_density() +
  labs(title = "Distribution of the IMDB Score of Movies",
       subtitle = "In the mv.Rds Database",
       y = "Number of Movies",
       x = "IMDB Score of the Movies")

var3
```

```
## Warning: Removed 3 rows containing non-finite values (stat_density).
```



```
var4 <- mv %>%
  mutate(american = ifelse(country == "United States", 1, 0)) %>%
  ggplot(aes(x = factor(american), na.rm=T)) +
  geom_bar() +
  labs(title = "Distribution of the Production Countries",
       subtitle = "In the mv.Rds Database",
       y = "Number of Movies",
       x = "Non-American Movies(0), American Movies(1)")

var4
```

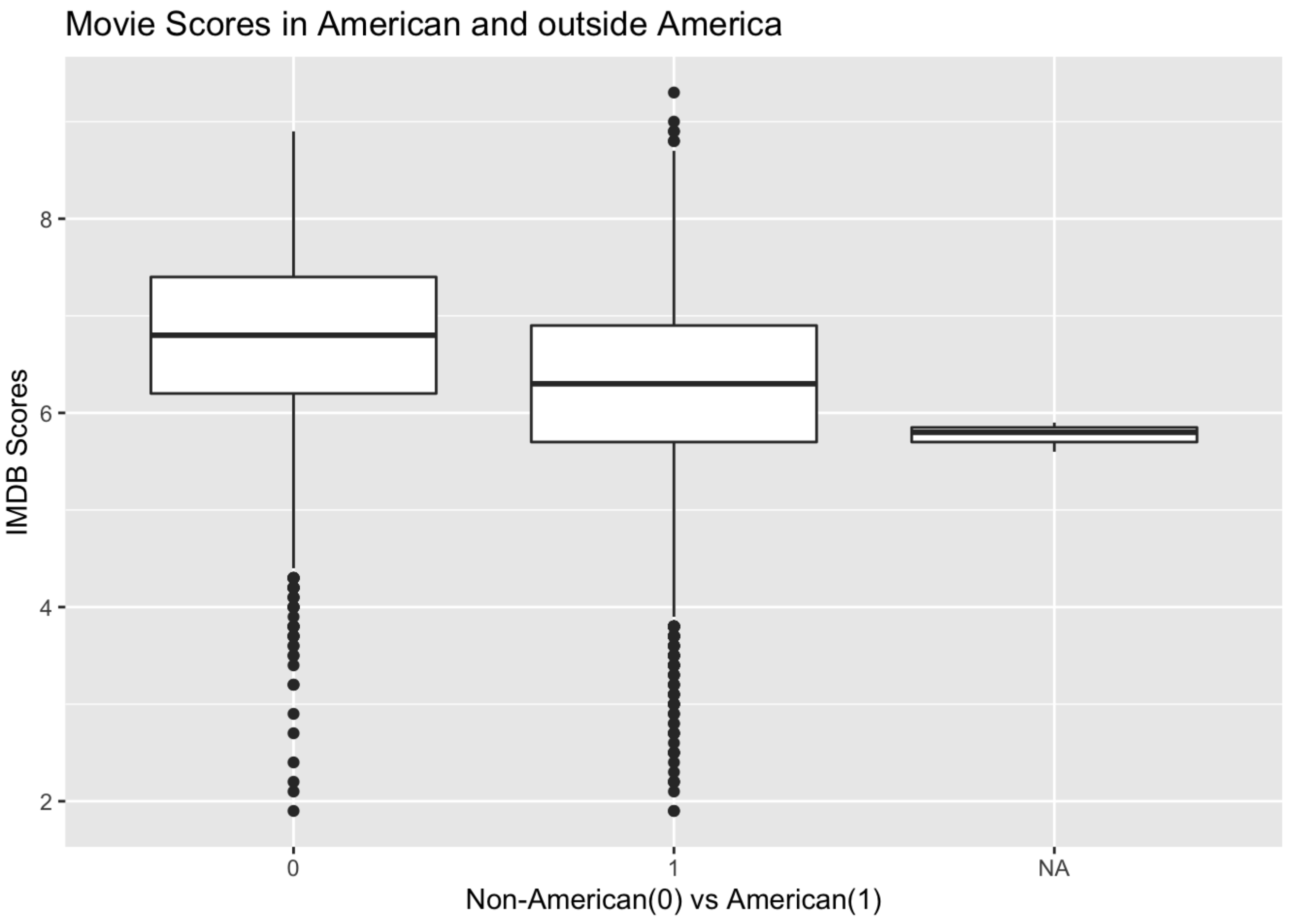


```
# 2. Plot the conditional relationship between the two variables using the appropriat
e figure. Transform the variables if needed by logging highly skewed data.

p <- mv %>%
  mutate(american = ifelse(country == "United States", 1, 0)) %>%
  mutate(american_factor = recode_factor(american,`1` = "american", `0` = "non-americ
an",)) %>%
  select(american, score) %>%
  ggplot(aes(x = factor(american), y = score, na.rm=T)) +
  geom_boxplot() +
  geom_smooth(method = 'lm', se = F) +
  labs(title = "Movie Scores in American and outside America",
       y = "IMDB Scores",
       x = "Non-American(0) vs American(1)")
p
```

```
## Warning: Removed 3 rows containing non-finite values (stat_boxplot).
```

```
## `geom_smooth()` using formula 'y ~ x'
```

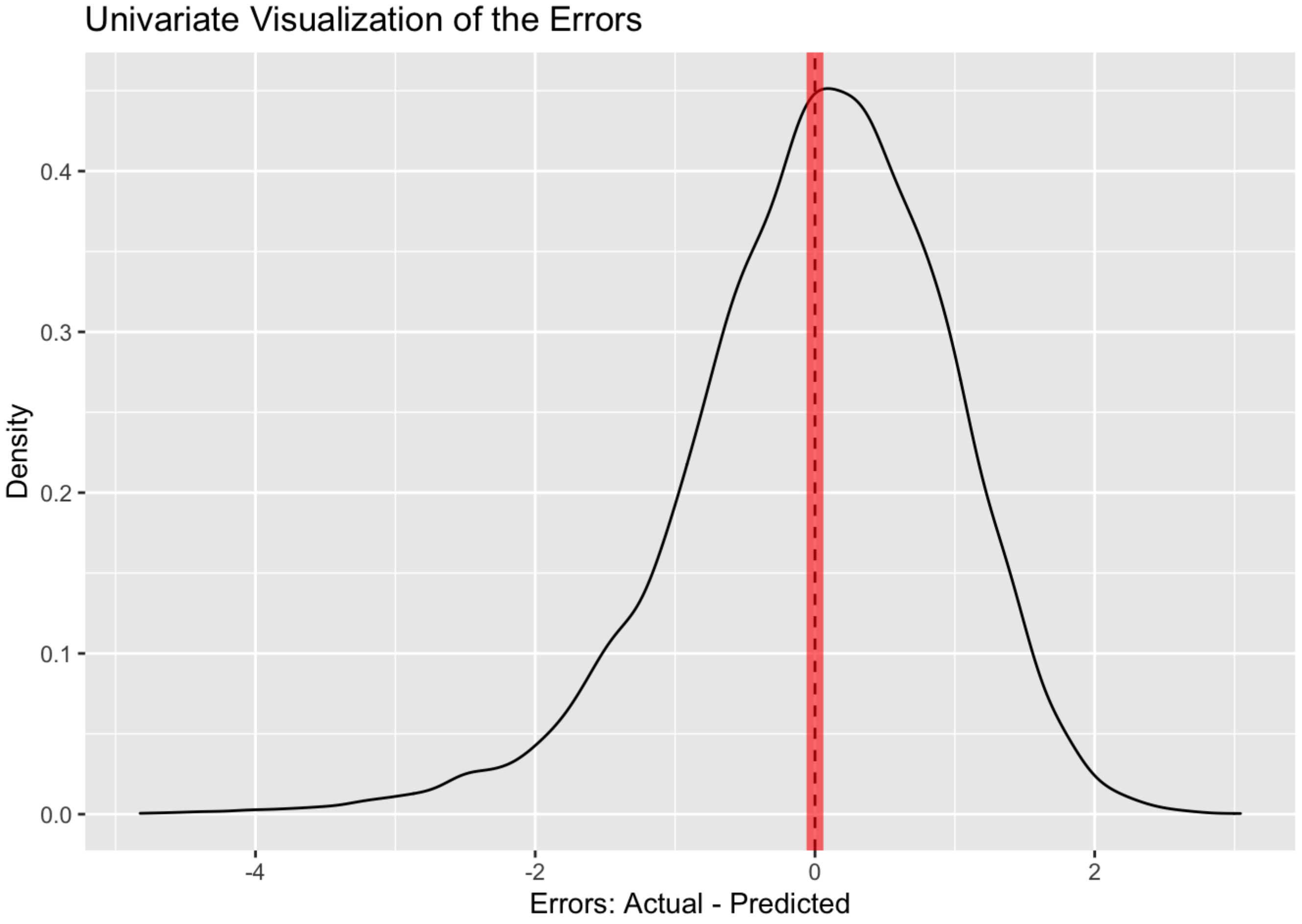
```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```



```
# 3. Estimate the linear regression model.
#clear the data set of NA values
q2 <- mv %>%
  drop_na(country,score) %>%
  mutate(american = ifelse(country == "United States", 1, 0)) %>%
  mutate(american_factor = recode_factor(american,`1` = "american", `0` = "non-americ
an",))

m2 <- lm(formula = score ~ american_factor, data = q2)

q2 <- q2 %>%
  mutate(predicted_score = predict(m2)) %>%
  mutate(errors = score - predicted_score)

#RMSE
# E: errors
e2 <- q2$score - q2$predicted_score
# SE: squared
se2 <- e2^2
# MSE: mean
mse2 <- mean(se2, na.rm = T)
# RMSE: root
(rmse2 <- sqrt(mse2))
```

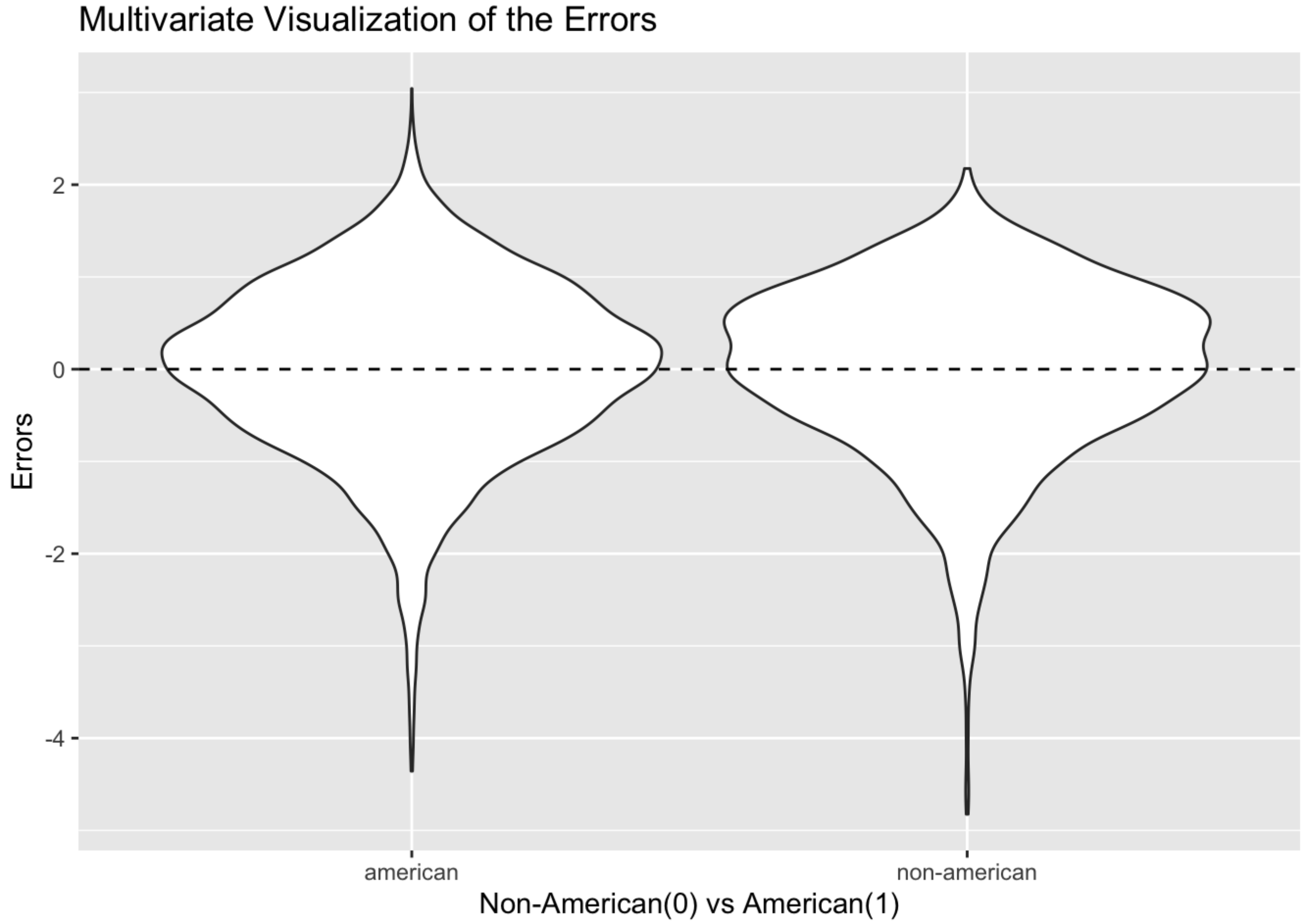
```
## [1] 0.9455776
```

```
# 4. Evaluate model fit with 1) visual analysis of the residuals
mvAnalysis2 <- mv %>%
  select(score, country) %>%
  drop_na() %>%
  mutate(american = ifelse(country == "United States", 1, 0)) %>%
  mutate(american_factor = recode_factor(american,`1` = "american", `0` = "non-americ
an",))


m_3 <- lm(score ~ american_factor,mvAnalysis2)

pred_vals <- predict(m_3)

errors <- resid(m_3)

mvAnalysis2$pred_vals <- pred_vals

mvAnalysis2$errors <- mvAnalysis2$score - mvAnalysis2$pred_vals

mvAnalysis2 <- mvAnalysis2 %>%
  mutate(errors = score - pred_vals)

# Uni-variate visualization of the errors
mvAnalysis2 %>%
  ggplot(aes(x = errors)) +
  geom_density() +
  geom_vline(xintercept = 0,linetype = 'dashed') +
  geom_vline(xintercept = mean(mvAnalysis2$errors),
             color = 'red',size = 3,alpha = .6) +
  labs(title = "Univariate Visualization of the Errors",
       y = "Density",
       x = "Errors: Actual - Predicted")
```



```
# Multivariate visualization of the errors
mvAnalysis2 %>%
  ggplot(aes(x = american_factor,y = errors)) +
  geom_violin() +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  geom_smooth() +
  labs(title = "Multivariate Visualization of the Errors",
       y = "Errors",
       x = "Non-American(0) vs American(1)")
```

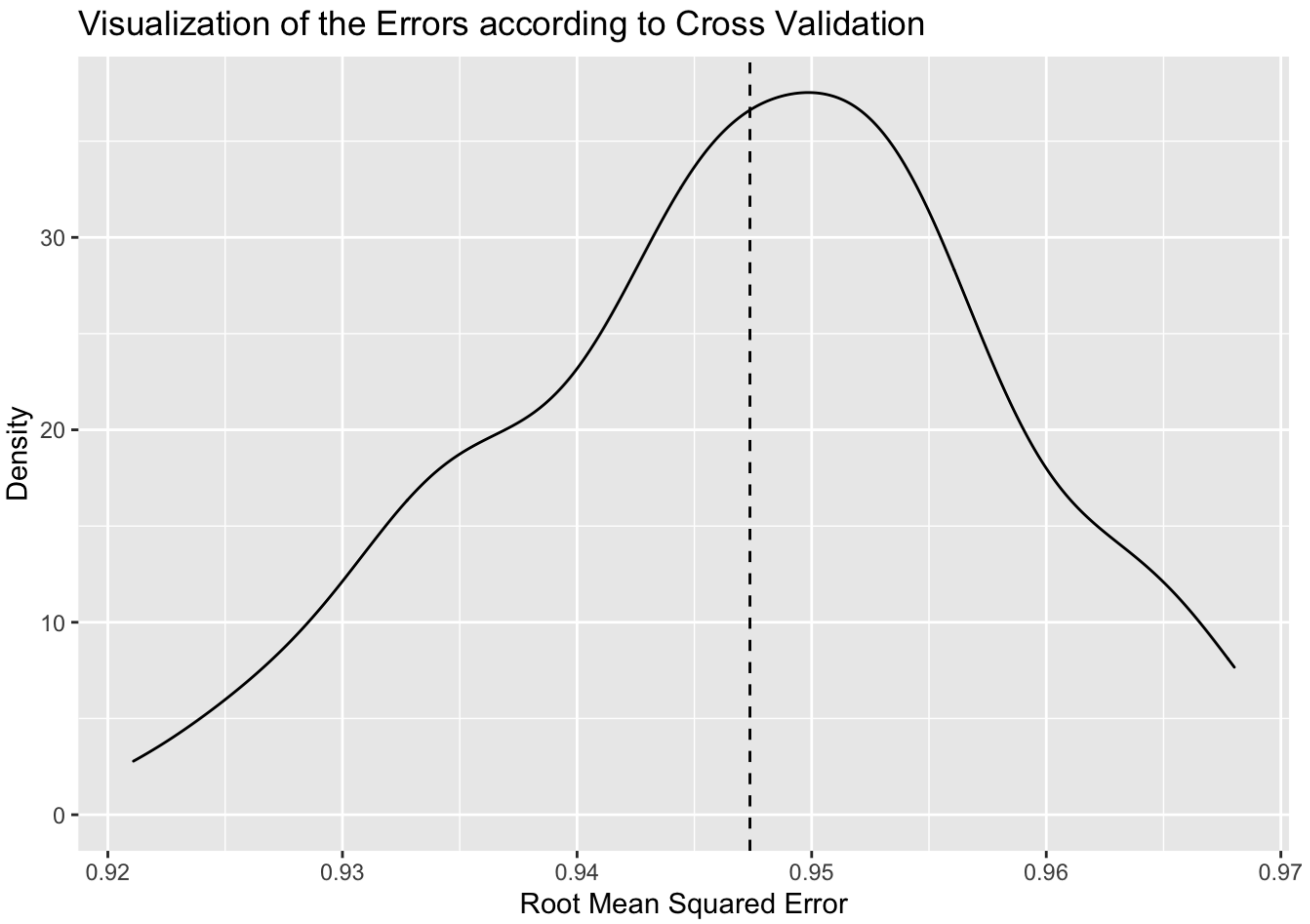
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
# 4. Evaluate model fit with and 2) cross validation.
m <- mv %>%
  drop_na(score, country) %>%
  mutate(american = ifelse(country == "United States", 1, 0)) %>%
  mutate(american_factor = recode_factor(american,`1` = "american", `0` = "non-americ
an",))

set.seed(123)
bsRes <- NULL
for(i in 1:100) {
  inds <- sample(1:nrow(m), size = round(nrow(m)/2),replace = F)
  train <- m %>% slice(inds)
  test <- m %>% slice(-inds)

  mTrain <- lm(score ~ american_factor, train)

  test$preds <- predict(mTrain, newdata = test)
  rmse <- sqrt(mean((test$score - test$preds)^2,na.rm=T))
  bsRes <- c(bsRes,rmse)
}
mean(rmse)
```

```
## [1] 0.9514615
```

```
data.frame(rmseBS = bsRes) %>%
  ggplot(aes(x = rmseBS)) +
  geom_density() +
  geom_vline(xintercept = mean(bsRes), linetype = 'dashed') +
  labs(title = "Visualization of the Errors according to Cross Validation",
       y = "Density",
       x = "Root Mean Squared Error")
```



- #1 Country is stored as a character class variables while the score variable is stored as a double class. The score variable is continuous and the country variable is categorical. I am going to further categorize the country variable into a binary category: american vs non-american to better investigate the research question. Score variable has 3 missing data values while the country variable does not have any missing data. #2 According to the box-plot visualization of the correlation between two variables, the research hypothesis does not hold: The mean IMDB score of non-american movies is higher than that of american movies. The reason for this could be that America produces more movies for mass consumption without focusing on the quality of the movie or its ranking (this is drawn from the uni-variate analysis of the american variable with the bar plot) #3 & 4 According to the uni-variate model fit with visual analysis of the residuals, there is a light left skew in the density distribution of the errors, meaning that we are overestimating (at average, the model predicts that the score is higher for a given movie than it should be). According to the multivariate visualization of the residuals, it can be concluded that our model over-predicts for both american and non-american movies given that there is a downward skew on both of the violin plots. However, the models prediction ability doesn’t greatly differ between the given binary categories of data. When I introduced cross validation and conducted a bootstrap to avoid over-fitting, I got a MSRE of 95 while the MSRE was 94.5 for the initial calculation without the test and trained data. This is in line with out assumption that we cross validation would result in less over-fitting of the data.