PS7

William L. Guzman

February 24, 2017

PS7: Resample Nonlinear

Part 1: Joe Biden (redux) [4 points]

1. Estimate the training MSE of the model using the traditional approach. Fit the linear regression model using the entire dataset and calculate the mean squared error for the training set.

```
#Get Libraries
library (dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.3.2
```

```
library(modelr)
```

```
## Warning: package 'modelr' was built under R version 3.3.2
```

```
library(broom)
```

```
##
## Attaching package: 'broom'
```

```
## The following object is masked from 'package:modelr':
##
##     bootstrap
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.3.2
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.3.2
```

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

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.3.2
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(purrr)
```

```
## Warning: package 'purrr' was built under R version 3.3.2
```

```
##
## Attaching package: 'purrr'
```

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

```
## The following objects are masked from 'package:dplyr':
##
##     contains, order_by
```

```
library(splines)
library(gam)
```

```
## Warning: package 'gam' was built under R version 3.3.2
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.3.2
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loaded gam 1.14
```

```
theme_set(theme_minimal())

#Get the Data
filePath <- ("C:/Users/Walle/Desktop/Winter Quarter (2016-2017)/MACS 30100/persp-model/students/
GuzmanDaughertyW/PS7/Data")

bidenDat <-read.csv(file=paste(filePath,"biden.csv",sep="/"))

#Part 1:
bidenLM <- lm(biden ~. , data = bidenDat)

#Get the MSE of the linear model.
mseModel <- function(sm)
  mean(sm$residuals^2)

(bidenLMMSE <- mseModel(bidenLM))
```

```
## [1] 395.2702
```

2. Estimate the test MSE of the model using the validation set approach.

How does this value compare to the training MSE from step 1?

The MSE value from the complete data set is 395. After using only the training data set, the MSE is 393. We had a difference of 2 unit.

```
#Split the sample in 2 (Validation Set): training set (70%) and a validation set (30%)
sampleSize <- floor(nrow(bidenDat)*0.70)

set.seed(1234)

datIndex <- sample(seq_len(nrow(bidenDat)), size = sampleSize)

bidenDataTrain <-  bidenDat[datIndex, ]
bidenDataTest  <-  bidenDat[-datIndex, ]

#Fit the linear regression model using only the training observations.
bidenLMTrain <- lm(biden ~. , data = bidenDataTrain)
bidenLMTest  <- lm(biden~., data = bidenDataTest)

#Calculate the MSE using only the test set observations.
(mseModel(bidenLM))
```

```
## [1] 395.2702
```

```
(mseModel(bidenLMTest))
```

```
## [1] 393.2954
```

3. Repeat the validation set approach 100 times, using 100 different splits of the observations into a training set and a validation set. Comment on the results obtained.

After doing repeating the setps 100 times, we can see that the MSE for the train data goes from 370 to 422 and the MSE of the Test data goes from 328 to 446. Compare to the MSE of the whole data set (395), we can see that the Test data has more variability on the results.

```r
#Making the 100 validation test
n <- 100

mseMatrix <- matrix(data=NA,nrow=100,ncol=2)

colnames(mseMatrix) <- c("TrainMSE", "TestMSE")

set.seed(1234)

for(n in 1:100)
{
  datIndex <- sample(seq_len(nrow(bidenDat)), size = sampleSize)

  bidenDataTrain <-  bidenDat[datIndex, ]
  bidenDataTest  <-  bidenDat[-datIndex, ]

  bidenLMTrain <- lm(biden ~. , data = bidenDataTrain)
  bidenLMTest  <- lm(biden~.,   data = bidenDataTest)

  mseMatrix[n,1] <- (mseModel(bidenLMTrain))
  mseMatrix[n,2] <- (mseModel(bidenLMTest))

}

head(mseMatrix)
```

```
##      TrainMSE  TestMSE
## [1,] 394.3264 393.2954
## [2,] 377.6952 431.1090
## [3,] 390.5892 403.4440
## [4,] 389.6681 406.5784
## [5,] 405.0373 370.6707
## [6,] 401.6274 376.4766
```

```r
summary(mseMatrix)
```

```
##     TrainMSE        TestMSE
##  Min.   :370.0   Min.   :328.3
##  1st Qu.:388.4   1st Qu.:375.1
##  Median :394.4   Median :393.0
##  Mean   :394.8   Mean   :392.2
##  3rd Qu.:401.8   3rd Qu.:407.1
##  Max.   :422.3   Max.   :446.3
```

4. Estimate the test MSE of the model using the leave-one-out cross-validation (LOOCV) approach. Comment on the results obtained.

```
#LOOCV Approach
#loocv_data <- crossv_kfold(bidenDat, k = nrow(bidenDat))
#loocv_models <- map(loocv_data$train, ~ lm(biden ~ age + female + educ + dem + rep, data = .))
#loocv_mse <- map2_dbl(loocv_models, loocv_data$test, mse)
#loocv_mean_mse <- mean(loocv_mse)
```

5. Estimate the test MSE of the model using the 10-fold cross-validation approach. Comment on the results obtained.

```
#cv10Data   <- crossv_kfold(bidenDat, k = 10)
#cv10Models <- map(cv10Data$train, ~ lm(biden ~ age + female + educ + dem + rep, data = .))
#cv10MSE    <- map2_dbl(cv10Models, cv10Data$test, mse)
#tenFold_mean_mse <- mean(tenFold_mse)

# mseFoldCal <- function(i) {
#   tenFold_data <- crossv_kfold(data, k = 10)
#   tenFold_models <- map(tenFold_data$train, ~ lm(biden ~ age + female + educ + dem + rep, data
 = .))
#   tenFold_mse <- map2_dbl(tenFold_models, tenFold_data$test, mse)
#   tenFold_mean_mse <- mean(tenFold_mse)
# }
```

6. Repeat the 10-fold cross-validation approach 100 times, using 100 different splits of the observations into 10-folds. Comment on the results obtained.

```
#
#set.seed(1234)
#cv10df <- data.frame(index = 1:100)
#cv10df$mse <- unlist(lapply(cv10df$index, mseFoldCal))
```

7. Compare the estimated parameters and standard errors from the original model in step 1 (the model estimated using all of the available data) to parameters and standard errors estimated using the bootstrap ($n = 1000$).

```
#biden_boot <- biden %>%
#  modelr::bootstrap(1000) %>%
#  mutate(model = map(bidenDat, ~ lm(biden ~ age + female + educ + dem + rep, data = bidenDat)),
#         coef = map(model, tidy))

#biden_boot %>%
#  unnest(coef) %>%
#  group_by(term) %>%
#  summarize(est.boot = mean(estimate),
#            se.boot = sd(estimate, na.rm = TRUE))
```

Part 2: College (bivariate) [3 points]

1. Explore the bivariate relationships between some of the available predictors and Outstate.

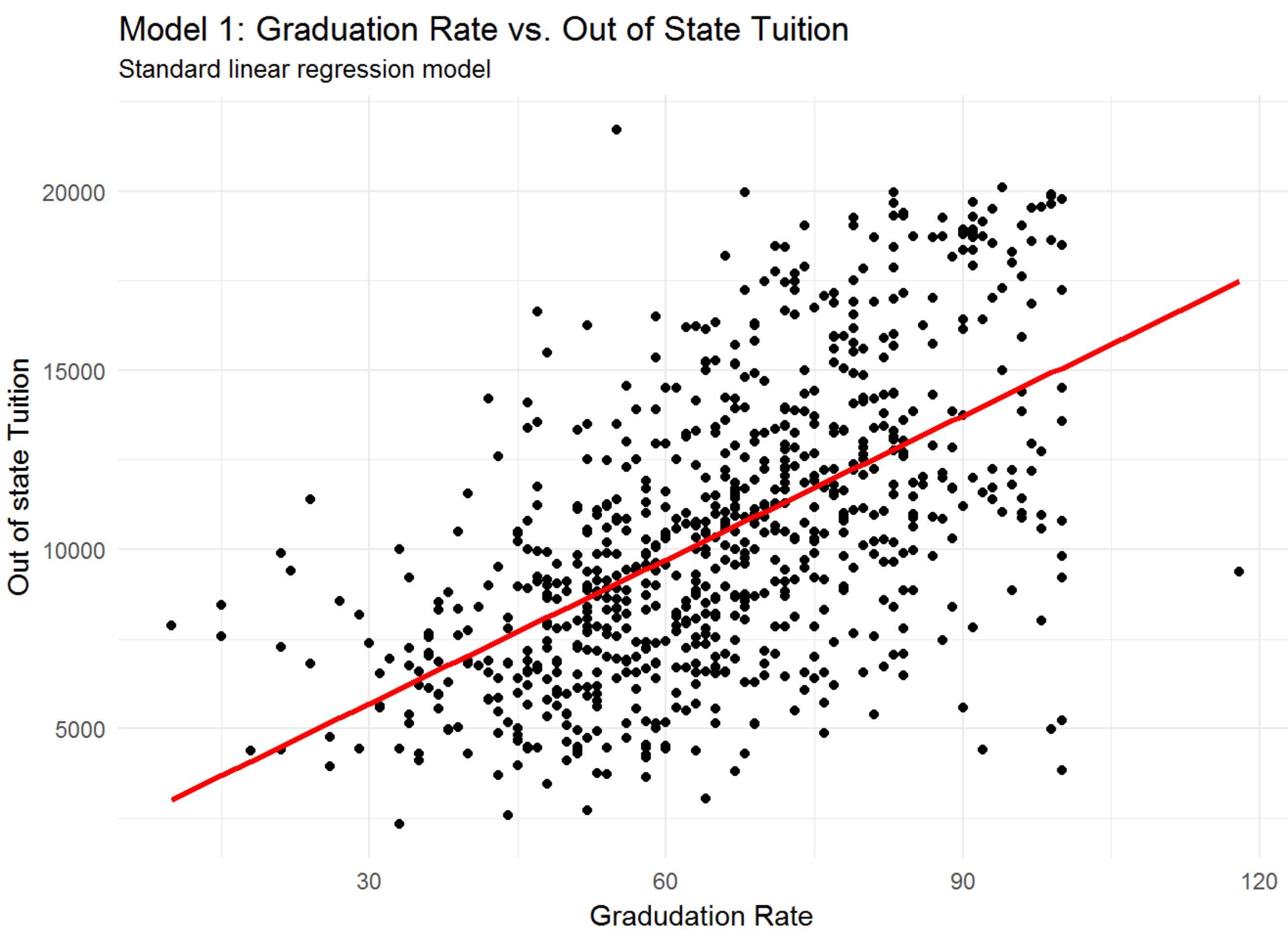
Model 1: Outstate ~ Grad.Rate

First we will create a linear model of the Oustate variable with the predictor variable, Grad.Rate. The purpose is to see if there exist a linear regression between these two variables. We can plot the data do have a visual idea of our model. After creating the model, we can see that our p-value is only .326. We can tell that Grad.Rate is not a good predictor variable.
After seeing the cross validation model, we have that the MSE of the model is 1,0888,407. We now perform a log transformation to see if our model can be improve. We can see that the Log transformation make the model worse, lowering the p-value to 0.284. The MSE for the model is now 11,567,662. We can see is off from our initial MSE. If we see our plot, we can see there is some kind of linear regression, but after doing the model, we can see that there is no relationship between Grad.Rate and Outstate.

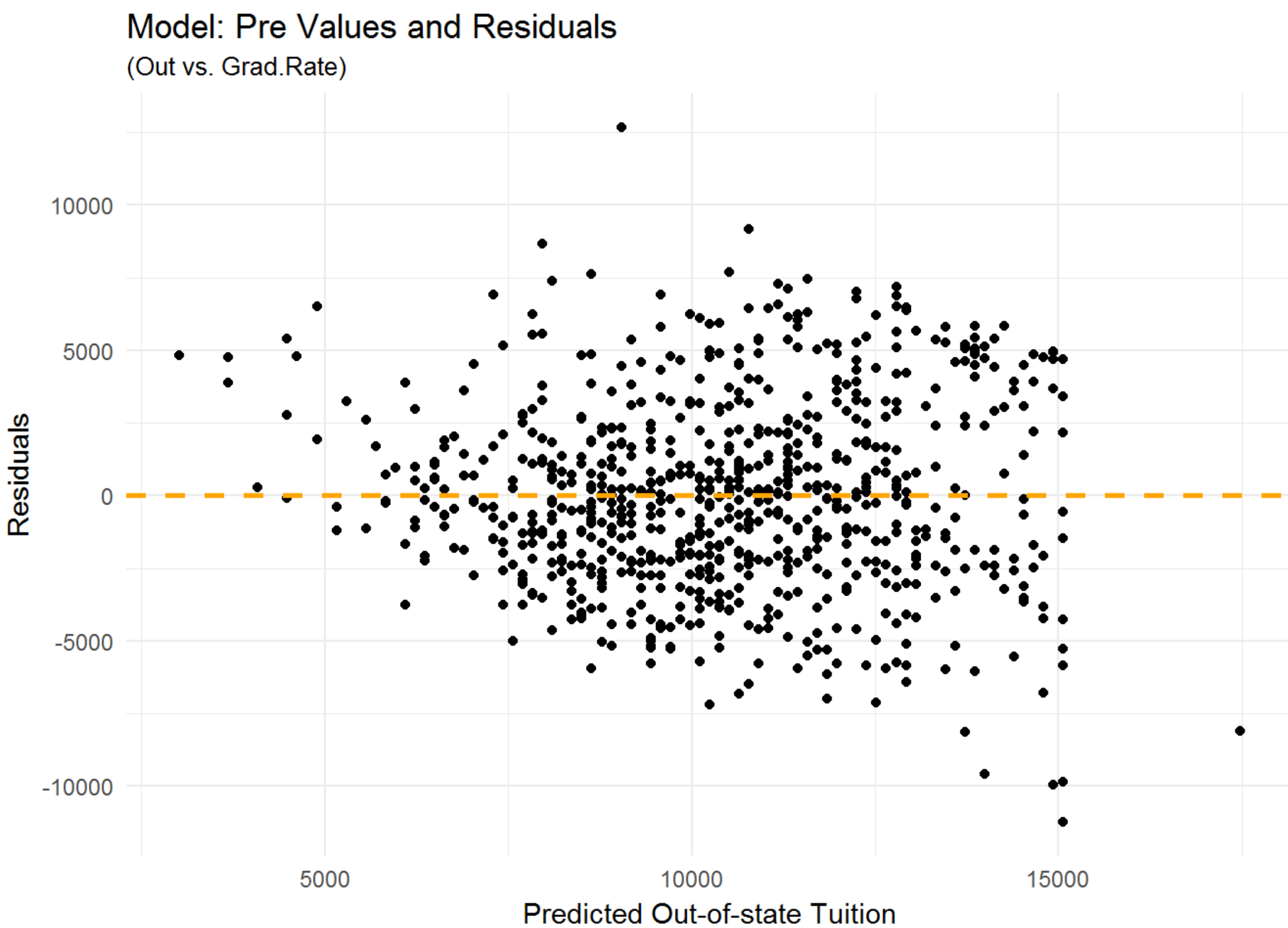
```
#Create 3 simplear model Regression

#Get the Data
collegeDat <-read.csv(file=paste(filePath,"college.csv",sep="/"))

#Create the linear model: Oustate ~ Grad.Rate
collegeLM1 <- lm(Outstate~Grad.Rate,data = collegeDat)

summary(collegeLM1)
```

```
##
## Call:
## lm(formula = Outstate ~ Grad.Rate, data = collegeDat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11221.5  -2251.7   -161.3   2156.5  12659.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1681.939    467.289   3.599 0.000339 ***
## Grad.Rate    133.796      6.905  19.377  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3304 on 775 degrees of freedom
## Multiple R-squared:  0.3264, Adjusted R-squared:  0.3255
## F-statistic: 375.5 on 1 and 775 DF,  p-value: < 2.2e-16
```

```
#Create the residuals and predictors of the linear model
collegeDat %>%
  add_predictions(collegeLM1) %>%
  add_residuals(collegeLM1) %>%
  {.} -> mod1grid

#Plot the data
collegeLM1Plot <- ggplot(aes(Grad.Rate, Outstate), data = collegeDat) +
  geom_point() +
  geom_line(aes(y=pred), data = mod1grid, color = 'red', size = 1) +
  labs(title = "Model 1: Graduation Rate vs. Out of State Tuition",
       subtitle = "Standard linear regression model",
       x = "Gradudation Rate ",
       y = "Out of state Tuition")

#plot the Data for the residual
collegeLM1RESPlot <- ggplot(mod1grid, aes(x = pred)) +
  geom_point(aes(y = resid)) +
  geom_hline(yintercept = 0, color = 'orange', size = 1, linetype = 'dashed') +
  labs(title = "Model: Pre Values and Residuals",
       subtitle = "(Out vs. Grad.Rate)",
        x = "Predicted Out-of-state Tuition",
        y = "Residuals")

collegeLM1Plot
```



```
collegeLM1RESPlot
```



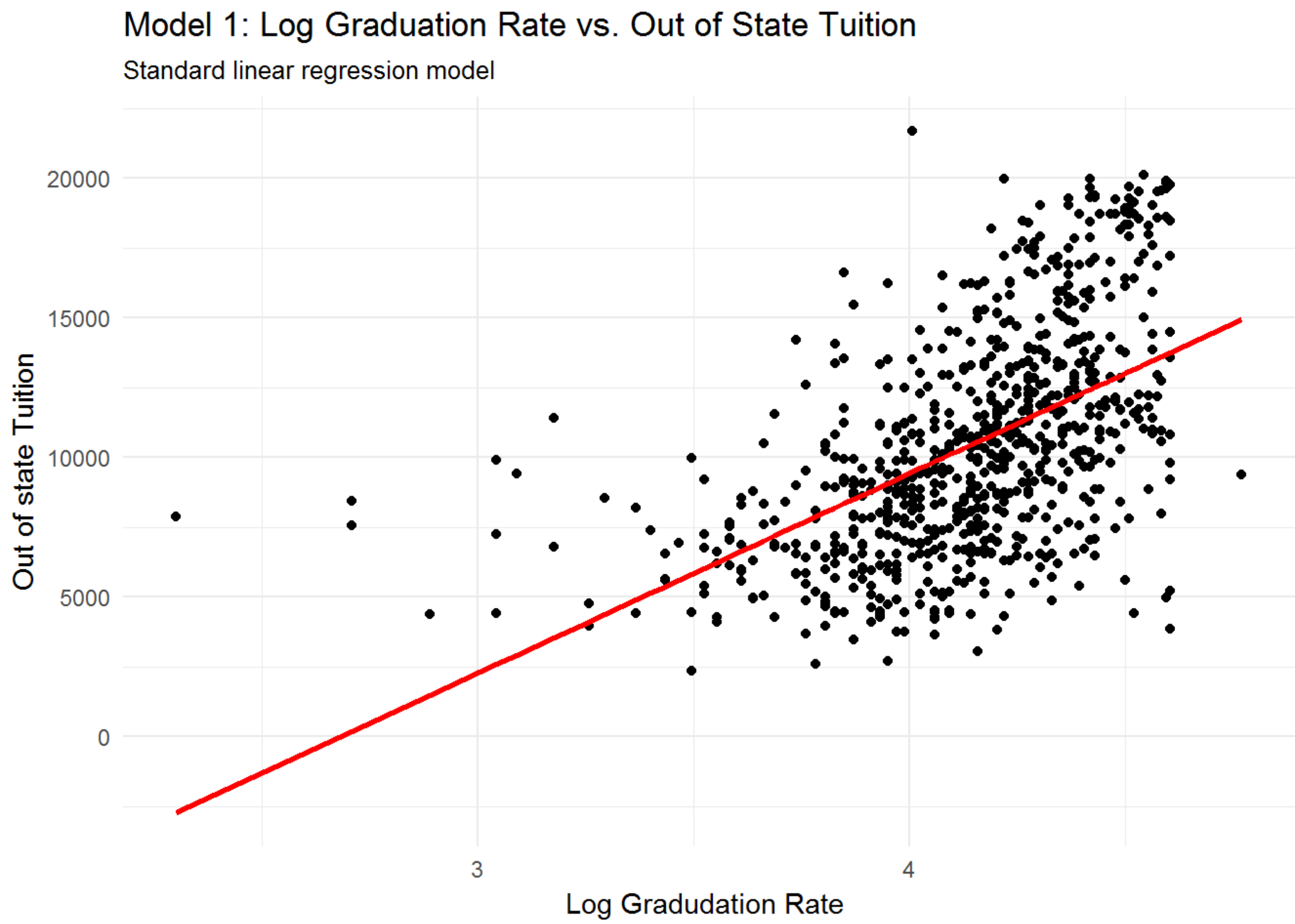

```
#Using the cross validation for a simple linear model

# Calculate 10-fold cV for the standard linear regression
(gradRateMSE <- mseModel(collegeLM1))
```

```
## [1] 10888407
```

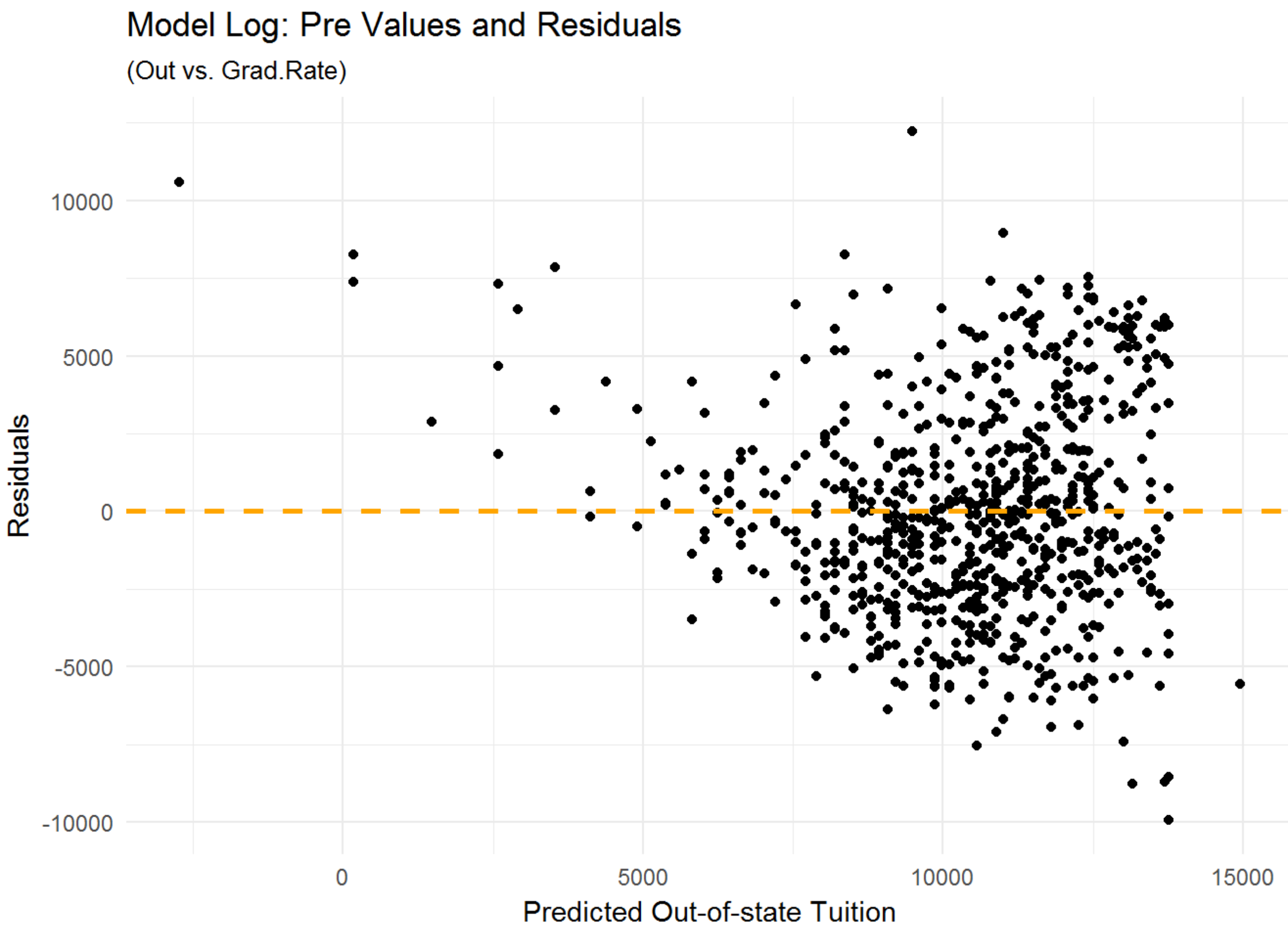
```
#Log transformation
collegeLM1LOG <- lm(Outstate ~ log(Grad.Rate), data = collegeDat)

summary(collegeLM1LOG)
```

```
##
## Call:
## lm(formula = Outstate ~ log(Grad.Rate), data = collegeDat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9921.9 -2379.0  -236.2  1993.2 12219.6
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -19218.5     1694.6  -11.34   <2e-16 ***
## log(Grad.Rate)   7161.6      408.1   17.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3406 on 775 degrees of freedom
## Multiple R-squared:  0.2843, Adjusted R-squared:  0.2834
## F-statistic: 307.9 on 1 and 775 DF,  p-value: < 2.2e-16
```

```
#Create the residuals and predictors of the linear model
collegeDat %>%
  add_predictions(collegeLM1LOG) %>%
  add_residuals(collegeLM1LOG) %>%
  {.} -> mod1gridLOG

#Plot the data
collegeLM1LOGPlotLOG <- ggplot(aes(log(Grad.Rate), Outstate), data = collegeDat) +
  geom_point() +
  geom_line(aes(y=pred), data = mod1gridLOG, color = 'red', size = 1) +
  labs(title = "Model 1: Log Graduation Rate vs. Out of State Tuition",
       subtitle = "Standard linear regression model",
       x = "Log Gradudation Rate ",
       y = "Out of state Tuition")

#plot the Data for the residual
collegeLM1RESPlotLOG <- ggplot(mod1gridLOG, aes(x = pred)) +
  geom_point(aes(y = resid)) +
  geom_hline(yintercept = 0, color = 'orange', size = 1, linetype = 'dashed') +
  labs(title = "Model Log: Pre Values and Residuals",
       subtitle = "(Out vs. Grad.Rate)",
        x = "Predicted Out-of-state Tuition",
        y = "Residuals")

collegeLM1LOGPlotLOG
```




```
collegeLM1RESPlotLOG
```




```
#Find the MSE of the model Log
(gradRateMSELOG <- mseModel(collegeLM1LOG))
```

```
## [1] 11567662
```

Model 2: Outstate ~ Expend

In this example, we have Oustate vs Expend and also Outstate vs Log(Expend). This time, we can see that our model improve after transforming the Expend variable with log. We can say that the variable has a no-linear relationship with Outstate, by improving the RSquared by more than 10%.

```
#The normal Regression
expendLM <- lm(Outstate ~ Expend, data = collegeDat)

summary(expendLM)
```

```
##
## Call:
## lm(formula = Outstate ~ Expend, data = collegeDat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15780.8  -2088.7     57.6   2010.8   7784.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.434e+03  2.248e+02   24.17   <2e-16 ***
## Expend      5.183e-01  2.047e-02   25.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2978 on 775 degrees of freedom
## Multiple R-squared:  0.4526, Adjusted R-squared:  0.4519
## F-statistic: 640.9 on 1 and 775 DF,  p-value: < 2.2e-16
```

```
(mseModel(expendLM))
```

```
## [1] 8847579
```

```
#Use the log transformation
expendLMLOG <- lm(Outstate ~ log(Expend), data = collegeDat)

summary(expendLMLOG)
```

```
##
## Call:
## lm(formula = Outstate ~ log(Expend), data = collegeDat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10650.6  -1571.5    100.5   1805.8   6603.9
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -57502.0     2089.9  -27.51   <2e-16 ***
## log(Expend)   7482.1      229.9   32.54   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2617 on 775 degrees of freedom
## Multiple R-squared:  0.5774, Adjusted R-squared:  0.5769
## F-statistic:  1059 on 1 and 775 DF,  p-value: < 2.2e-16
```

```
(mseModel(expendLMLOG))
```

```
## [1] 6830217
```

Model 3: Outstate ~ Room.Board

After doing a linear regresion, we have that Room.board has a significance in the model with a p-value less than 0.05. Still, our R-Squared is not good with only 0.428 in both cases.

```
#The normal Regression
roomLM <- lm(Outstate ~ Room.Board, data = collegeDat)

summary(roomLM)
```

```
##
## Call:
## lm(formula = Outstate ~ Room.Board, data = collegeDat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8781.0 -2070.6  -350.8  1877.4 11877.4
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.44525  447.76786  -0.039    0.969
## Room.Board    2.40001    0.09965  24.084   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3044 on 775 degrees of freedom
## Multiple R-squared:  0.4281, Adjusted R-squared:  0.4273
## F-statistic:   580 on 1 and 775 DF,  p-value: < 2.2e-16
```

```
(mseModel(roomLM))
```

```
## [1] 9244880
```

```
#Use the log transformation
roomLMLOG <- lm(Outstate ~ log(Room.Board), data = collegeDat)

summary(roomLMLOG)
```

```
##
## Call:
## lm(formula = Outstate ~ log(Room.Board), data = collegeDat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8844.2 -2060.7  -300.7  1902.4 11562.4
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -76152       3600  -21.16   <2e-16 ***
## log(Room.Board)    10373        431   24.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3045 on 775 degrees of freedom
## Multiple R-squared:  0.4277, Adjusted R-squared:  0.427
## F-statistic: 579.2 on 1 and 775 DF,  p-value: < 2.2e-16
```

```
(mseModel(roomLMLOG))
```

```
## [1] 9250602
```

Part 3: College (GAM) [3 points]

1. Split the data into a training set and a test set.

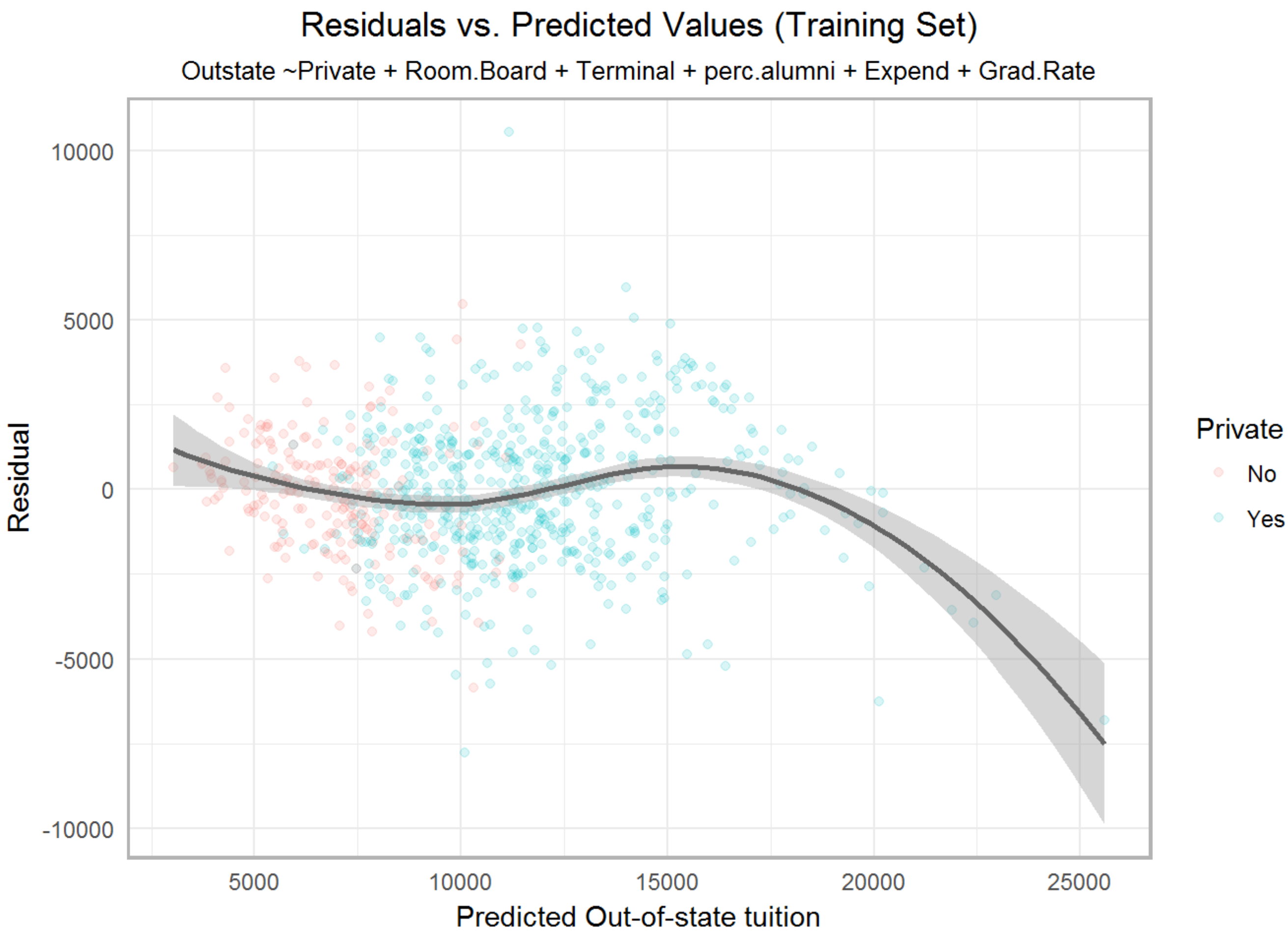
```
#Set the random generator
set.seed(1234)

#Split the data in 70% to 30% ration
collegeSplit <- resample_partition(collegeDat, c(test = 0.3, train = 0.7))
```

2. Estimate an OLS model on the training data, using out-of-state tuition (Outstate) as the response variable and the other six variables as the predictors.

```
# Create the Linear model using only the Train Data
collegeTrainLM <- lm(Outstate ~ Private + Room.Board + PhD + perc.alumni + Expend + Grad.Rate, d
ata = collegeSplit$train)

college_70 <- collegeSplit$train %>%
  tbl_df()

college_30 <- collegeSplit$test %>%
  tbl_df()

summary(collegeTrainLM)
```

```
##
## Call:
## lm(formula = Outstate ~ Private + Room.Board + PhD + perc.alumni +
##     Expend + Grad.Rate, data = collegeSplit$train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7755.7 -1325.5  -112.8  1300.0 10537.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3595.7775   538.4138  -6.678 6.04e-11 ***
## PrivateYes   2575.4372   253.9221  10.143  < 2e-16 ***
## Room.Board      0.9927     0.1028   9.661  < 2e-16 ***
## PhD            36.5287     6.8007   5.371 1.17e-07 ***
## perc.alumni    53.3855     9.0482   5.900 6.43e-09 ***
## Expend          0.2067     0.0207   9.987  < 2e-16 ***
## Grad.Rate      30.7296     6.6964   4.589 5.55e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2090 on 537 degrees of freedom
## Multiple R-squared:  0.7263, Adjusted R-squared:  0.7232
## F-statistic: 237.5 on 6 and 537 DF,  p-value: < 2.2e-16
```

```
#Add pre and res.
collegeDat %>%
  add_predictions(collegeTrainLM) %>%
  add_residuals(collegeTrainLM) %>%
  {.} -> Q3grid

#Plot the residuals
ggplot(Q3grid, mapping = aes(pred, resid)) +
       geom_point(alpha = .15, size = 1.5, aes(color=Private)) +
       geom_smooth(method = 'loess', color = 'grey40') +
       labs(title = "Residuals vs. Predicted Values (Training Set)",
            subtitle = "Outstate ~Private + Room.Board + Terminal + perc.alumni + Expend + Grad.
Rate",
            x = "Predicted Out-of-state tuition",
            y = "Residual") +
       theme(plot.title = element_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5),
panel.border = element_rect(linetype = "solid",         color = "grey70", fill=NA, size=1.2))
```



3. Estimate a GAM on the training data, using out-of-state tuition (Outstate) as the response variable and the other six variables as the predictors.

Adter creating the GAM model, we can see that the variables of Expend and PhD had a better outcome when we put them trought a log transformation. Leaving Private, Perc.Alumni and Grad.Rate linear, we can see that the two log variables (Room.Board and PhD) have a significance variance in the model. Also, we foun that all the estimator where statistically significance to the model having a p-value below 0.05.

```
#Create the GAM model with the train data
collegeGAM <- gam(Outstate ~ Private + lo(Room.Board) + lo(PhD) + perc.alumni + log(Expend) + Gr
ad.Rate,
                         data = collegeSplit$train)
summary(collegeGAM)
```

```
## 
## Call: gam(formula = Outstate ~ Private + lo(Room.Board) + lo(PhD) + 
##     perc.alumni + log(Expend) + Grad.Rate, data = collegeSplit$train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max 
## -7281.28 -1236.46    14.88  1256.22  8197.66 
## 
## (Dispersion Parameter for gaussian family taken to be 3775137)
## 
##     Null Deviance: 8567705097 on 543 degrees of freedom
## Residual Deviance: 2005836555 on 531.3281 degrees of freedom
## AIC: 9796.635 
## 
## Number of Local Scoring Iterations: 2 
## 
## Anova for Parametric Effects
##                    Df     Sum Sq    Mean Sq F value    Pr(>F)    
## Private          1.00 2331537895 2331537895 617.604 < 2.2e-16 ***
## lo(Room.Board)   1.00 1991075540 1991075540 527.418 < 2.2e-16 ***
## lo(PhD)          1.00  823589130  823589130 218.161 < 2.2e-16 ***
## perc.alumni      1.00  411366973  411366973 108.967 < 2.2e-16 ***
## log(Expend)      1.00  681244754  681244754 180.456 < 2.2e-16 ***
## Grad.Rate        1.00   87147802   87147802  23.085  2.02e-06 ***
## Residuals      531.33 2005836555    3775137                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Anova for Nonparametric Effects
##                Npar Df Npar F   Pr(F)  
## (Intercept)                            
## Private                                
## lo(Room.Board)     3.0 2.7184 0.04435 *
## lo(PhD)            2.7 3.4926 0.01922 *
## perc.alumni                            
## log(Expend)                            
## Grad.Rate                              
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4. Use the test set to evaluate the model fit of the estimated OLS and GAM models, and explain the results obtained.

We can see that the MSE for the linear test model is lower (3,031,241) than the GAM Model of the test data (3,282,077).

```
#Test MSE for College Data
#Create the Linear model with the test data
collegeLMTEST <- lm(Outstate ~ ., data = collegeSplit$test)

(collegeOLSMSE <- mseModel(collegeLMTEST))
```

```
## [1] 3031241
```

```r
#Gam Test Model
collegeGAMTEST <- gam(Outstate ~ Private + lo(Room.Board) + lo(PhD) + perc.alumni + log(Expend)
+ Grad.Rate,
                       data = collegeSplit$test)

(collegeGAMTEST <- mseModel(collegeGAMTEST))
```

```
## [1] 3282077
```

5. For which variables, if any, is there evidence of a non-linear relationship with the response?

After doing the Anova test, we can see that the are two variables that have a non-linear relationship with the response variable. Log(Room.Board) and Log(Ph.D)

```r
#See the Gam model
summary(collegeGAM)
```

```
##
## Call: gam(formula = Outstate ~ Private + lo(Room.Board) + lo(PhD) +
##     perc.alumni + log(Expend) + Grad.Rate, data = collegeSplit$train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -7281.28 -1236.46    14.88  1256.22  8197.66
##
## (Dispersion Parameter for gaussian family taken to be 3775137)
##
##     Null Deviance: 8567705097 on 543 degrees of freedom
## Residual Deviance: 2005836555 on 531.3281 degrees of freedom
## AIC: 9796.635
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                    Df     Sum Sq    Mean Sq F value    Pr(>F)
## Private          1.00 2331537895 2331537895 617.604 < 2.2e-16 ***
## lo(Room.Board)   1.00 1991075540 1991075540 527.418 < 2.2e-16 ***
## lo(PhD)          1.00  823589130  823589130 218.161 < 2.2e-16 ***
## perc.alumni      1.00  411366973  411366973 108.967 < 2.2e-16 ***
## log(Expend)      1.00  681244754  681244754 180.456 < 2.2e-16 ***
## Grad.Rate        1.00   87147802   87147802  23.085  2.02e-06 ***
## Residuals      531.33 2005836555    3775137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                Npar Df Npar F   Pr(F)
## (Intercept)
## Private
## lo(Room.Board)     3.0 2.7184 0.04435 *
## lo(PhD)            2.7 3.4926 0.01922 *
## perc.alumni
## log(Expend)
## Grad.Rate
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Get the Anova for the Predictors Variables.

# Room and Board
anova(collegeGAM)
```

```
## Anova for Nonparametric Effects
##                 Npar Df Npar F   Pr(F)
## (Intercept)
## Private
## lo(Room.Board)     3.0 2.7184 0.04435 *
## lo(PhD)            2.7 3.4926 0.01922 *
## perc.alumni
## log(Expend)
## Grad.Rate
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```