Problem Set 5

Noah Estrada-Rand

10/1/2019

1

b) Running Given Instructions

```
movies <- movies[!is.na(movies$budget),]
movies <- movies[!is.na(movies$gross),]
movies <- movies[(movies$content_rating != "" & movies$content_rating != "Not Rated"), ]
movies <- movies[movies$budget<4e+8,]
movies$grossM <- movies$gross/1e+6
movies$budgetM <- movies$budget/1e+6
movies$profitM <- movies$grossM-movies$budgetM
movies$rating_simple <- fct_lump(movies$content_rating, n = 4)
set.seed(2019)
train_indx <- sample(1:nrow(movies), 0.8 * nrow(movies), replace=FALSE)
movies_train <- movies[train_indx, ]
movies_test <- movies[-train_indx, ]
```

c) Summary of the Linear Model Regressing GrossM by IMDB_score and BudgetM

```
mod.gross.score <- lm(grossM ~ imdb_score + budgetM, data = movies_train)
summary(mod.gross.score)
```

```
##
## Call:
## lm(formula = grossM ~ imdb_score + budgetM, data = movies_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -389.14  -26.24  -10.30   14.87  490.14
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -74.51571    6.03477  -12.35   <2e-16 ***
## imdb_score   13.59706    0.91594   14.85   <2e-16 ***
## budgetM       1.00195    0.02187   45.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.53 on 3026 degrees of freedom
## Multiple R-squared:  0.4373, Adjusted R-squared:  0.437
## F-statistic:  1176 on 2 and 3026 DF,  p-value: < 2.2e-16
```

d)

According to the coefficient outputs of the linear model regressing Gross revenue against IMDB scores and Budget, it appears that when holding IMDB scores fixed, increasing the budget has a net positive return for that particular movie. Thus, increasing spending by one million translates to an increase in gross revenue by $1,009,500.

e)

```r
mod.gross.score.budgetsquared <- lm(grossM~ imdb_score + budgetM + I(budgetM^2), data = movies_train)
summary(mod.gross.score.budgetsquared)
```

```
##
## Call:
## lm(formula = grossM ~ imdb_score + budgetM + I(budgetM^2), data = movies_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -321.12  -26.00   -9.78   15.26  503.61
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7.902e+01  6.220e+00  -12.70  < 2e-16 ***
## imdb_score    1.388e+01  9.197e-01   15.09  < 2e-16 ***
## budgetM       1.141e+00  5.231e-02   21.82  < 2e-16 ***
## I(budgetM^2) -7.864e-04  2.684e-04   -2.93  0.00341 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.47 on 3025 degrees of freedom
## Multiple R-squared:  0.4389, Adjusted R-squared:  0.4384
## F-statistic: 788.8 on 3 and 3025 DF,  p-value: < 2.2e-16
```

f) Marginal Effects Estimation of Budget Squared Model

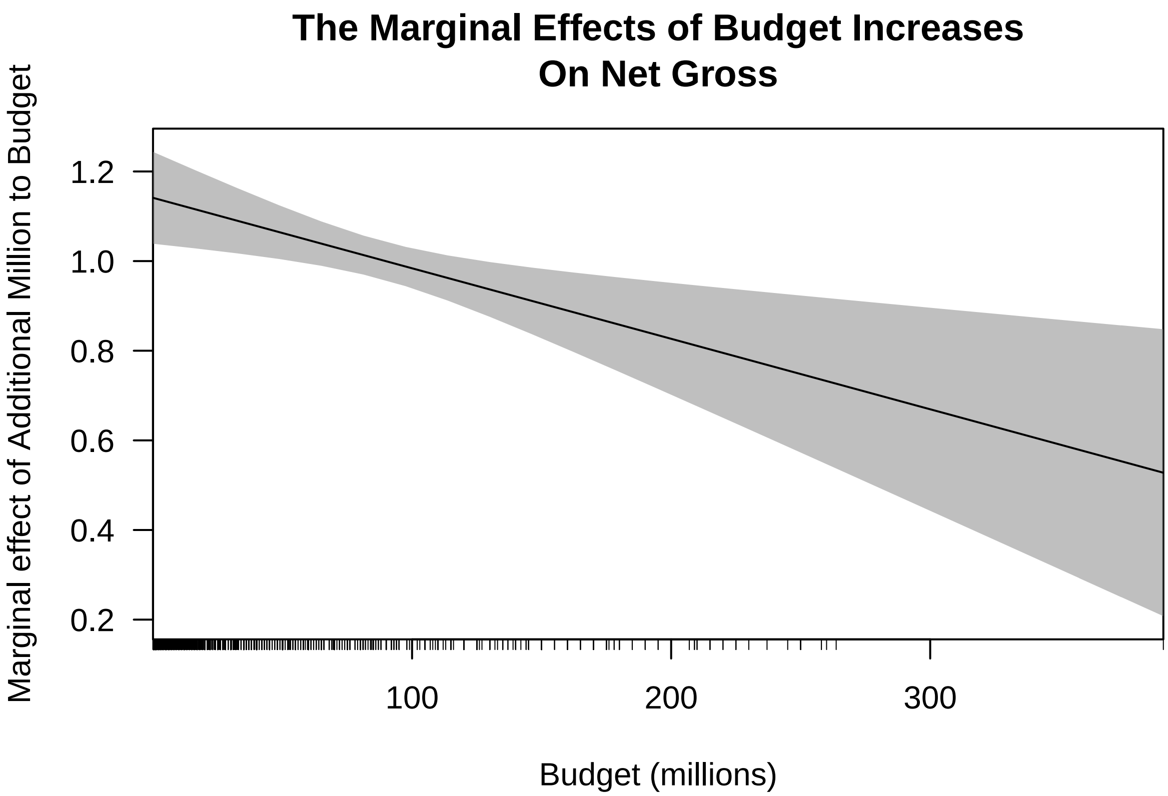
```r
margins(mod.gross.score.budgetsquared, at = list(budgetM = c(25, 50, 75, 90,
                                                             100, 200,300)))
```

```
## Average marginal effects at specified values

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

##  at(budgetM) imdb_score budgetM
##           25      13.88  1.1019
##           50      13.88  1.0626
##           75      13.88  1.0233
##           90      13.88  0.9997
##          100      13.88  0.9839
##          200      13.88  0.8267
##          300      13.88  0.6694
```

Looking at the results above, it becomes clear that it makes the most sense to increase your budget at lower budget levels to see a larger return. This effect dissipates, however, the higher your budget becomes. Thus, if a movie already has a large budget, the returns from an increase in budget will not be as significant, and thus not necessarily worth the investment.

g)

```
cplot(mod.gross.score.budgetsquared,"budgetM", what = "effect",
      main = "The Marginal Effects of Budget Increases\nOn Net Gross",
      xlab = "Budget (millions)",ylab = "Marginal effect of Additional Million to Budget")
```



2

a) Linear Model for Movie Gross Regressed by Budget, Budget Squared, and Ratings

```
mod3 <- lm(grossM ~ imdb_score + budgetM + I(budgetM^2) +
             relevel(rating_simple,ref = "R"), data = movies_train)
summary(mod3)
```

```
##
## Call:
## lm(formula = grossM ~ imdb_score + budgetM + I(budgetM^2) + relevel(rating_simple,
##     ref = "R"), data = movies_train)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -316.51  -25.36   -7.99   15.35  503.19 
## 
## Coefficients:
##                                         Estimate Std. Error t value
## (Intercept)                           -9.386e+01  6.374e+00 -14.725
## imdb_score                             1.519e+01  9.230e-01  16.459
## budgetM                                1.020e+00  5.367e-02  19.011
## I(budgetM^2)                          -4.646e-04  2.679e-04  -1.735
## relevel(rating_simple, ref = "R")G      3.325e+01  6.540e+00   5.084
## relevel(rating_simple, ref = "R")PG     2.331e+01  2.867e+00   8.130
## relevel(rating_simple, ref = "R")PG-13  1.542e+01  2.247e+00   6.864
## relevel(rating_simple, ref = "R")Other  6.901e+00  7.644e+00   0.903
##                                       Pr(>|t|)    
## (Intercept)                            < 2e-16 ***
## imdb_score                             < 2e-16 ***
## budgetM                                < 2e-16 ***
## I(budgetM^2)                            0.0829 .  
## relevel(rating_simple, ref = "R")G     3.92e-07 ***
## relevel(rating_simple, ref = "R")PG    6.17e-16 ***
## relevel(rating_simple, ref = "R")PG-13 8.12e-12 ***
## relevel(rating_simple, ref = "R")Other   0.3667    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 51.69 on 3021 degrees of freedom
## Multiple R-squared:  0.4561, Adjusted R-squared:  0.4548 
## F-statistic: 361.8 on 7 and 3021 DF,  p-value: < 2.2e-16
```

b)

When considering the coefficient for the G-rating on movies, it appears that G rated movies gross \$33,250,000.00 more than R-rated movies.

c) Predicted Values for Test and Training Sets

```r
trainingPredictions <- predict(mod3)
testPredictions <- predict(mod3,movies_test)
head(trainingPredictions)
```

```
##      1381      1780      4295      4677      3926      3177 
##  81.638927  43.965497 -20.509034  11.734680  25.678336   8.972807
```

```r
head(testPredictions)
```

```
##       24       27       32       33       44       45 
## 247.6437 224.0332 217.9563 216.4370 250.9750 207.3216
```

d) The Residuals for Both the Test and Training Data are Shown Below

```
trainingResids <- movies_train$grossM - trainingPredictions
testResids <- movies_test$grossM - testPredictions
head(trainingResids)
```

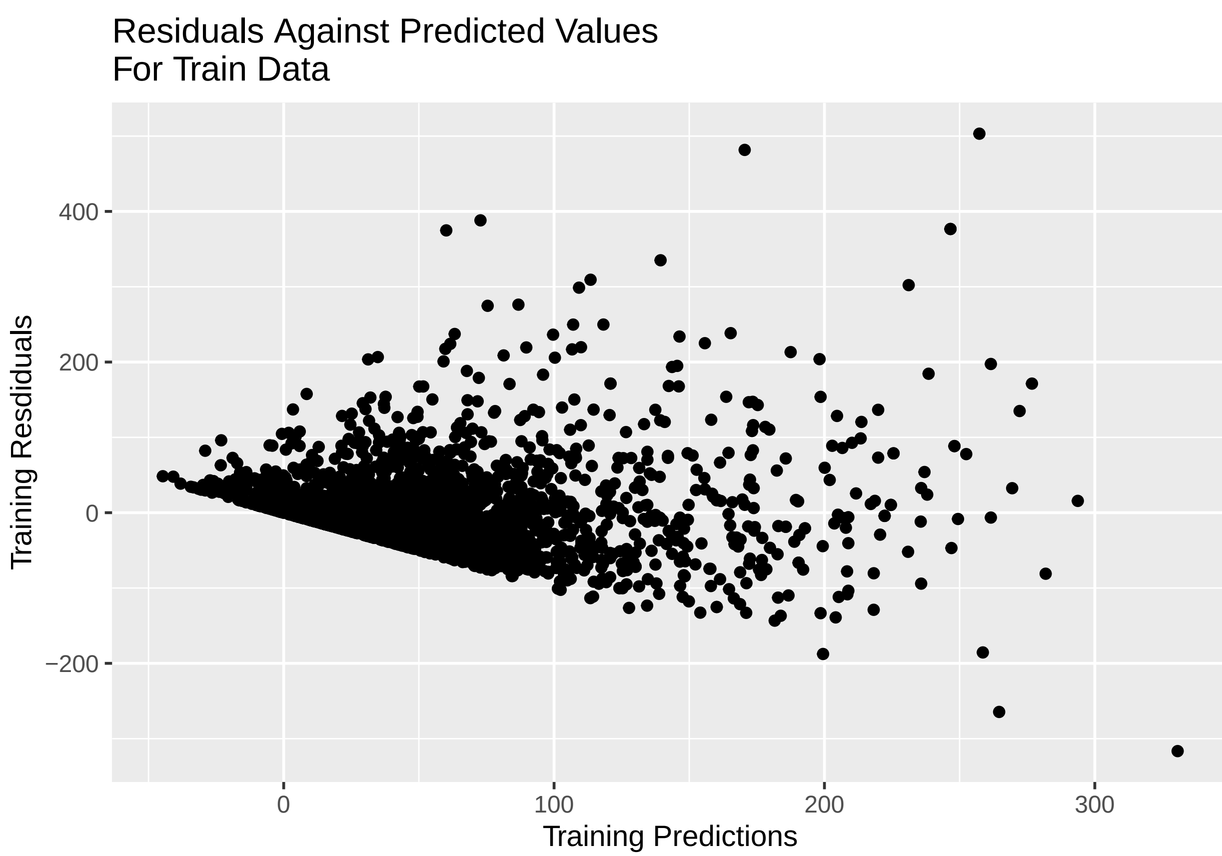
```
##        1381       1780       4295       4677       3926       3177
## -13.430737 -13.277133  21.044283  -6.215762 -22.785754  -2.130749
```

```
head(testResids)
```

```
##         24        27        32        33        44        45
##   10.71165 434.63906 155.42161 192.55523 164.00952 -82.00159
```

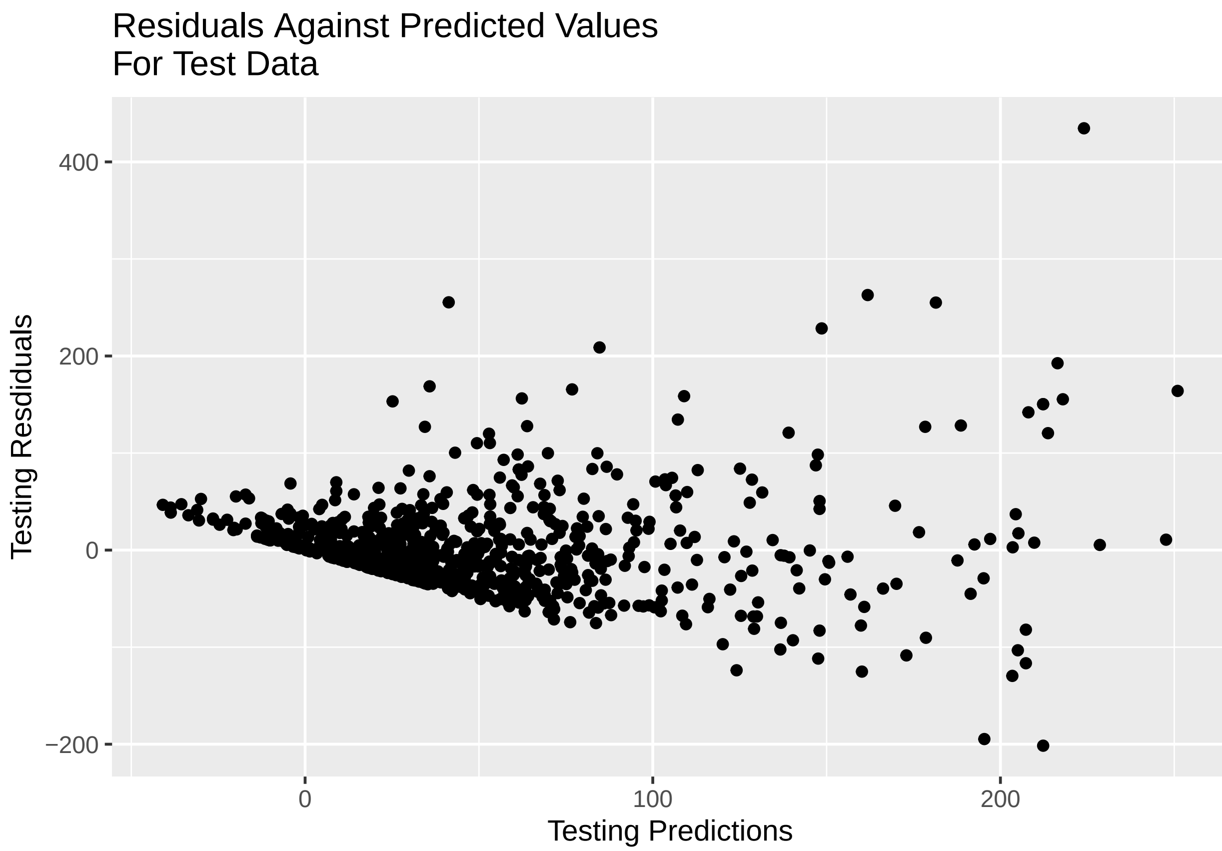
e)

```
dummySet <- data.frame(trainingResids = trainingResids,
                       trainingPredictions = trainingPredictions,
                       movies_train)
ggplot(dummySet,aes(x = trainingPredictions,y = trainingResids)) + geom_point()+
   labs(title = "Residuals Against Predicted Values\nFor Train Data",
        y = "Training Resdiduals",x = "Training Predictions")
```




```
dummySet2 <- data.frame(testingResids = testResids,
                        testingPredictions = testPredictions,
```

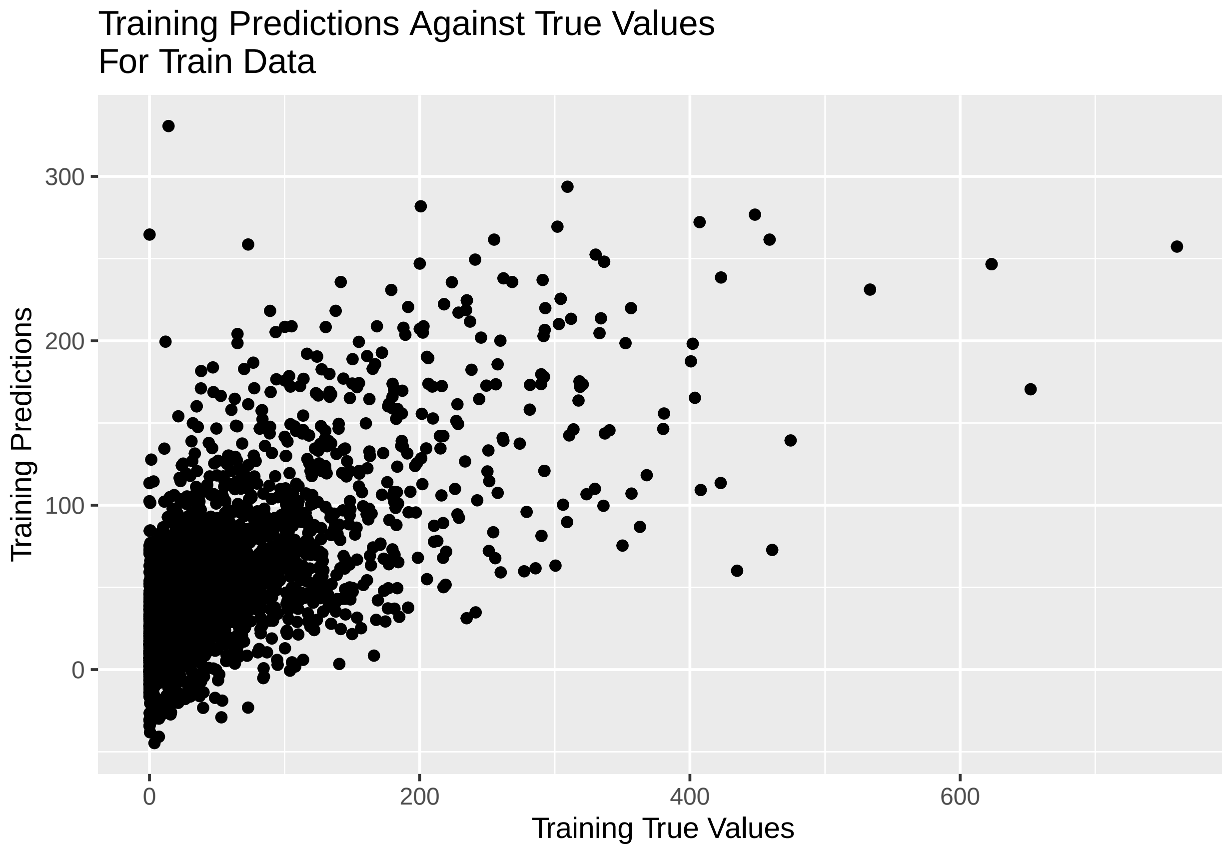
```r
                    movies_test)
ggplot(dummySet2,aes(x = testingPredictions,y = testingResids)) + geom_point()+
   labs(title = "Residuals Against Predicted Values\nFor Test Data",
        y = "Testing Resdiduals",x = "Testing Predictions")
```



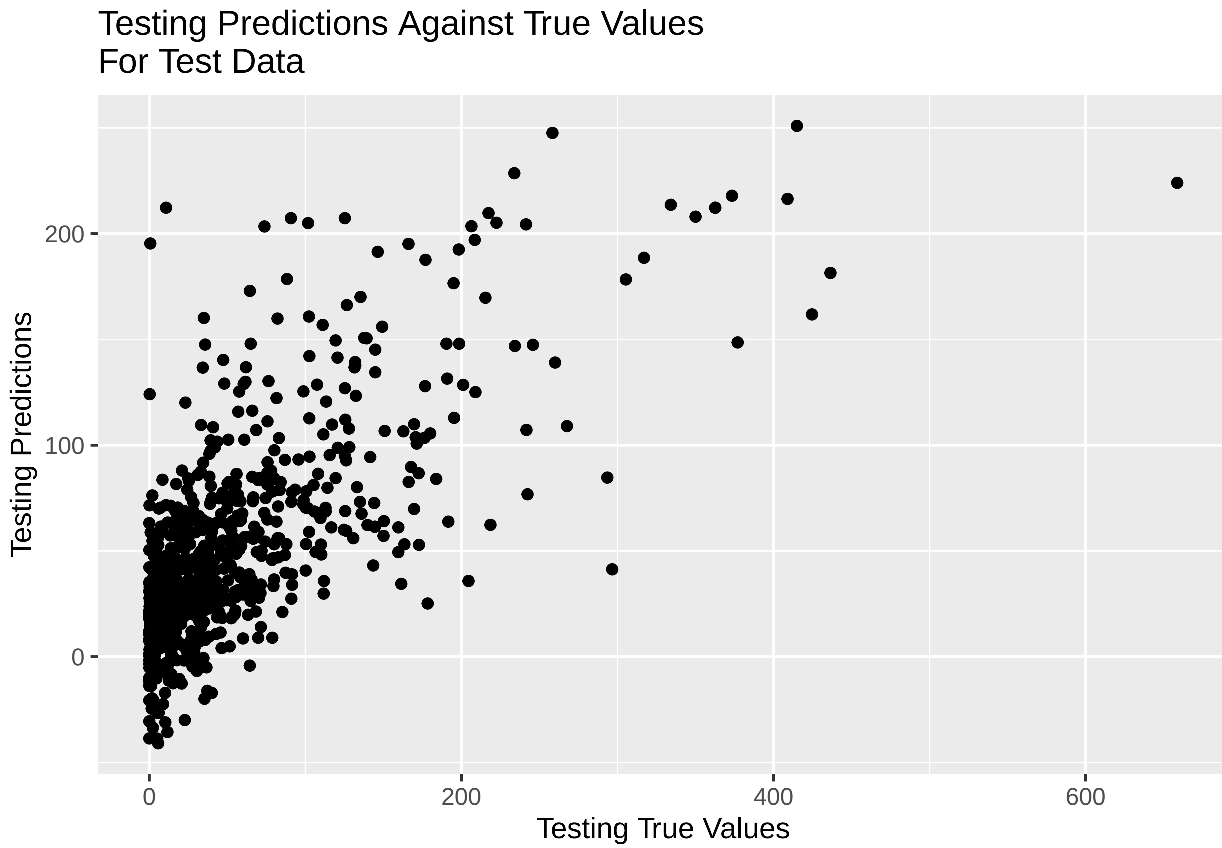
From the above plots, it becomes clear that both sets of residuals and predicted values are heteroskedastic as their residuals vary widely as the predicted values change. Plus the residuals do not appear to be equally distributed above and below zero, further indicative of heteroskedasticity.

f)

```r
ggplot(dummySet,aes(x = grossM,y = trainingPredictions)) + geom_point()+
   labs(title = "Training Predictions Against True Values\nFor Train Data",
        y = "Training Predictions",x = "Training True Values")
```




```r
ggplot(dummySet2,aes(x = grossM,y = testingPredictions)) + geom_point()+
  labs(title = "Testing Predictions Against True Values\nFor Test Data",
       y = "Testing Predictions",x = "Testing True Values")
```



g) Examining RMSE of both Test and Train Sets

```
RMSE <- function(t, p) {
   sqrt(sum(((t - p)^2)) * (1/length(t)))
}
rmse.train <-RMSE(movies_train$grossM,trainingPredictions)
rmse.test <-RMSE(movies_test$grossM,testPredictions)
accuracy(trainingPredictions,movies_train$grossM)
```

```
##                      ME     RMSE      MAE       MPE     MAPE
## Test set -5.687971e-13 51.62507 32.99394 -3680.763 7029.888
```

```
accuracy(testPredictions,movies_test$grossM)
```

```
##                ME     RMSE      MAE     MPE     MAPE
## Test set 2.352148 50.54496 33.27347 2243.78 7009.265
```

```
print(rmse.train)
```

```
## [1] 51.62507
```

```
print(rmse.test)
```

```
## [1] 50.54496
```

Based on the root mean standard error caclulated for both the training and testing data, it becomes apparent that the model is not overfit. This is due to the fact that the root mean squared error for both sets are similar, indicative of similar performance for both datasets. In the event that the test data had much larger root mean squared error than the training data, it would have been safe to say that the model was overfit to the training data. In this instance, however, this is not the case.