

# Regression

## Part 3

Prof. Bisbee

Vanderbilt University

Slides Updated: 2024-01-09

# Agenda

1. Recap of Movie Analysis
2. Multiple Regression
3. Categorical Predictors

# Recap of Movie Analysis

```
require(tidyverse)

mv <-
read_rds('https://github.com/jbisbee1/DS1000_S2024/raw/main/data/mv.Rds')
```

- **Theory**: the more a movie costs, the more it should make
  - If not, Hollywood would go out of business!
- $X$ : budget
- $Y$ : gross

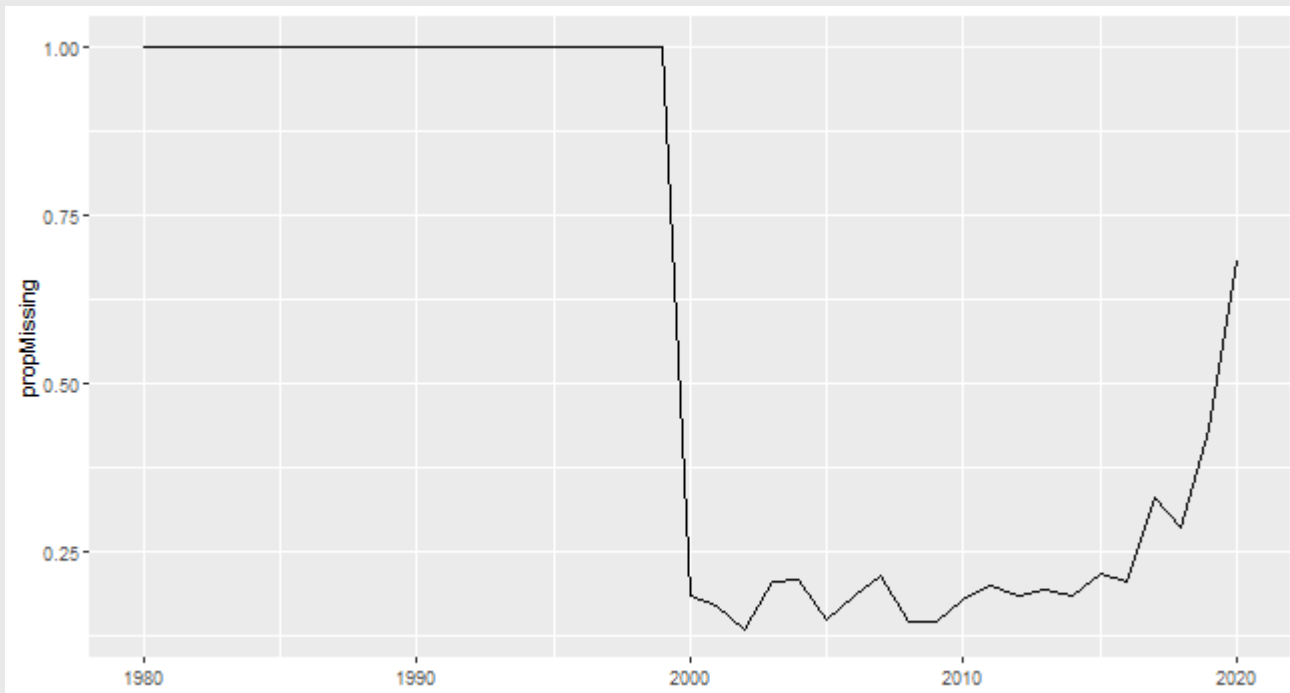
# Step 1: Look

```
summary(mv %>% select(gross,budget))
```

```
##      gross      budget
##  Min.   :7.140e+02   Min.    :    5172
## 1st Qu.:1.121e+07   1st Qu.: 16865322
## Median :5.178e+07   Median : 37212044
## Mean   :1.402e+08   Mean    : 57420173
## 3rd Qu.:1.562e+08   3rd Qu.: 77844746
## Max.   :3.553e+09   Max.    :387367903
## NA's   :3668       NA's     :4482
```

# Step 1: Look

```
mv %>%  
  mutate(missing = ifelse(is.na(gross) | is.na(budget),1,0)) %>%  
  group_by(year) %>%  
  summarise(propMissing = mean(missing)) %>%  
  ggplot(aes(x = year,y = propMissing)) +  
  geom_line()
```



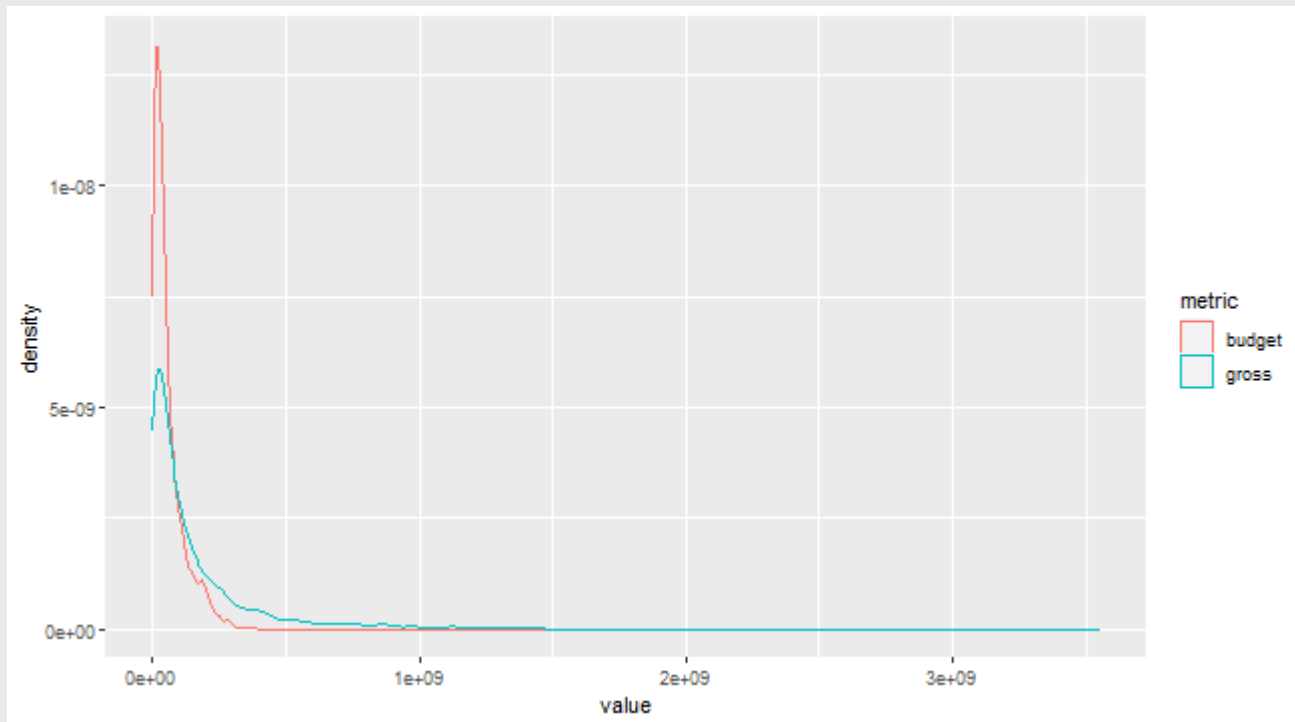
# Some quick wrangling

```
mv <- mv %>%  
  drop_na(gross,budget)  
  
mv %>%  
  select(gross,budget) %>%  
  glimpse()
```

```
## Rows: 3,179  
## Columns: 2  
## $ gross   <dbl> 73677478, 53278578, 723586629, 11490339, 62...  
## $ budget  <dbl> 93289619, 10883789, 160147179, 6996721, 139...
```

# Step 2: Univariate Viz

```
mv %>%  
  select(title,gross,budget) %>%  
  gather(metric,value,-title) %>%  
  ggplot(aes(x = value,color = metric)) +  
  geom_density()
```



# More Wrangling?

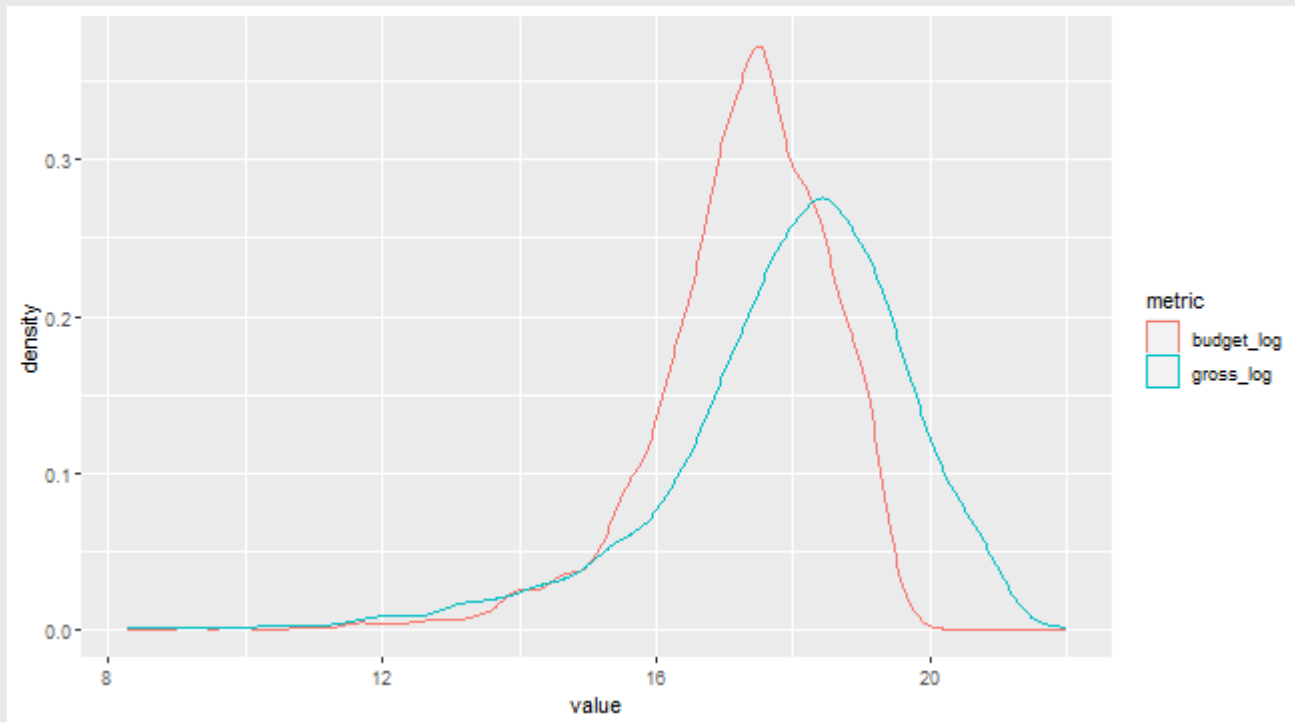
- Univariate visualization highlights significant **skew** in both measures
  - Most movies don't cost a lot and don't make a lot
  - But there are a few blockbusters that pull the density way out
- Let's **wrangle** two new variables that take the log of these skewed measures
  - Logging transforms skewed measures to more "normal" measures

```
mv <- mv %>%  
  mutate(gross_log = log(gross),  
         budget_log = log(budget))
```



# Step 2: Univariate Viz

```
mv %>%  
  select(title,gross_log,budget_log) %>%  
  gather(metric,value,-title) %>%  
  ggplot(aes(x = value,color = metric)) +  
  geom_density()
```



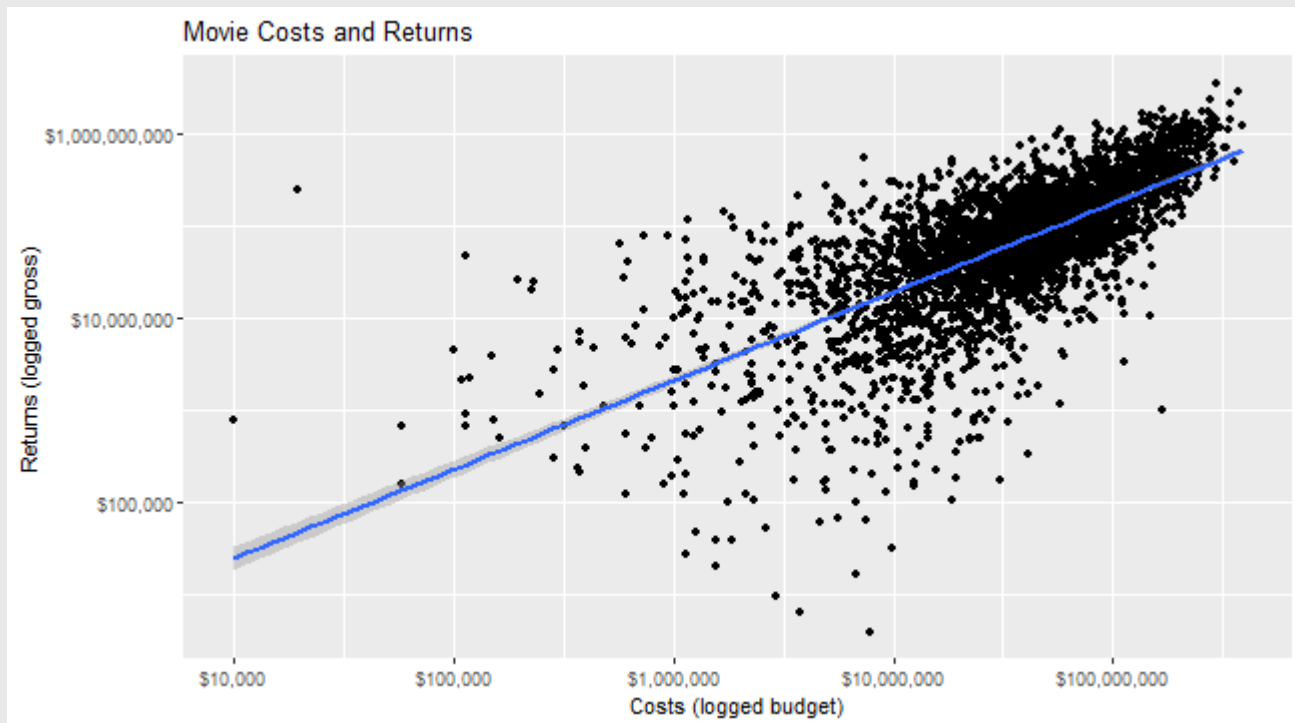
# Step 3: Multivariate Viz

```
pClean <- mv %>%  
  ggplot(aes(x = budget,y = gross)) +  
  geom_point() +  
  scale_x_log10(labels = scales::dollar) +  
  scale_y_log10(labels = scales::dollar) +  
  labs(title = "Movie Costs and Returns",  
        x = "Costs (logged budget)",  
        y = "Returns (logged gross)")
```

# Step 3: Multivariate Viz

```
pClean + geom_smooth(method = 'lm')
```

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



# Step 4: Regression!

```
m <- lm(gross_log ~ budget_log, data = mv)
summary(m)
```

```
##
## Call:
## lm(formula = gross_log ~ budget_log, data = mv)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.2672 -0.6354  0.1648  0.7899  8.5599
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.26107     0.30953   4.074 4.73e-05 ***
## budget_log    0.96386     0.01786  53.971 < 2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.281 on 3177 degrees of freedom
## Multiple R-squared:  0.4783,    Adjusted R-squared:  0.4782
## F-statistic: 2913 on 1 and 3177 DF,  p-value: < 2.2e-16
```

# Step 5.1: Univariate Viz of Errors

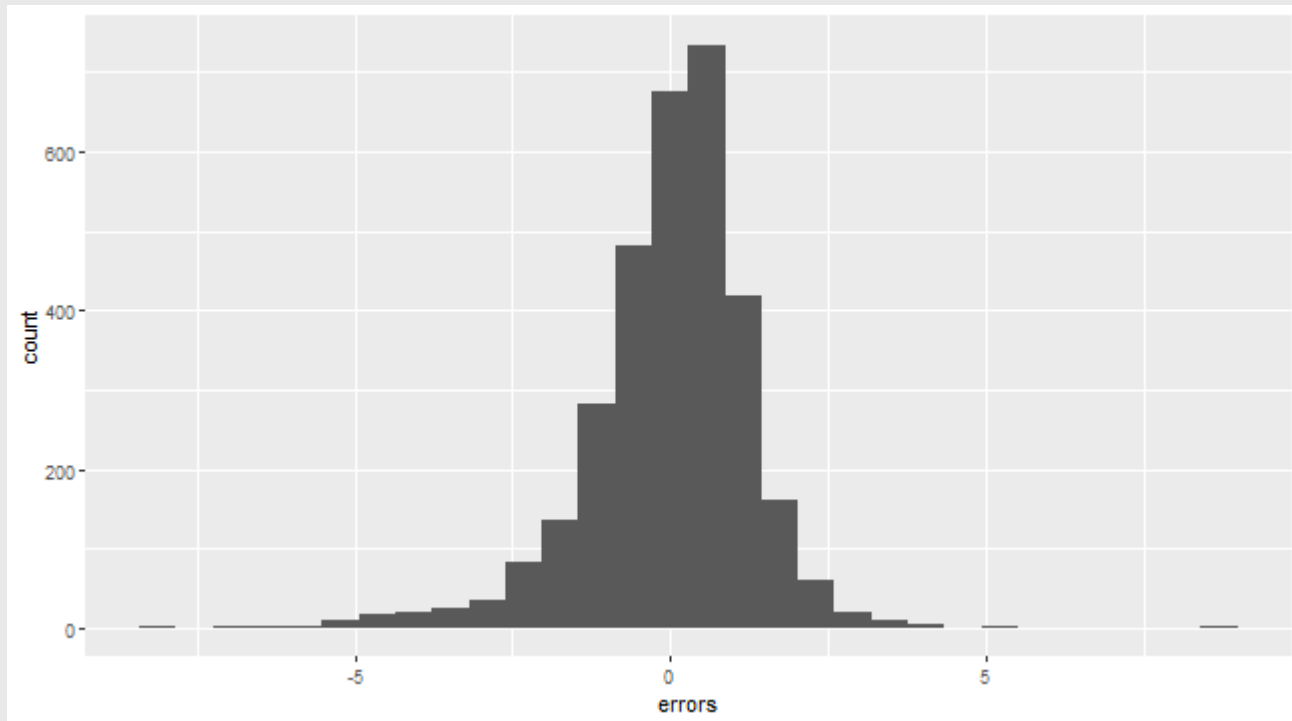
- Errors  $\varepsilon = Y - \hat{Y}$ 
  - In R, can also get them via `resid()` function

```
mv %>%  
  mutate(errors_manual = gross_log - predict(m),  
         errors_resid = resid(m))
```

```
## # A tibble: 3,179 × 24  
##   title    rating genre  year released  score  votes director  
##   <chr>   <chr>  <chr> <dbl> <chr>    <dbl> <dbl> <chr>  
## 1 Almost... R      Adve... 2000 Septemb... 7.9 2.6 e5 Cameron...  
## 2 Americ... R      Come... 2000 April 1... 7.6 5.14e5 Mary Ha...  
## 3 Gladia... R      Acti... 2000 May 5, ... 8.5 1.4 e6 Ridley ...  
## 4 Requie... Unrat... Drama 2000 Decembe... 8.3 7.86e5 Darren ...  
## 5 Memento R      Myst... 2000 May 25,... 8.4 1.20e6 Christo...  
## 6 Cast A... PG-13  Adve... 2000 Decembe... 7.8 5.42e5 Robert ...  
## 7 Scary ... R      Come... 2000 July 7,... 6.2 2.38e5 Keenen ...  
## 8 The Pe... PG-13  Acti... 2000 June 30... 6.4 1.6 e5 Wolfgan...  
## 9 Coyote... PG-13  Come... 2000 August ... 5.7 1.08e5 David M...  
## 10 X-Men PG-13  Acti... 2000 July 14... 7.4 5.82e5 Bryan S...  
## # i 3 169 more rows
```

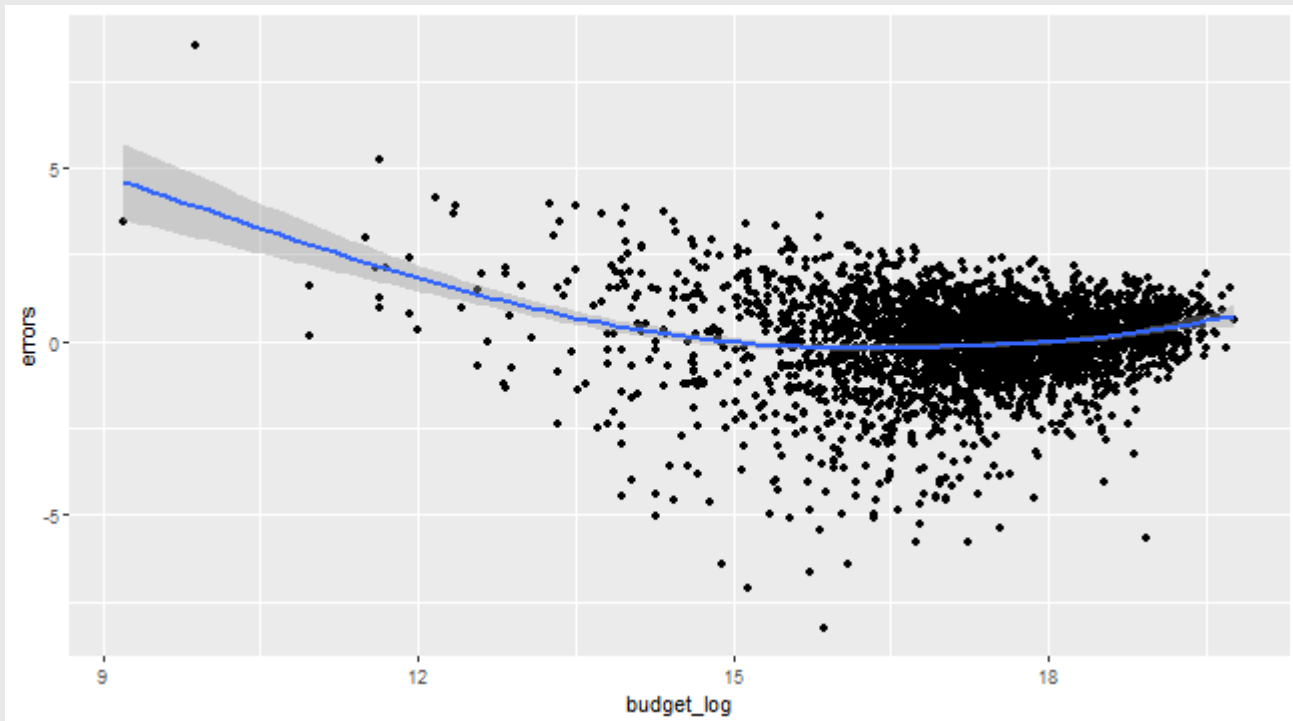
# Step 5.1: Univariate Viz of Errors

```
mv %>%  
  ggplot(aes(x = errors)) +  
  geom_histogram()
```



# Step 5.2: Multivariate Viz of Errors

```
mv %>%  
  ggplot(aes(x = budget_log, y = errors)) +  
  geom_point() +  
  geom_smooth()
```



# Step 5.3: Cross Validated RMSE

```
set.seed(123)
rmseBudget <- 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(gross_log ~ budget_log,train)

  test$preds <- predict(mTrain,newdata = test)

  rmse <- sqrt(mean((test$gross_log - test$preds)^2,na.rm=T))
  rmseBudget <- c(rmseBudget,rmse)
}

mean(rmseBudget)
```

```
## [1] 1.279899
```



# Thinking like a scientist

- Our previous model predicted **gross** as a function of **budget**
- **Theoretically**, is this sensible?
  1. Bigger budgets → famous actors → mass appeal → more tickets
  2. Bigger budgets → advertising money → mass appeal → more tickets
- But what if the movie is just...not good?

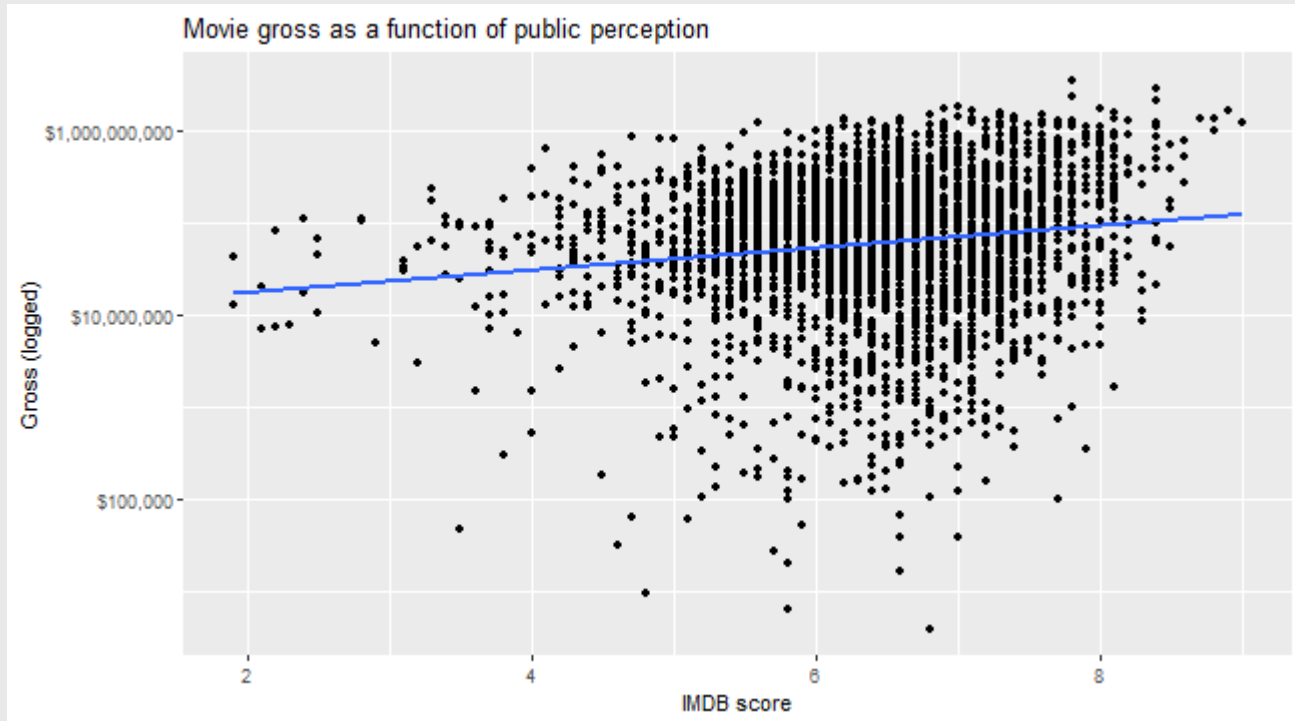
# Alternative Theory

- Good movies make more money
  - Theory: good movies → recommendations → more tickets
- Predict gross with IMDB rating (score)

```
pIMDB <- mv %>%  
  ggplot(aes(x = score, y = gross)) +  
  geom_point() +  
  labs(title = "Movie gross as a function of public perception",  
        x = "IMDB score",  
        y = "Gross (logged)") +  
  scale_y_log10(label = scales::dollar) +  
  geom_smooth(method = 'lm', se = F)
```

# Alternative Model

pIMDB



# Evaluating the Model

- Let's go straight to RMSE
  - We can have R calculate errors for us with `residuals()` command

```
m2 <- lm(gross_log ~ score,mv)
error <- residuals(m2)
(rmseScore <- sqrt(mean(error^2)))
```

```
## [1] 1.753146
```

- Even worse!

# Multivariate Regression

- Recall that we can **model** our outcome with multiple **predictors**

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \varepsilon$$

- How much better can we predict **gross** with **BOTH** **budget** and **score**?

```
m3 <- lm(gross_log ~ budget_log + score, mv)
error <- residuals(m3)
(rmseBudgScore <- sqrt(mean(error^2)))
```

```
## [1] 1.248817
```

# Comparing Models

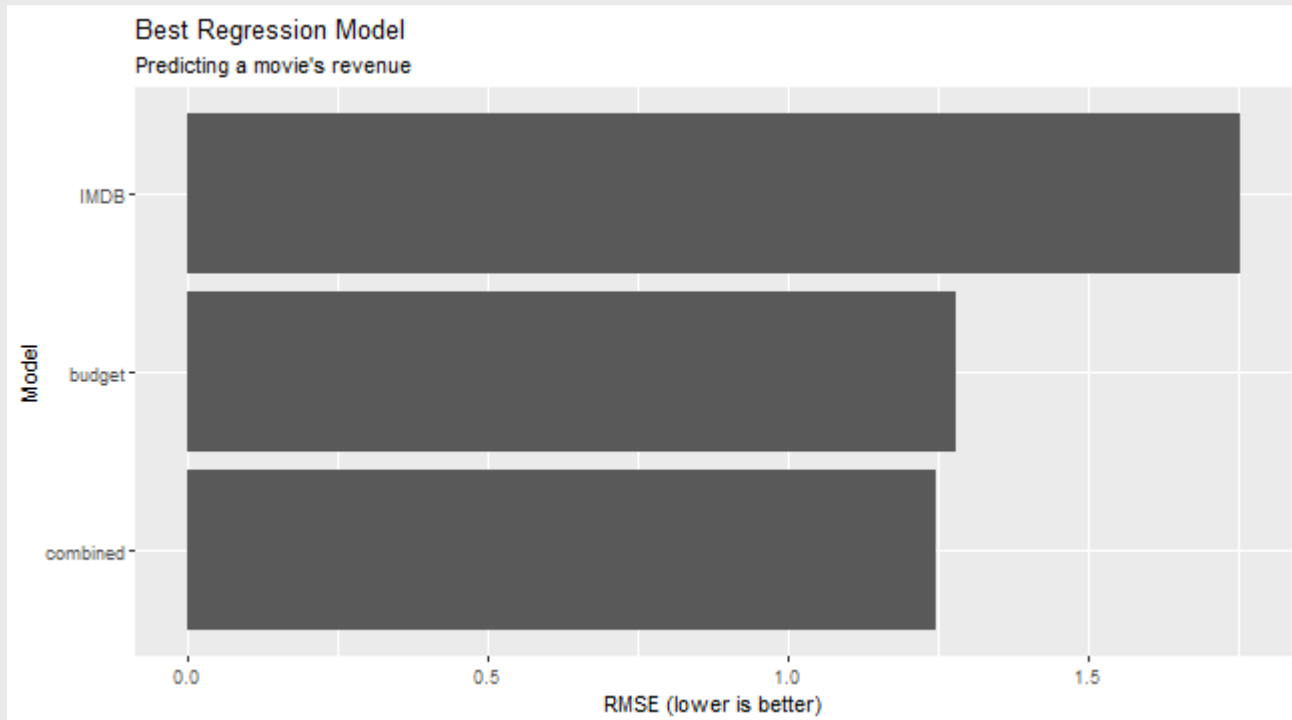
- Which model best predicts movie revenues?

```
p <- data.frame(budget = mean(rmseBudget),  
               IMDB = rmseScore,  
               combined = rmseBudgScore) %>%  
  gather(model,rmse) %>%  
  ggplot(aes(x = reorder(model,rmse),y = rmse)) +  
  geom_bar(stat = 'identity') +  
  labs(title = "Best Regression Model",  
       subtitle = "Predicting a movie's revenue",  
       y = "RMSE (lower is better)",  
       x = "Model") +  
  coord_flip()
```

# Comparing Models

- Which model best predicts movie revenues?

p



# Why RMSE?

- Want to understand how good / bad our model is
- Can use it to compare models



# Why RMSE?

- Do we improve our model with `score`?

```
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)

  mB <- lm(gross_log ~ budget_log,train)
  mS <- lm(gross_log ~ score,train)
  mC <- lm(gross_log ~ budget_log + score,train)

  bsRes <- test %>%
    mutate(pB = predict(mB,newdata = test),
           pS = predict(mS,newdata = test),
           pC = predict(mC,newdata = test)) %>%
    summarise(Budget = sqrt(mean((gross_log - pB)^2,na.rm=T)),
              Score = sqrt(mean((gross_log - pS)^2,na.rm=T)),
              Combined = sqrt(mean((gross_log - pC)^2,na.rm=T))) %>%
    bind_rows(bsRes)
}
```

# ASIDE: alternative code

- `sample_n()` and `anti_join()`

```
set.seed(123)
bsRes <- NULL
for(i in 1:100) {
  train <- mv %>%
    sample_n(size = round(nrow(.)*.8),replace = F)
  test <- mv %>%
    anti_join(train)

  mB <- lm(gross_log ~ budget_log,train)
  mS <- lm(gross_log ~ score,train)
  mC <- lm(gross_log ~ budget_log + score,train)

  bsRes <- test %>%
    mutate(pB = predict(mB,newdata = test),
           pS = predict(mS,newdata = test),
           pC = predict(mC,newdata = test)) %>%
    summarise(Budget = sqrt(mean((gross_log - pB)^2,na.rm=T)),
              Score = sqrt(mean((gross_log - pS)^2,na.rm=T)),
              Combined = sqrt(mean((gross_log - pC)^2,na.rm=T))) %>%
    bind_rows(bsRes)
}
```

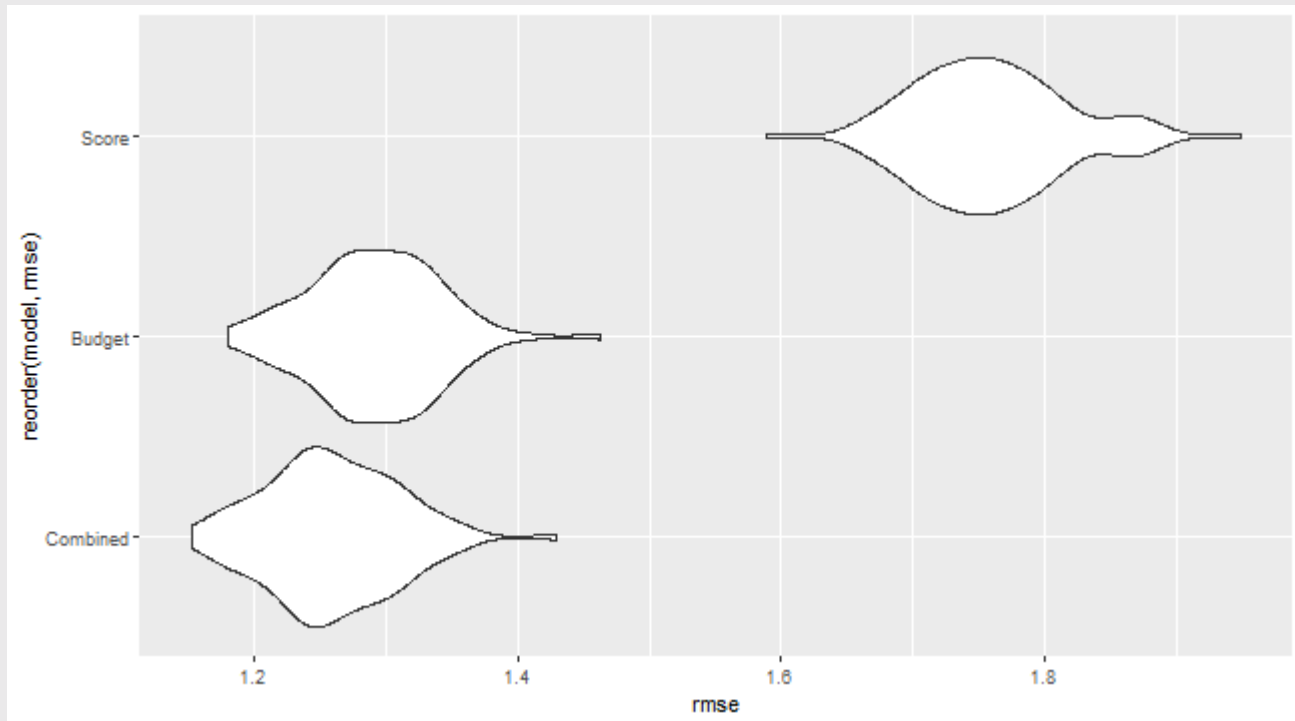
# Why RMSE?

```
bsRes %>%  
  summarise_all(mean, na.rm=T)
```

```
## # A tibble: 1 × 3  
##   Budget Score Combined  
##   <dbl> <dbl>    <dbl>  
## 1   1.29  1.76    1.26
```

# Visualizing

```
bsRes %>%  
  gather(model,rmse) %>%  
  ggplot(aes(x = rmse,y = reorder(model,rmse))) +  
  geom_violin()
```



# Categorical Data

- Thus far, only using continuous variables
- But we can do regression with categorical data too!
- The Bechdel Test: 3 questions of a movie
  1. Does it have two women in it?
  2. Who talk to each other?
  3. About something other than a man?

```
mv %>%  
  count(bechdel_score)
```

```
## # A tibble: 5 × 2  
##   bechdel_score      n  
##           <dbl> <int>  
## 1             0    141  
## 2             1    526  
## 3             2    206  
## 4             3   1185  
## 5            NA   1121
```

# Research Question

- Do movies that pass the Bechdel Test make more money?
  - **Theory:** Women are ~50% of the population. Movies that pass the test are more appealing to women.
  - **Hypothesis:** Movies that pass the test make more money.
- **Wrangling:** Let's turn the `bechdel_score` variable into a binary

```
mv <- mv %>%  
  mutate(bechdel_bin = ifelse(bechdel_score == 3,1,0)) %>%  
  mutate(bechdel_factor=recode_factor(bechdel_bin,  
                                       `1`="Pass",  
                                       `0`="Fail",  
                                       ))
```

# Regression

- We can add the binary factor to our regression

```
summary(lm(gross_log ~ bechdel_factor, mv))
```

```
##
## Call:
## lm(formula = gross_log ~ bechdel_factor, data = mv)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.8817 -0.7918  0.2253  1.0831  3.8225
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    18.16844    0.04835  375.794  <2e-16 ***
## bechdel_factorFail  0.15969    0.07423   2.151   0.0316 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.664 on 2056 degrees of freedom
## (1121 observations deleted due to missingness)
```

# Regression

- Coefficient is positive
- What is the interpretation?
  - Movies that fail make more money...
  - ...than what?
  - Movies that pass the Bechdel Test
- Categorical variables are **always interpreted in relation to the hold-out category!**



# Regression

- Movies that fail the test make more money!?
- **REMEMBER:** Correlation  $\neq$  causation
  - What might explain this pattern?
  - Budgets in a sexist Hollywood!
  - Movies that fail the test get larger budgets
  - Budgets are positively associated with gross
- So we want to "control" for budget by adding it to our regression

```
mBechCtrl <- lm(gross_log ~ budget_log + bechdel_factor, mv)
```

# Regression

```
summary(mBechCtrl)
```

```
##
## Call:
## lm(formula = gross_log ~ budget_log + bechdel_factor, data = mv)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.6325 -0.5305  0.1287  0.6792  7.9370
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.30814      0.34497   6.691 2.85e-11 ***
## budget_log        0.92089      0.01993  46.199 < 2e-16 ***
## bechdel_factorFail -0.18795      0.05254  -3.577 0.000356 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.166 on 2055 degrees of freedom
## (1121 observations deleted due to missingness)
## Multiple R-squared:  0.5106,    Adjusted R-squared:  0.5101
```

# Regression

- Our hypothesis is supported!
- What about non-binary categorical variables?

```
mv %>%  
  count(rating)
```

```
## # A tibble: 9 × 2  
##   rating      n  
##   <chr>    <int>  
## 1 G          54  
## 2 NC-17       6  
## 3 Not Rated   34  
## 4 PG        434  
## 5 PG-13     1249  
## 6 R        1388  
## 7 TV-MA       2  
## 8 Unrated     7  
## 9 <NA>        5
```

# Categorical

- Let's first remove rarely-occurring ratings

```
mvAnalysis <- mv %>%  
  filter(!rating %in% c('Approved', 'TV-14', 'TV-MA', 'TV-PG', 'X'))
```

# Categorical

```
summary(lm(gross_log ~ rating,mvAnalysis))
```

```
##
## Call:
## lm(formula = gross_log ~ rating, data = mvAnalysis)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.6749 -0.8189  0.1630  1.1082  5.2339
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    19.1818     0.2177   88.113 < 2e-16 ***
## ratingNC-17     -2.4483     0.6884   -3.556 0.000381 ***
## ratingNot Rated -4.4322     0.3502  -12.655 < 2e-16 ***
## ratingPG        -0.3905     0.2308   -1.692 0.090784 .
## ratingPG-13     -0.7633     0.2224   -3.433 0.000605 ***
## ratingR         -1.9123     0.2219   -8.618 < 2e-16 ***
## ratingUnrated  -4.6564     0.6426   -7.246 5.38e-13 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Categorical

- Everything makes less money than the hold-out category!
  - "G"-rated movies are powered by children
- What if we wanted to compare to a different reference category?

```
mvAnalysis <- mvAnalysis %>%  
  mutate(rating = factor(rating,  
                          levels = c('R', 'PG-13', 'PG', 'G', 'Not  
Rated')))  
  
mRating2 <- lm(gross_log ~ rating, mvAnalysis)
```

# Categorical

```
summary(mRating2)
```

```
##
## Call:
## lm(formula = gross_log ~ rating, data = mvAnalysis)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.6749 -0.8184  0.1610  1.1082  5.2339
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   17.26952    0.04296  402.005  <2e-16 ***
## ratingPG-13    1.14905    0.06242   18.408  <2e-16 ***
## ratingPG       1.52178    0.08802   17.289  <2e-16 ***
## ratingG        1.91231    0.22199    8.614  <2e-16 ***
## ratingNot Rated -2.51988    0.27782   -9.070  <2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.6 on 3154 degrees of freedom
```

# Cross Validation

- This is why `sample_n()` is useful

```
set.seed(123)
rmseRes_rating <- NULL
for(i in 1:100) {
  train <- mvAnalysis %>%
    group_by(rating) %>%
    sample_n(size = round(n()*.8),replace = F)
  test <- mvAnalysis %>% anti_join(train)

  m <- lm(gross_log ~ rating,train)
  rmseRes_rating <- test %>%
    mutate(preds = predict(m,newdata = test)) %>%
    summarise(rmse = sqrt(mean((gross_log - preds)^2,na.rm=T))) %>%
    bind_rows(rmseRes_rating)
}
```

```
## Joining with `by = join_by(title, rating, genre, year,
## released, score, votes, director, writer, star, country,
## budget, gross, company, runtime, id, imdb_id,
## bechdel_score, boxoffice_a, language, gross_log,
## budget_log, errors, bechdel_bin, bechdel_factor)`
```



# Quiz & Homework

- Go to Brightspace and take the **13th** quiz
  - The password to take the quiz is ####
- **Homework:**
  1. Study for midterm!