Problem Set 4

MGSC 310, Fall 2019, Professor Hersh (BEST PROFESSOR EVER!!!)

Geoffrey Hughes

9/27/2019

Question 1) Does Increasing a Movie's Budget Ever Pay Out?

a. & b. Import data, create new variables, filter, and split

```
library('tidyverse')
```

```
## -- Attaching packages ---------------------------------------------------------------------------

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ---------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
getwd()
```

```
## [1] "/Users/geoffreyhughes/Documents/MGSC_310/MGSC310/ProblemSets"
```

```
options(scipen = 10)
movies <- read.csv("/Users/geoffreyhughes/Documents/MGSC_310/MGSC310/Datasets/movie_metadata.csv")
set.seed(1861)


movies <- movies %>% filter(budget < 4e+08) %>% filter(content_rating !=
"", content_rating != "Not Rated") %>% drop_na(gross)
movies <- movies %>% mutate(genre_main = unlist(map(strsplit(as.character(movies$genres),
"\\|"), 1)), grossM = gross/1e+06, budgetM = budget/1e+06,
profitM = grossM - budgetM, rating_simple = fct_lump(content_rating,
n = 4), genre_main = factor(genre_main) %>% fct_drop())
set.seed(1861)
train_idx <- sample(1:nrow(movies), 0.8 * nrow(movies))
movies_train <- movies %>% slice(train_idx)
movies_test <- movies %>% slice(-train_idx)
```

c. Linear Regression Model

```r
mod_lm <- lm(grossM ~ imdb_score + budgetM,
             data = movies_train)

summary(mod_lm)
```

```
##
## Call:
## lm(formula = grossM ~ imdb_score + budgetM, data = movies_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -390.24  -26.12   -9.58   14.91  490.26
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -70.97634    5.88405  -12.06   <2e-16 ***
## imdb_score   13.05488    0.89545   14.58   <2e-16 ***
## budgetM       1.00460    0.02252   44.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.48 on 3026 degrees of freedom
## Multiple R-squared:  0.4272, Adjusted R-squared:  0.4268
## F-statistic:  1128 on 2 and 3026 DF,  p-value: < 2.2e-16
```

d. Interpretting the budgetM coefficient

- The budgetM coefficient is 1.0046 (magnitude), which means that **for every 1 unit change of budgetM, the movie's profitM will, on average, change by 1.0046** Since this coefficient is positive, that means that a positive change will elicit a positive change in profitM, and silimarly a negative change to budgetM will elicit a negative change in profitM (on average). So for every $1,000,000 more invested into a movie's budget, the profit will (on average) increase by $1,004,600.

e. Linear Regression model with added variable

```r
mod_lm2 <- lm(grossM ~ imdb_score + budgetM + I(budgetM^2),
             data = movies_train)

summary(mod_lm2)
```

```
##
## Call:
## lm(formula = grossM ~ imdb_score + budgetM + I(budgetM^2), data = movies_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -326.99  -25.78   -9.08   15.22  503.33
##
## Coefficients:
##                 Estimate  Std. Error t value Pr(>|t|)
## (Intercept)  -74.6517587   6.0512621 -12.337   <2e-16 ***
```

```
## imdb_score    13.2633770   0.8983249  14.765   <2e-16 ***
## budgetM        1.1277146   0.0530727  21.249   <2e-16 ***
## I(budgetM^2)  -0.0007161   0.0002796  -2.561   0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.43 on 3025 degrees of freedom
## Multiple R-squared:  0.4284, Adjusted R-squared:  0.4279
## F-statistic: 755.9 on 3 and 3025 DF,  p-value: < 2.2e-16
```

f. The budgetM and budgetM Squared Coefficients

- By use of polynomial regression, we now have not only budgetM, but budgetM Squared, to try to better fit our model to the training data. What we got as outcome, 1.1277146 as a coefficient for budgetM and -0.0007161 as a coefficient for budgetM Squared, shows that in this non-linear curve, we now have a function that looks like this: y_hat(profitM) = 13.2633(imdb_score) + 1.1277(budgetM) - 0.0007(budgetM)^2. These coefficients show that although this added term tries to create a parabolic line of best fit, the extremely small budgetM Squared coefficient show that there is not much change between models, and that the relationship between profitM and budgetM is mostly linear.
- It also means that with a negative budgetM Squared value, there are diminishing returns on increasing budgetM, since the parabola would be bent as if you hit it from the bottom right. (Starts out with a steeper slope, then flattens out a tad bit.)

g. Use margins to compare the relationship between profitM and budgetM at different budgetM levels

```r
library(margins)
margins(mod_lm2, at = list(budgetM = seq(25, 300, by = 5)))
```

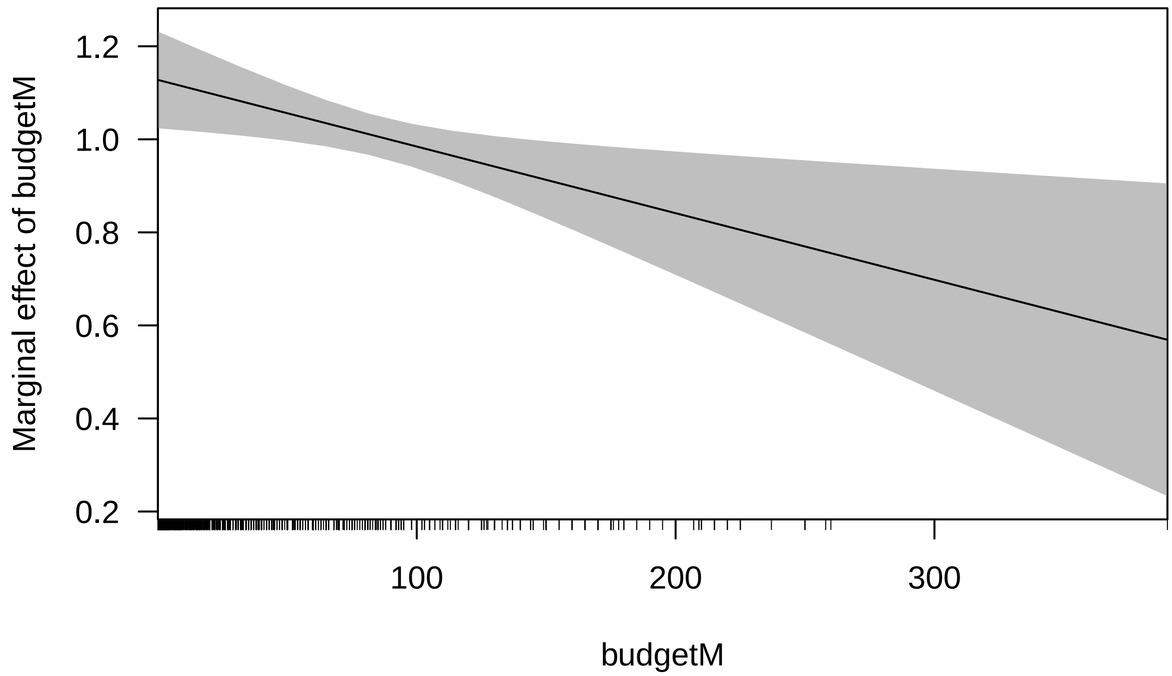
```
## Average marginal effects at specified values

## lm(formula = grossM ~ imdb_score + budgetM + I(budgetM^2), data = movies_train)

##  at(budgetM) imdb_score budgetM
##           25      13.26  1.0919
##           30      13.26  1.0847
##           35      13.26  1.0776
##           40      13.26  1.0704
##           45      13.26  1.0633
##           50      13.26  1.0561
##           55      13.26  1.0489
##           60      13.26  1.0418
##           65      13.26  1.0346
##           70      13.26  1.0275
##           75      13.26  1.0203
##           80      13.26  1.0131
##           85      13.26  1.0060
##           90      13.26  0.9988
##           95      13.26  0.9916
##          100      13.26  0.9845
##          105      13.26  0.9773
##          110      13.26  0.9702
##          115      13.26  0.9630
```

```
##          120      13.26  0.9558
##          125      13.26  0.9487
##          130      13.26  0.9415
##          135      13.26  0.9344
##          140      13.26  0.9272
##          145      13.26  0.9200
##          150      13.26  0.9129
##          155      13.26  0.9057
##          160      13.26  0.8986
##          165      13.26  0.8914
##          170      13.26  0.8842
##          175      13.26  0.8771
##          180      13.26  0.8699
##          185      13.26  0.8627
##          190      13.26  0.8556
##          195      13.26  0.8484
##          200      13.26  0.8413
##          205      13.26  0.8341
##          210      13.26  0.8269
##          215      13.26  0.8198
##          220      13.26  0.8126
##          225      13.26  0.8055
##          230      13.26  0.7983
##          235      13.26  0.7911
##          240      13.26  0.7840
##          245      13.26  0.7768
##          250      13.26  0.7696
##          255      13.26  0.7625
##          260      13.26  0.7553
##          265      13.26  0.7482
##          270      13.26  0.7410
##          275      13.26  0.7338
##          280      13.26  0.7267
##          285      13.26  0.7195
##          290      13.26  0.7124
##          295      13.26  0.7052
##          300      13.26  0.6980
```

- Given movies with 25, 50, 75, 90, 100, 200, and 300 million dollars in budget, **it only makes sense to increase movie budget for movies with a budgetM of 25, 50, or 75.**

h. *Extra Credit:* Cplot of marginal impact of an addictional dollar in budget for all levels of budget

```r
cplot(mod_lm2, x = "budgetM", what = "effect")
```



Question 2) Movie Residuals and Predicted Values

a. Linear Regression Model predicting for grossM using imdb_score, budgetM, the square of budgetM and rating_simple (Note: it says to use the movies data set and doesn't specify movies_train, so I used movies)

```
movies$rating_simple <- relevel(movies$rating_simple, ref = "R")
mod_lm3 <- lm(grossM ~ imdb_score + budgetM + I(budgetM^2) + rating_simple,
              data = movies)

summary(mod_lm3)
```

```
##
## Call:
## lm(formula = grossM ~ imdb_score + budgetM + I(budgetM^2) + rating_simple,
##     data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -340.12  -25.36   -8.00   16.05  497.24
##
## Coefficients:
##                       Estimate  Std. Error t value Pr(>|t|)
## (Intercept)        -89.0530642   5.5688288 -15.991  < 2e-16 ***
## imdb_score          14.5997095   0.8091740  18.043  < 2e-16 ***
## budgetM              0.9905262   0.0486393  20.365  < 2e-16 ***
## I(budgetM^2)        -0.0002396   0.0002493  -0.961    0.337
## rating_simpleG      28.3285399   5.5643903   5.091 3.73e-07 ***
## rating_simplePG     23.3174802   2.5701207   9.073  < 2e-16 ***
## rating_simplePG-13  15.6853673   1.9971896   7.854 5.22e-15 ***
```

```
## rating_simpleOther   1.1919369   6.6357311   0.180    0.857
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.44 on 3779 degrees of freedom
## Multiple R-squared:  0.465,  Adjusted R-squared:  0.464
## F-statistic: 469.2 on 7 and 3779 DF,  p-value: < 2.2e-16
```

b. Interpret the coefficient for rating_simple = G

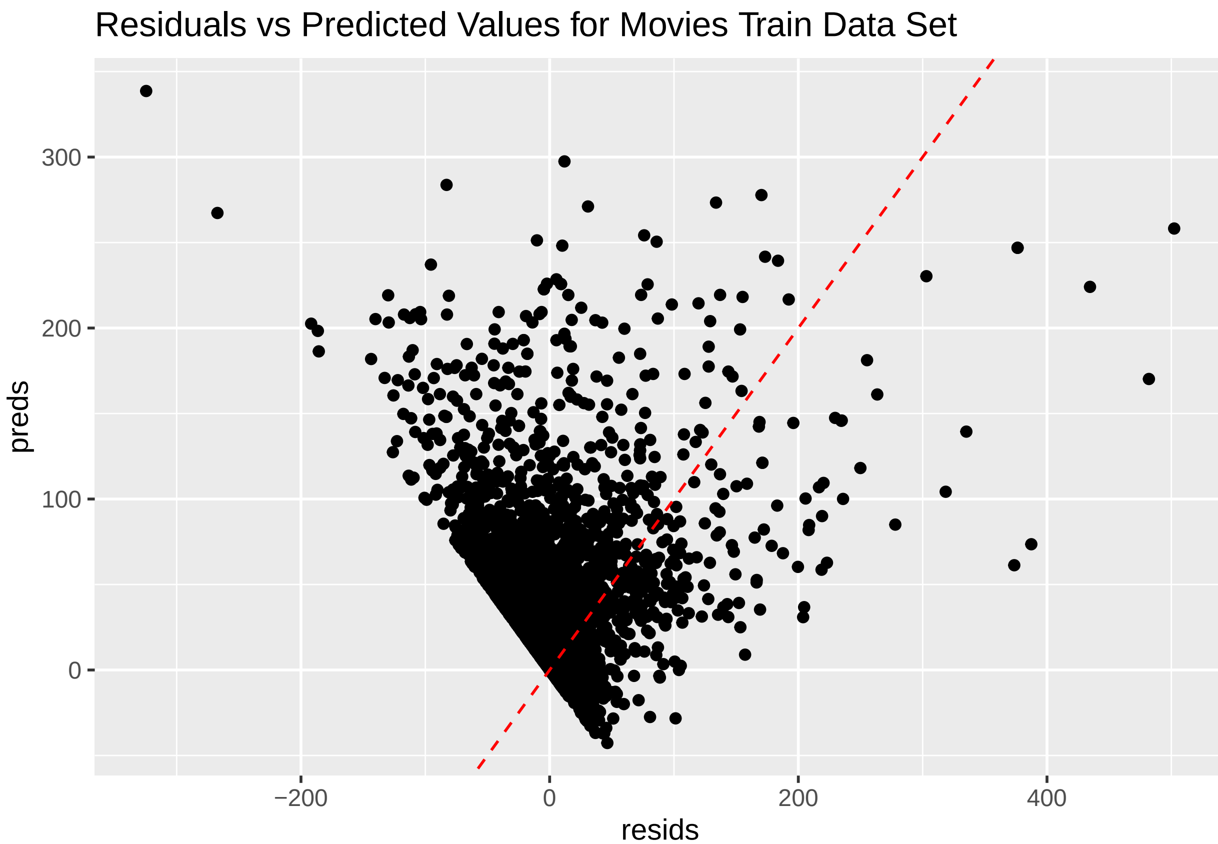
- The coefficient for a movie rated G is 28.32854, and since R is our base level, we can interpret this as such: if a movie were rated G, it would (on average) make 28.32854 million more in gross earnings than a movie rated R.

c. Use predict() to generate the predictions and residuals for both the movies_train and movies_test data sets

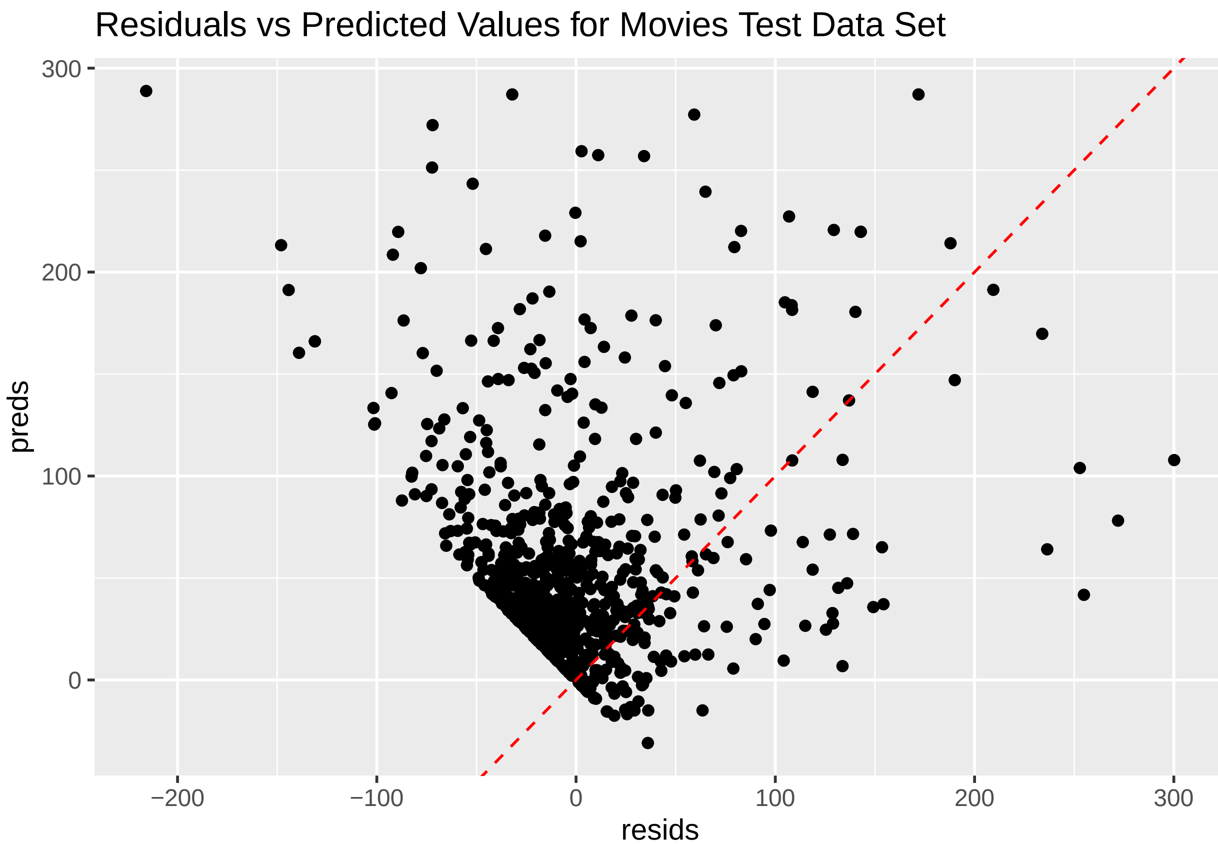
```
pred_movies_train <- predict(mod_lm3 <- lm(grossM ~ imdb_score + budgetM + I(budgetM^2) + rating_simple
                                           data = movies_train))
train_preds_DF <- data.frame(
  preds = pred_movies_train,
  resids = movies_train$grossM - predict(mod_lm3),
  resids2 = mod_lm3$residuals
)


pred_movies_test <- predict(mod_lm3 <- lm(grossM ~ imdb_score + budgetM + I(budgetM^2) + rating_simple,
                                           data = movies_test))
test_preds_DF <- data.frame(
  preds = pred_movies_test,
  resids = movies_test$grossM - predict(mod_lm3),
  resids2 = mod_lm3$residuals
)
```

d. Plot the residuals against the predicted values for both test and train data sets

```
ggplot(train_preds_DF, aes(x = resids, y = preds)) + geom_point() + geom_abline(intercept = 0, slope = 
```



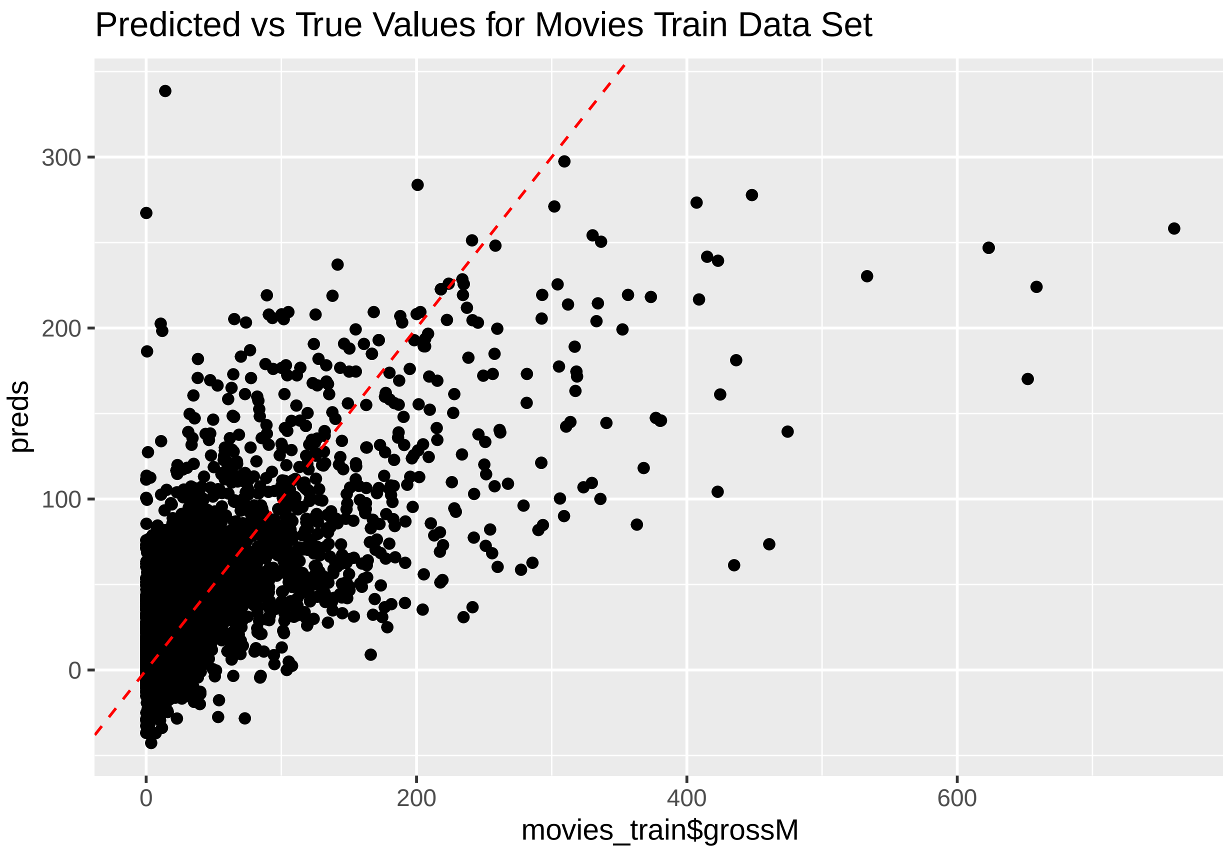

```
ggplot(test_preds_DF, aes(x = resids, y = preds)) + geom_point() + geom_abline(intercept = 0, slope = 1
```



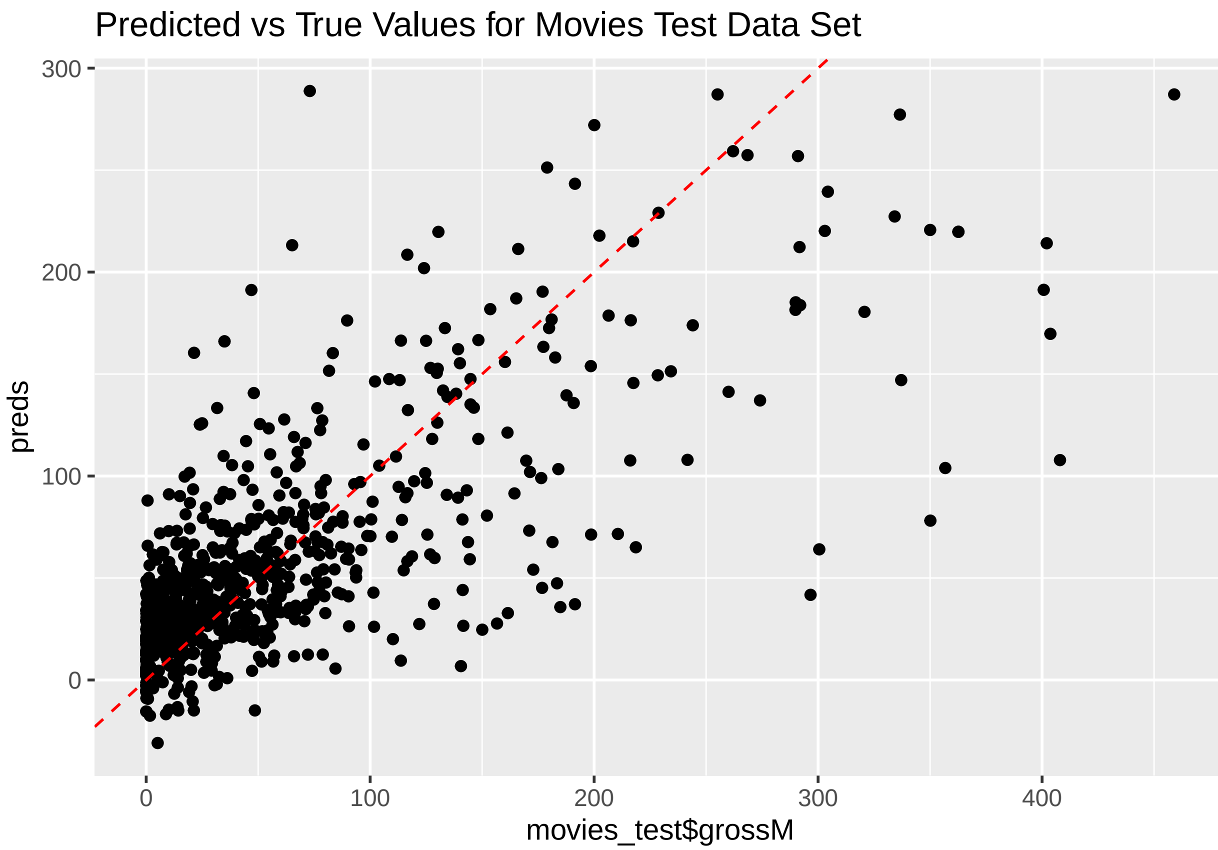
*
The errors in the train data set appear to be slightly heteroskedastic, sort of forming a trapezoidal shape. * Whereas the errors in the test data set are more homoskedastic, but this could be a product of having fewer values. * Overall, I'd say the error is **much more homoskedastic** (which is good!)

e. Plot predicted values vs true values for train and test data sets

```
ggplot(train_preds_DF, aes(x = movies_train$grossM, y = preds)) + geom_point() + geom_abline(intercept =
```




```
ggplot(test_preds_DF, aes(x = movies_test$grossM, y = preds)) + geom_point() + geom_abline(intercept = 0
```



f. In-Sample and Out-of-Sample R2 values; is our model overfit? How do we know?

```
library("caret")
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

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

```
RMSE <- function(t, p)
{
sqrt(sum(((t - p)^2)) * (1/length(t)))
}

train_RMSE <- RMSE(train_preds_DF$preds, movies_train$grossM)
train_RMSE
```

```
## [1] 51.51953
```

```r
test_RMSE <- RMSE(test_preds_DF$preds, movies_test$grossM)
test_RMSE
```

```
## [1] 50.60499
```

```r
postResample(pred = train_preds_DF$preds, obs = movies_train$grossM)
```

```
##       RMSE   Rsquared        MAE
## 51.5195279  0.4473679 32.8703193
```

```r
postResample(pred = test_preds_DF$preds, obs = movies_test$grossM)
```

```
##       RMSE   Rsquared        MAE
## 50.6049907  0.5280235 33.4096427
```

- Our function has an in-sample RMSE of 51.5195 and an R2 value of 0.4474, whereas our out-of-sample has an RMSE value of 50.605 and an R2 value of 0.528. So, since our Root Mean Squared Error is actually less in our test (out-of-sample) data set than our training (in-sample) data set, we can say that our model actually does a good job, and is **not overfit** to our train data set. Also, it is also important to note that the R2 value is higher in the test data set, which indicates that more of the sum of squares are explained by our regression model! If our RMSE was higher in our out-of-sample data, then we would probably be overfitting. Thanks, and goodnight!