ProblemSet5

David Aarhus

3/15/2020

Question 1

a)

```
movies <- read.csv("/Users/DavidAarhus/Documents/310 R/Datasets/movie_metadata.csv")
```

loads dataset

b)

```
#install.packages("tidyverse")
library("tidyverse")
```

```
## -- Attaching packages ----------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

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

```
#removing missing values of budget and gross
movies <- movies[!is.na(movies$budget),]
movies <- movies[!is.na(movies$gross),]
# removing empty content rating or not rated
movies <- movies[(movies$content_rating != "" & movies$content_rating != "Not Rated"), ] # removing mov
movies <- movies[movies$budget<4e+8,]
# simplifying variables
movies$grossM <- movies$gross/1e+6
movies$budgetM <- movies$budget/1e+6
movies$profitM <- movies$grossM-movies$budgetM
# creating new column `rating_simple` using `fct_lump` (from `tidyverse` package) # to pick 4 major lev
movies$rating_simple <- fct_lump(movies$content_rating, n = 4)
# creating train and test sets
set.seed(310)
train_indx <- sample(1:nrow(movies), 0.8 * nrow(movies), replace=FALSE)
movies_train <- movies[train_indx, ]
movies_test <- movies[-train_indx, ]
```

c)

```
# creates a linear model using the movies_train dataset, to predict grossM
model <- lm(grossM ~ imdb_score + budgetM, movies_train)
```

```r
# prints summary of model
summary(model)
```

```
##
## Call:
## lm(formula = grossM ~ imdb_score + budgetM, data = movies_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -403.15  -26.68   -9.59   16.19  481.60
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -75.50080    6.12502  -12.33   <2e-16 ***
## imdb_score   13.70041    0.93185   14.70   <2e-16 ***
## budgetM       1.03872    0.02235   46.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53.86 on 3026 degrees of freedom
## Multiple R-squared:  0.4464, Adjusted R-squared:  0.446
## F-statistic:  1220 on 2 and 3026 DF,  p-value: < 2.2e-16
```

d) The coefficent of budgetM shows that for every unit increase of budgetM there is a 1.03872 unit increase in grossM.

e)

```r
model2 <- lm(grossM ~ imdb_score + budgetM + I(budgetM^2), movies_train)
summary(model2) # prints off summary for model2
```

```
##
## Call:
## lm(formula = grossM ~ imdb_score + budgetM + I(budgetM^2), data = movies_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -350.10  -26.41   -9.43   16.03  492.38
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7.874e+01  6.301e+00 -12.497   <2e-16 ***
## imdb_score    1.389e+01  9.353e-01  14.849   <2e-16 ***
## budgetM       1.144e+00  5.349e-02  21.394   <2e-16 ***
## I(budgetM^2) -6.060e-04  2.791e-04  -2.171     0.03 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53.83 on 3025 degrees of freedom
## Multiple R-squared:  0.4472, Adjusted R-squared:  0.4467
## F-statistic: 815.8 on 3 and 3025 DF,  p-value: < 2.2e-16
```

f)

```r
library("margins")
m <- margins(model2, at = list(budgetM = c(25,50,75,90,100,200,300)))
```

```
m
```

```
## Average marginal effects at specified values

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

##  at(budgetM) imdb_score budgetM
##           25      13.89  1.1139
##           50      13.89  1.0836
##           75      13.89  1.0533
##           90      13.89  1.0352
##          100      13.89  1.0230
##          200      13.89  0.9019
##          300      13.89  0.7807
```

ANSWER: this figure shows that we have a diminishing return on investment for increasing our budget marginally. For a budget less than about 100 million, we'll have a good return on our investment and earn more money if we increase our budget. After about 100 million (when our marginal impact dips below 1, we'll be losing money on our investment and it doesn't make sense to increase our budget anymore.

Question 2

a)

```
model3 <- lm(grossM ~ imdb_score
             + budgetM
             + I(budgetM^2)
             + relevel(rating_simple, ref = "R"), movies_train)
summary(model3)
```

```
##
## Call:
## lm(formula = grossM ~ imdb_score + budgetM + I(budgetM^2) + relevel(rating_simple,
##     ref = "R"), data = movies_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -344.81  -26.27   -8.08   16.63  491.67
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          -9.517e+01  6.471e+00 -14.706  < 2e-16
## imdb_score                            1.543e+01  9.396e-01  16.424  < 2e-16
## budgetM                               1.012e+00  5.486e-02  18.444  < 2e-16
## I(budgetM^2)                         -2.585e-04  2.783e-04  -0.929    0.353
## relevel(rating_simple, ref = "R")G    2.976e+01  6.410e+00   4.643 3.58e-06
## relevel(rating_simple, ref = "R")PG   2.437e+01  2.966e+00   8.217 3.05e-16
## relevel(rating_simple, ref = "R")PG-13 1.678e+01  2.307e+00   7.275 4.38e-13
## relevel(rating_simple, ref = "R")Other 3.897e+00  7.757e+00   0.502    0.615
##
## (Intercept)                          ***
## imdb_score                           ***
## budgetM                              ***
## I(budgetM^2)
## relevel(rating_simple, ref = "R")G   ***
## relevel(rating_simple, ref = "R")PG  ***
```

```
## relevel(rating_simple, ref = "R")PG-13 ***
## relevel(rating_simple, ref = "R")Other
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53.03 on 3021 degrees of freedom
## Multiple R-squared:  0.4642, Adjusted R-squared:  0.463
## F-statistic:   374 on 7 and 3021 DF,  p-value: < 2.2e-16
```

b) ANSWER: a movie with a G rating, holding budget and IMDB score fixed, will earn $29M more in gross.

c)

```
preds_train1 <- predict(model3, newdata = movies_train)
preds_test1 <- predict(model3, newdata = movies_test)
```

d)

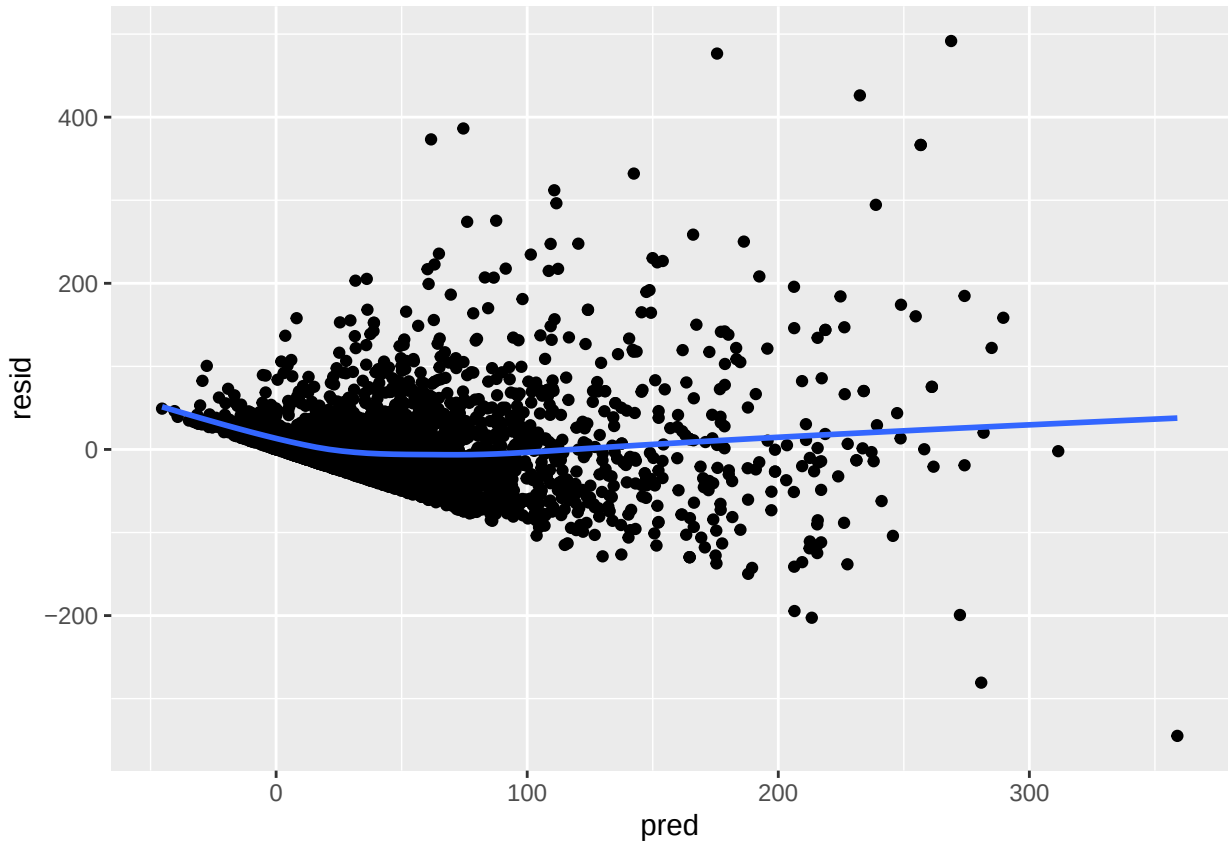
```
preds_train1_df <- data.frame(true = movies_train$grossM,
                              pred = predict(model3, newdata = movies_train),
                              resid = movies_train$grossM - predict(model3, newdata = movies_train))

preds_test1_df <- data.frame(true = movies_test$grossM,
                              pred = predict(model3, newdata = movies_test),
                              resid = movies_test$grossM - predict(model3, newdata = movies_test))
```

e)

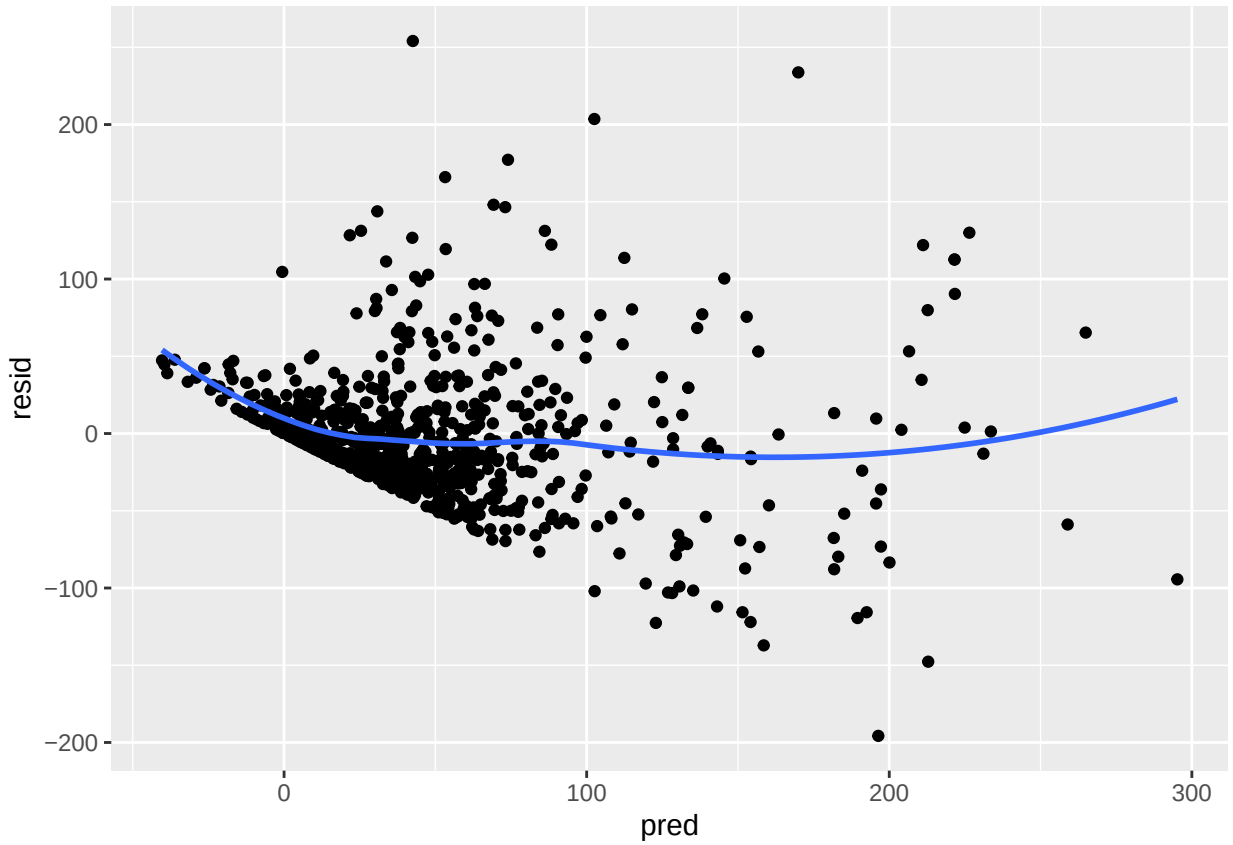
```
# heteroskedasticity - variance of error
library(ggplot2)
# visualize distribution of errors of residuals over prediction
ggplot(preds_train1_df, aes(x=pred, y=resid)) +
  geom_point() +
  geom_smooth(se=FALSE)
```

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




```
ggplot(preds_test1_df, aes(x=pred, y=resid)) +
  geom_point() +
  geom_smooth(se=FALSE)
```

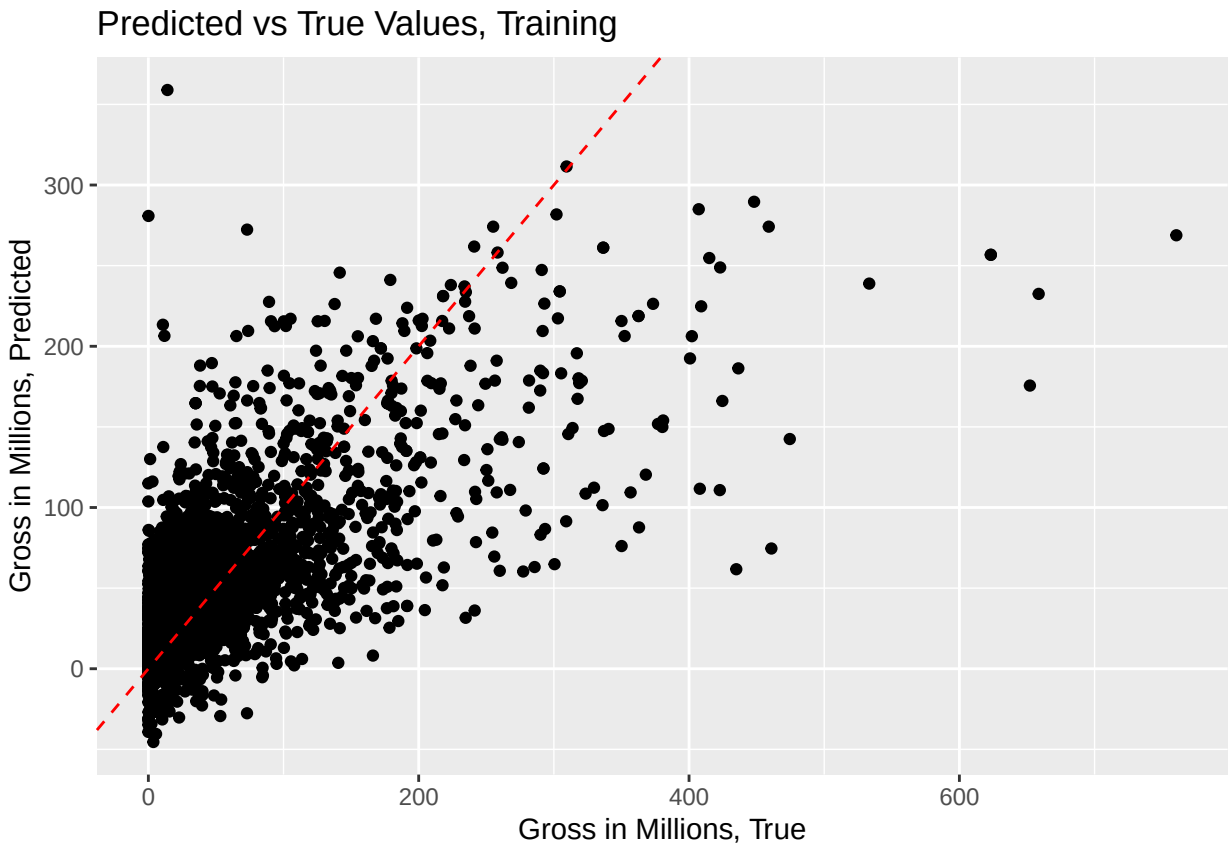
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Both Training and Test sets appear to have heteroskedasticity graphs.

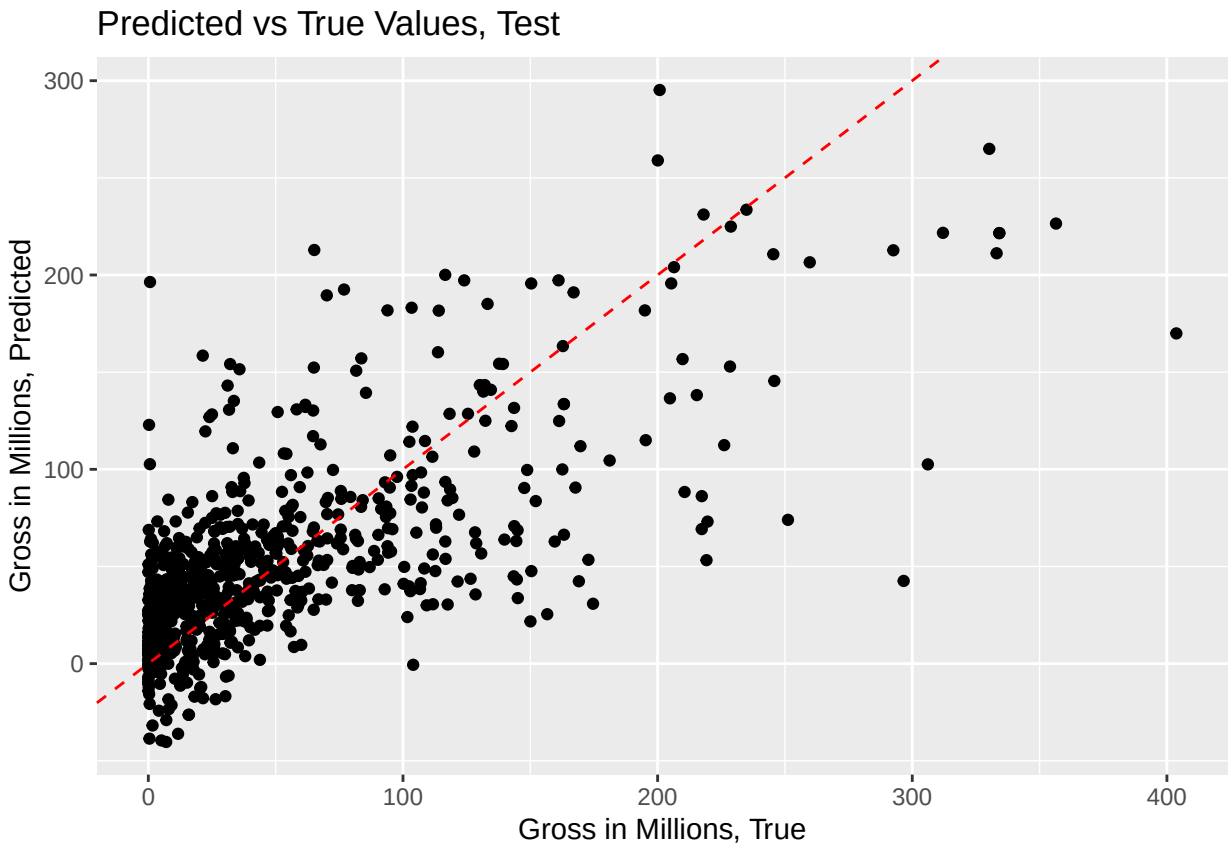
f) Training Set

```
ggplot(preds_train1_df, aes(x = true, y = pred)) +
  geom_point() +
  labs(x = "Gross in Millions, True",
       y = "Gross in Millions, Predicted",
       title = "Predicted vs True Values, Training") +
  geom_abline(intercept = 0, slope = 1,
              color = "red", linetype = "dashed")
```



Test Set

```
ggplot(preds_test1_df, aes(x = true, y = pred)) +
  geom_point() +
  labs(x = "Gross in Millions, True",
       y = "Gross in Millions, Predicted",
       title = "Predicted vs True Values, Test")  +
  geom_abline(intercept = 0, slope = 1,
              color = "red", linetype = "dashed")
```



g)

```
# using caret package
library(caret)
```

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

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

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

```
# training error (root mean squared error)
RMSE(preds_train1_df$pred, preds_train1_df$true)
```

```
## [1] 52.96042
```

```
rmse_train <-   sqrt(mean((preds_train1_df$true - preds_train1_df$pred)^2))
rmse_train
```

```
## [1] 52.96042
```

```
# test error
RMSE(preds_test1_df$pred, preds_test1_df$true)
```

```
## [1] 44.68023
```

```
rmse_test <- sqrt(mean((preds_test1_df$true - preds_test1_df$pred)^2))
rmse_test
```

```
## [1] 44.68023
```

ANSWER: Model is underfit because RMSE in the test set is lower than training RMSE.