MAT 325 Essentials of Data Science

Final Project (in lieu of a traditional 2 hour final examination)

Due: December 16, 2021

Matt McCullough and Andy Mac

Note: *Insert blocks of R code and answers to the questions below. Answers in text should be displayed in italics.*

Objective: The objective of this project is to use the statistical and machine learning methods we learned this semester to model the relationship between a quantitative dependent (response or target) variable and several independent (explanatory or input) variables. Each group will work on a data set, and perform an analysis using R by following the guidelines below. Then, use this markdown file to prepare a report with the R scripts, outputs and graphs (at most 15 pages) you used to answer the questions below. Please submit three files: the data set, the updated version of this markdown (rmd) file, and a pdf rendering of the rmd file.

Part I. Write a block of R code that loads the libraries and implements any helper functions needed to answer the questions below.

```
library(ggplot2)
library(dplyr)
library(class)
library(caret)
library(ModelMetrics)
library(neuralnet)
library(tree)
library(Stat2Data)

# Normalization function
# Input: vector x
# Output: normalized vector
nor <- function(x) {
  return( (x -min(x))/(max(x)-min(x)) )
}

# This function calculates the accuracy of a
# learning algorithm
accuracy <- function(T){
    return (sum(diag(T)/(sum(rowSums(T)))) * 100)
}

# function to transform [0,1] to original scale
origscale <-function(x, xmin, xmax) {
  return( xmin + x*(xmax - xmin) )
}
```

Part II. Choose a data set suitable for a regression task that we have not used in class and that other groups are not using. Please send me the data set you plan to use (original version) by Dec 13 (Monday). Each data set must have at least 3 quantitative independent variables and at least 50 data points.

Write one or two paragraphs below that describe your data set and indicate which variable you are trying to predict and what variables you are using as predictors. Please make sure you provide the source or link to your data set.

The data set we used is the First Year GPA data set located in the Stat2Data R package. The goal of the data set is to predict first year college GPA through three quantitative predictors: High School GPA (ranges from 0.0 - 4.0), SAT Verbal scores (ranges from 0 - 800), and SAT Math scores (ranges from 0 - 800). This will be accomplished by using the sample of 219 first year college students

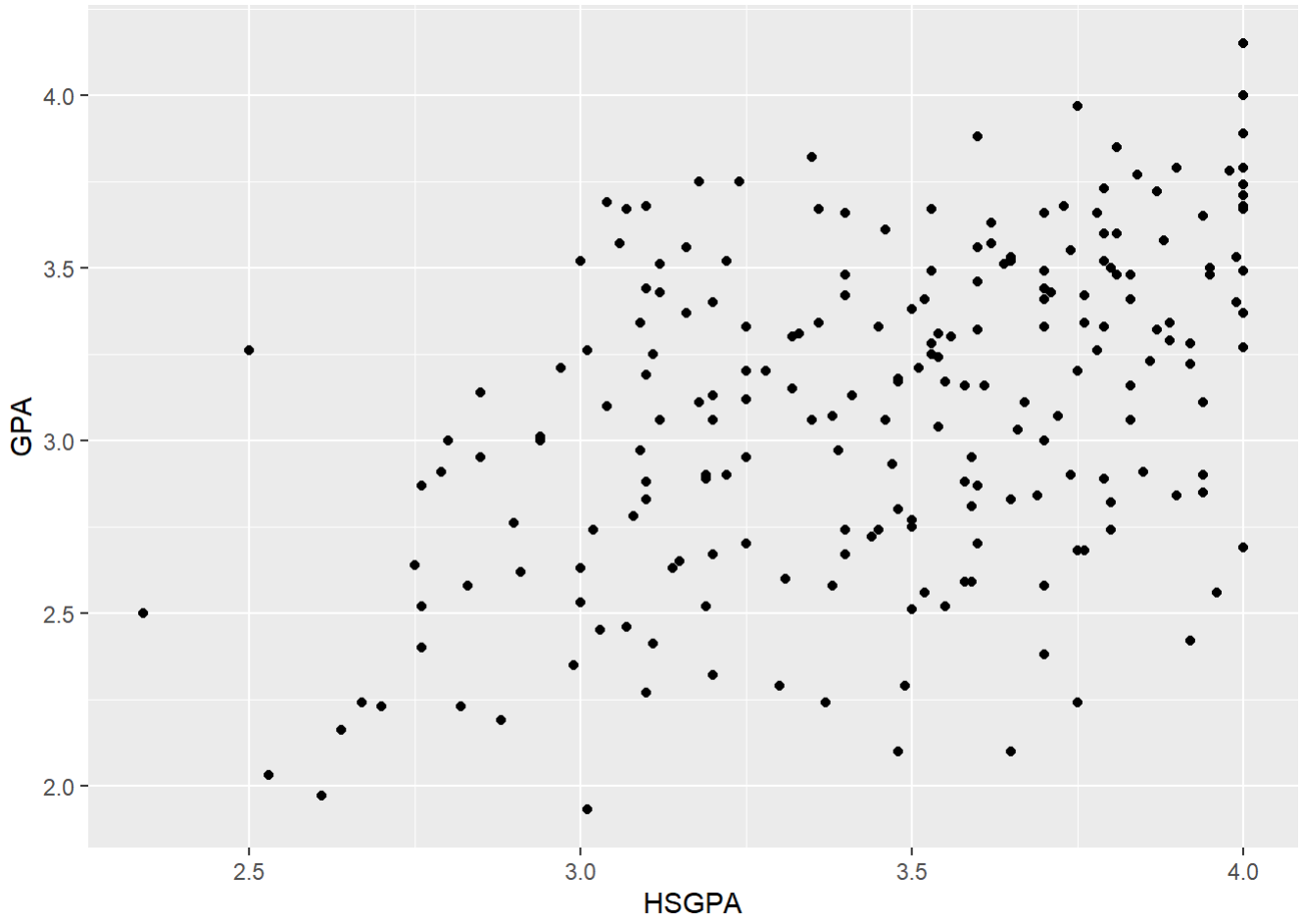
Please use any data wrangling methods you need to prepare your data set to a form that is ready for analysis. Insert the block of R code below.

```
data(FirstYearGPA)
gpaData = FirstYearGPA[,c("GPA", "HSGPA", "SATV", "SATM")]
gpaData = na.omit(gpaData)
```

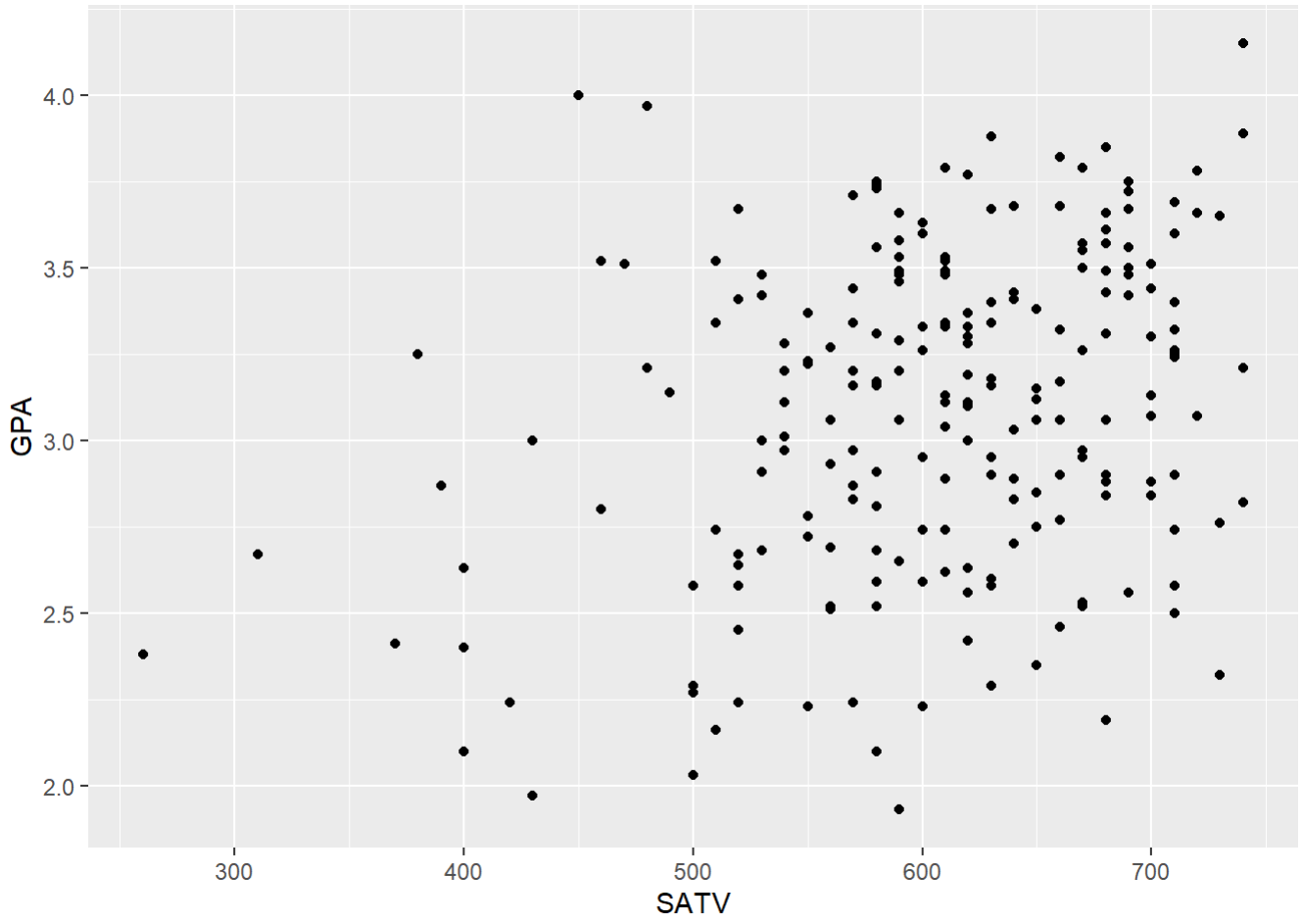
Part III. Perform Multiple Linear Regression on the Data Set

(a) Use ggplot package in R to construct scatterplots relating y to each of *at least three* of your quantitative explanatory variables $x_1, \ldots, x_k$.

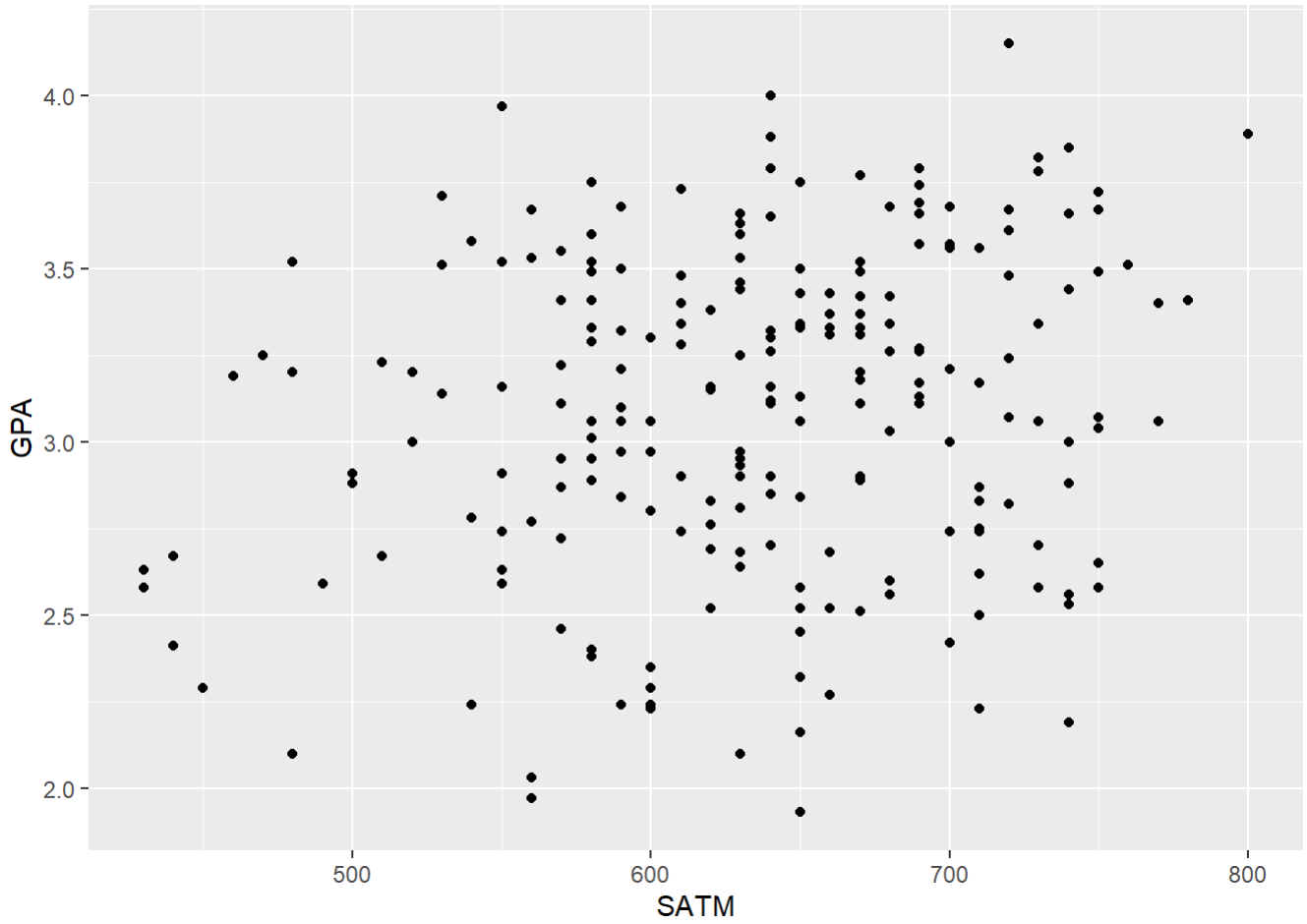
```
ggplot(data = gpaData) +
  geom_point(mapping = aes(x = HSGPA, y = GPA))
```




```
ggplot(data = gpaData) +
  geom_point(mapping = aes(x = SATV, y = GPA))
```




```
ggplot(data = gpaData) +
  geom_point(mapping = aes(x = SATM, y = GPA))
```



(b) Use R to fit a first-order linear regression model of your dependent variable y as a function of *at least three* of the quantitative explanatory variables you used in (a). Attach the output of the summary and anova commands, and give the least-squares prediction equation.

```
y=gpaData$GPA
x1=gpaData$HSGPA
x2=gpaData$SATV
x3=gpaData$SATM
m=lm(y~x1+x2+x3)
summary(m)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97999 -0.27898  0.03614  0.29873  0.88376
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.5877912  0.3269497   1.798  0.07361 .
## x1          0.4962319  0.0753072   6.589 3.35e-10 ***
## x2          0.0011595  0.0003935   2.946  0.00357 **
## x3          0.0001473  0.0004315   0.341  0.73310
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4069 on 215 degrees of freedom
## Multiple R-squared:  0.2464, Adjusted R-squared:  0.2359
## F-statistic: 23.43 on 3 and 215 DF,  p-value: 3.666e-13
```

```
anova(m)
```

```
## Analysis of Variance Table
##
## Response: y
##            Df Sum Sq Mean Sq F value    Pr(>F)
## x1          1  9.433  9.4329 56.9762 1.237e-12 ***
## x2          1  2.186  2.1861 13.2042 0.0003494 ***
## x3          1  0.019  0.0193  0.1166 0.7331038
## Residuals 215 35.595  0.1656
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Least Squares Regression Line: y = 0.5877912 + 0.4962319x_1 + 0.0011595x_2 + 0.0001473x_3 + ε

(c) Give a practical interpretation of the estimate of the coefficients of the explanatory variables in your model, if appropriate.

B0 = 0.5877912
This is the intercept for the first year college GPA of a student who has a 0.0 for high school GPA, verb SAT score, and math SAT score.
B1 = 0.4962319
When high school GPA increase by 1, assuming the verbal SAT score and math SAT score stay the same. The first year GPA increases on average by 0.4962319
B2 = 0.0011595
When verbal SAT score increases by 1, assuming the high school GPA and math SAT score stay the same. The first year GPA increases on average by 0.0011595

B3 = 0.0001473
When math SAT score increases by 1, assuming the high school GPA and verbal SAT score stay the same. The first year GPA increases on average by 0.0001473

(d) Find the model standard deviation, s, and interpret its value. Also, calculate the coefficient of variation (C.V.)

s is .4069 which means that 95% of the data should lie within 2s or .8138

```
cv <- 0.4069 / mean(gpaData$GPA) * 100
cv
```

```
## [1] 13.14207
```

Using R, we found the C.V. to be 13.14%. In general, we would like this to be around 10% as that is a sign of a preferred data set

(e) Report and interpret the adjusted R^2 values for the model in (b).

The adjusted R^2 value is 0.2359. This value represents an adjusted R^2 for both the sample size n and the number of β parameters in the model. The interpretation of this value is that approximately 23.59% of the variation of the First Year College GPA is explained by the regression line.

(f) Conduct a global F-test for overall model adequacy for the model by performing the following steps: (i) state the hypotheses; (ii) provide the value of the test statistic; (iii) provide the p-value and the formula used to calculate it; and (iv) state the conclusion.
Hypotheses: H0 : β1 = β2 = β3 = 0 vs. Ha : at least 1 βi != 0
F statistic: 23.43
P value: 3.666e-13
Conclusion: Due to the p value being less than .05, we can say that there is strong enough evidence to where we can reject the null hypothesis and say that at least 1 of the independent variables has a relationship with our response variable of first year college GPA due to one of them not being 0

(g) Conduct tests of significance to determine whether each of the explanatory variables in your first-order model in (b) are statistically useful for predicting y.

```
t.test(gpaData$HSGPA, gpaData$GPA)
```

```
##
##  Welch Two Sample t-test
##
## data:  gpaData$HSGPA and gpaData$GPA
## t = 8.8299, df = 417.02, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2771962 0.4359545
## sample estimates:
## mean of x mean of y
##  3.452740  3.096164
```

```
t.test(gpaData$SATV, gpaData$GPA)
```

```
##
##  Welch Two Sample t-test
##
## data:  gpaData$SATV and gpaData$GPA
## t = 106.82, df = 218.01, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  590.8657 613.0790
## sample estimates:
##  mean of x  mean of y
## 605.068493   3.096164
```

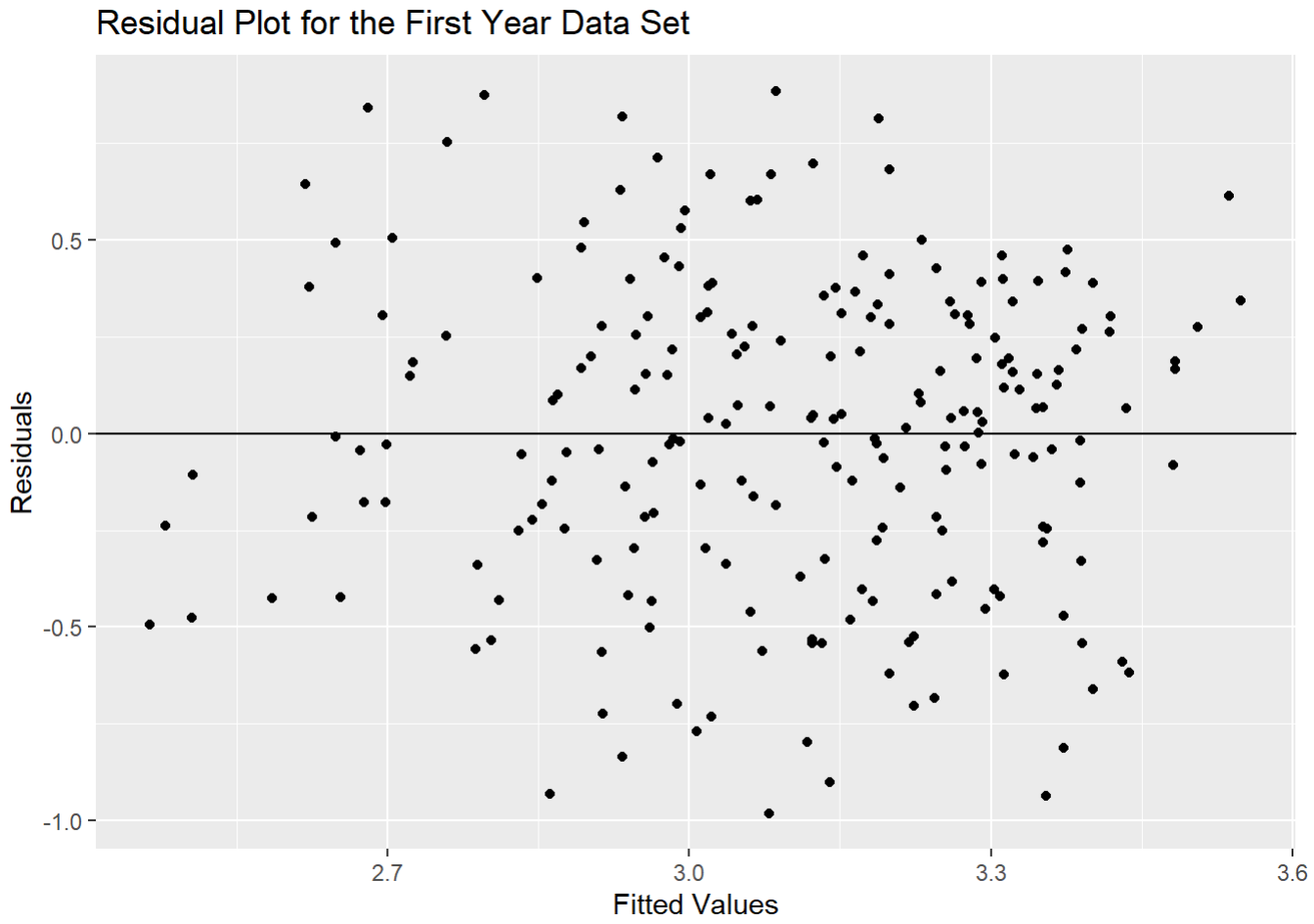
```
t.test(gpaData$SATM, gpaData$GPA)
```

```
##
##  Welch Two Sample t-test
##
## data:  gpaData$SATM and gpaData$GPA
## t = 124.15, df = 218.02, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  621.1759 641.2162
## sample estimates:
##  mean of x  mean of y
## 634.292237   3.096164
```

All of the explanatory variables are statistically significant as their p values are very low as they are less than 0.05 and as such are statistically significant at the 0.05 level

(h) Use ggplot to create a residual plot for the model. That is, create a scatterplot of the residuals vs the fitted values (or predicted values).

```
g<-ggplot(data=gpaData,aes(x=m$fitted.values,y=m$residuals))
g+geom_point()+geom_abline(intercept=0,slope=0)+xlab("Fitted Values")+
  ylab("Residuals")+
  ggtitle("Residual Plot for the First Year Data Set")
```



(i) Based on the above results, would you recommend using the model to predict your response variable? Explain.

Based on the above results, we would recommend this model to predict the response variable given the fact that all three of our predictors show significance and have a generally positive correlation between our predictors and our response variable and given the fact of the C.V. is around 10%, and the fact that our r^2 value is fairly low at 23.59% with could be better with more predictors, but given the decent C.V. and significance test, it's safe to say we would recommend this model

Part IV. (Comparison of Linear Regression and Machine Learning Methods on the Data Set) Split your data set into a training set and a test set and compare the RMSE (root mean square error) of the following models on the test set. Use either 10% or 20% of the data for the test set. Using the quantitative input variables you used in Part III, report the RMSE of the following models on the test set and summarize in a table below. Determine which methods perform well on the test set. Provide the R scripts below.

- First-order multiple linear regression model.

```
#we already have a first order multiple linear regression model, so we just need to find RMSE of it

RSS <- c(crossprod(m$residuals))
MSE <- RSS / length(m$residuals)
RMSE <- sqrt(MSE)

RMSE
```

```
## [1] 0.4031568
```

- Second-order multiple linear regression model.

```
reg <- lm(formula = GPA ~ HSGPA +
SATV + SATM + I(HSGPA*SATV) + I(HSGPA*SATM) + I(SATM*SATV) + I(HSGPA^2) + I(SATV^2) + I(SATM^2), gpaData)

RSS <- c(crossprod(reg$residuals))
MSE <- RSS / length(reg$residuals)
RMSE <- sqrt(MSE)

RMSE
```

```
## [1] 0.401516
```

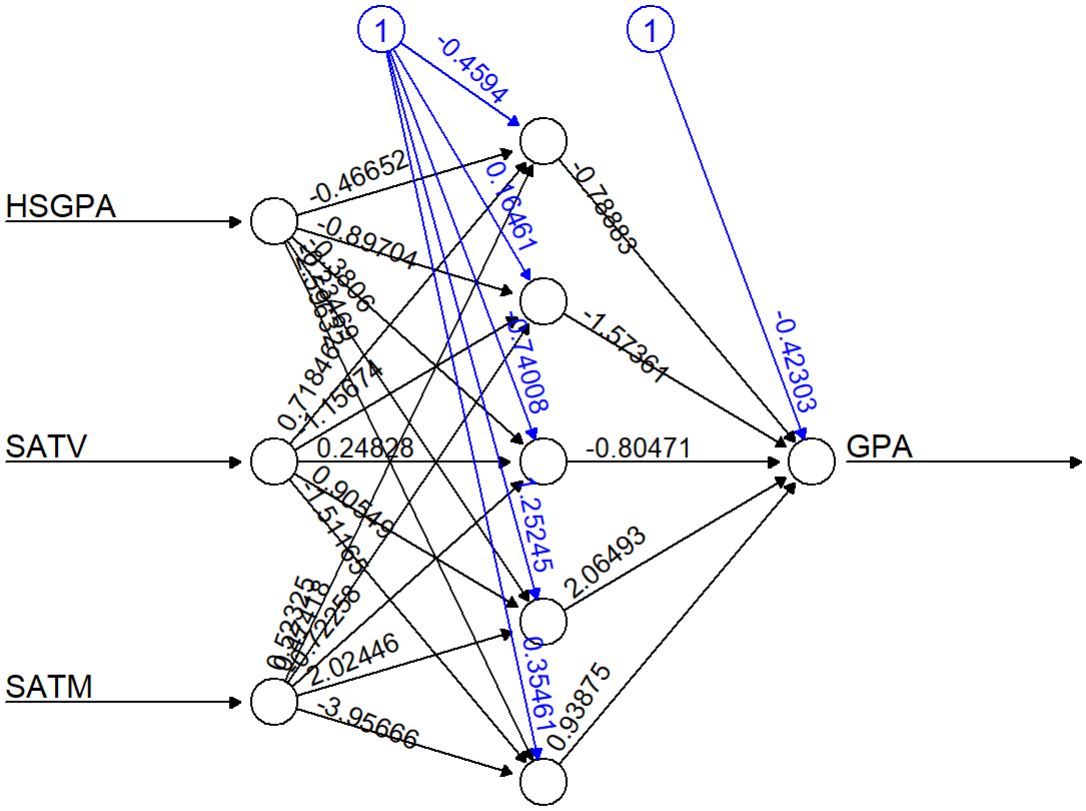
- k-NN with at least ten different values of k (include $k = 1, 3, 5$).

```
# normalize 3 columns of dataset (these
# correspond to the predictors)
gpa_norm <- as.data.frame(lapply(gpaData[,2:4], nor))

# set random seed
set.seed(1)

# Randomly select 90% of the indices (rows) of the
# dataset. This will be the indices of the training set
inTrain <- sample(1:nrow(gpaData), floor(0.9 * nrow(gpaData)))

# extract training set
gpa_trainX <- gpa_norm[inTrain,]
gpa_trainY <- gpaData[inTrain,1]

# extract testing set
gpa_testX <- gpa_norm[-inTrain,]
gpa_testY <- gpaData[-inTrain,1]

# run knnreg function
numk <- 10

for(k in 1:numk) {

  fit <- knnreg(gpa_trainX, gpa_trainY, k = k)

# predict target values (response values) on the test set
gpa_pred <- predict(fit, gpa_testX)

# calculate the RMSE (Root Mean Square Error)
gpa_rmse <- rmse(gpa_testY,gpa_pred)

print(paste('RMSE of kNNreg with k = ',k,
            ' on test set: ',gpa_rmse))

}
```

```
## [1] "RMSE of kNNreg with k =  1  on test set:  0.447320304195388"
## [1] "RMSE of kNNreg with k =  2  on test set:  0.364828726939094"
## [1] "RMSE of kNNreg with k =  3  on test set:  0.327532865187263"
## [1] "RMSE of kNNreg with k =  4  on test set:  0.358683672994363"
## [1] "RMSE of kNNreg with k =  5  on test set:  0.374870644356157"
## [1] "RMSE of kNNreg with k =  6  on test set:  0.389084817166508"
## [1] "RMSE of kNNreg with k =  7  on test set:  0.383915478484088"
## [1] "RMSE of kNNreg with k =  8  on test set:  0.368525548022731"
## [1] "RMSE of kNNreg with k =  9  on test set:  0.35250387044066"
## [1] "RMSE of kNNreg with k =  10  on test set:  0.360704802498462"
```

● Neural network at least three different architectures (include one hidden layer with 5 hidden nodes). Display the neural networks in your report.

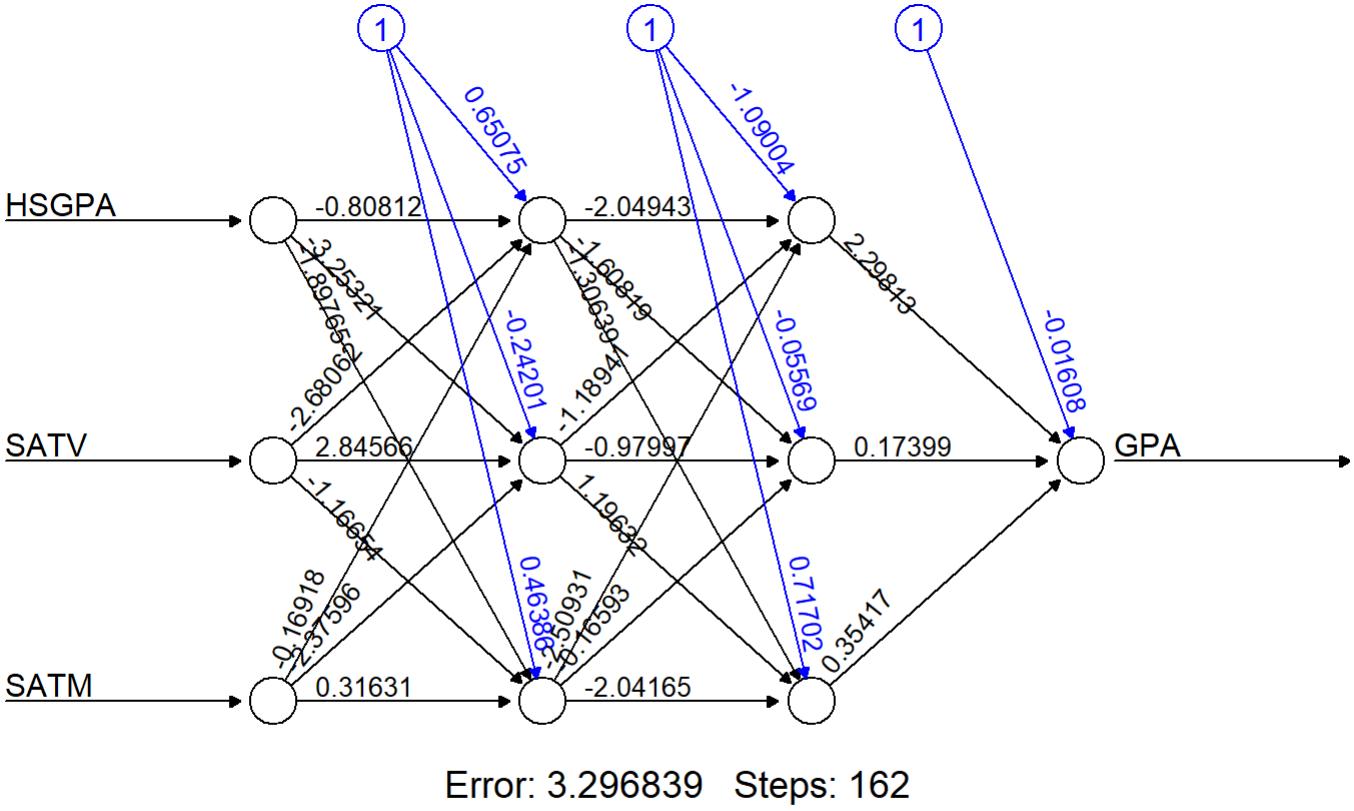
```
# normalize columns of the dataset
gpa_norm <- as.data.frame(lapply(gpaData, nor))

# Randomly select 90% of the indices (rows) of the
# dataset. This will be the indices of the training set
set.seed(1) # set random seed
inTrain <- sample(1:nrow(gpaData), floor(0.9 * nrow(gpaData)))

# extract training set
gpa_traindata <- gpa_norm[inTrain,]

# extract testing set
gpa_testdata <- gpa_norm[-inTrain,]


# train neural net with one hidden layer with
# 5 hidden nodes
hiddenlayerstruc <- 5
NN <- neuralnet(GPA~HSGPA+SATV+
        SATM,
        gpa_traindata, hidden = hiddenlayerstruc,
        linear.output = T)

# plot the neural net the "best" is required as it won't render on HTML for some reason
plot(NN, rep = "best")
```




```
# predict output on the test set
gpa_pred <- predict(NN, gpa_testdata[,2:4])


# scale the predictions back to the original scale
MinCollegeGPA<-min(gpaData[,1])
MaxCollegeGPA<-max(gpaData[,1])
gpa_pred_origscale <-origscale(gpa_pred,
          MinCollegeGPA,MaxCollegeGPA)

gpa_rmse <- rmse(gpaData[-inTrain,1],gpa_pred_origscale)

cat('RMSE of Neural Network with (',hiddenlayerstruc,
    ') hidden nodes on test set: ',gpa_rmse,'\n')
```

```
## RMSE of Neural Network with ( 5 ) hidden nodes on test set:  0.3814152
```

```
# train neural net with two hidden layer with
# 3 hidden nodes
hiddenlayerstruc <- c(3,3)
NN <- neuralnet(GPA~HSGPA+SATV+
        SATM,
        gpa_traindata, hidden = hiddenlayerstruc,
        linear.output = T)

# plot the neural net
plot(NN, rep = "best")
```

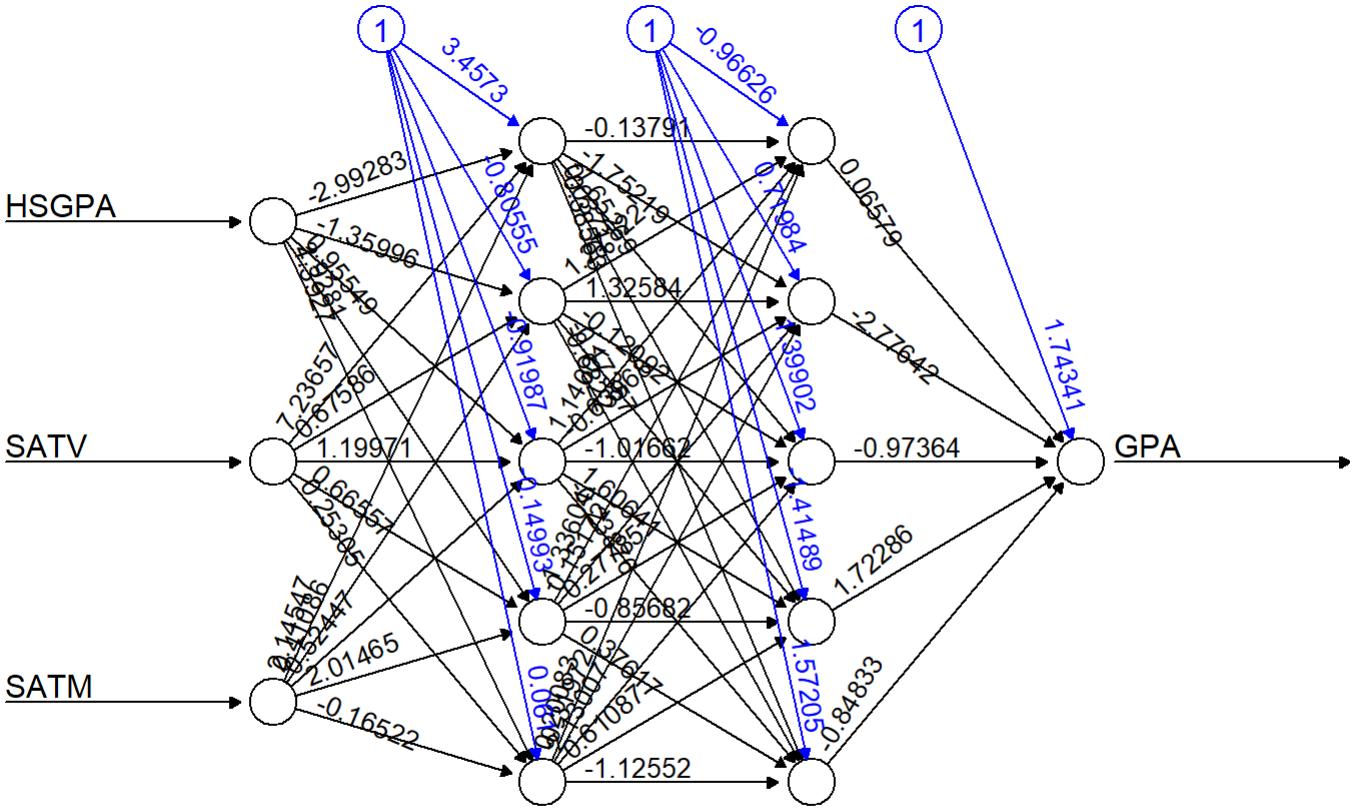


Error: 3.296839 Steps: 162

```
# predict output on the test set
gpa_pred <- predict(NN, gpa_testdata[,2:4])


# scale the predictions back to the original scale
MinCollegeGPA<-min(gpaData[,1])
MaxCollegeGPA<-max(gpaData[,1])
gpa_pred_origscale <-origscale(gpa_pred,
          MinCollegeGPA,MaxCollegeGPA)

gpa_rmse <- rmse(gpaData[-inTrain,1],gpa_pred_origscale)

cat('RMSE of Neural Network with (',hiddenlayerstruc,
    ') hidden nodes on test set: ',gpa_rmse,'\n')
```

```
## RMSE of Neural Network with ( 3 3 ) hidden nodes on test set:  0.3795818
```

```
# train neural net with two hidden layers with
# 5 hidden nodes
hiddenlayerstruc <- c(5,5)
NN <- neuralnet(GPA~HSGPA+SATV+
        SATM,
        gpa_traindata, hidden = hiddenlayerstruc,
        linear.output = T)

# plot the neural net
plot(NN, rep = "best")
```



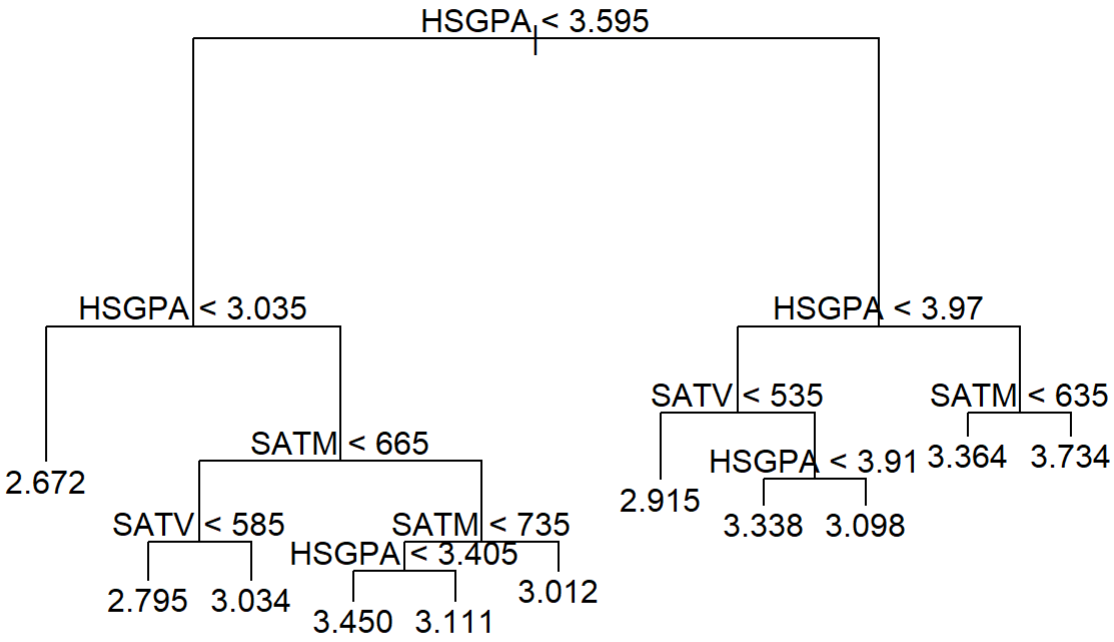
Error: 3.284492 Steps: 546

```
# predict output on the test set
gpa_pred <- predict(NN, gpa_testdata[,2:4])


# scale the predictions back to the original scale
MinCollegeGPA<-min(gpaData[,1])
MaxCollegeGPA<-max(gpaData[,1])
gpa_pred_origscale <-origscale(gpa_pred,
          MinCollegeGPA,MaxCollegeGPA)

gpa_rmse <- rmse(gpaData[-inTrain,1],gpa_pred_origscale)

cat('RMSE of Neural Network with (',hiddenlayerstruc,
    ') hidden nodes on test set: ',gpa_rmse,'\n')
```

```
## RMSE of Neural Network with ( 5 5 ) hidden nodes on test set:  0.3782195
```

- (extra credit) Decision tree. Display the tree in your report.

```
# Randomly select 90% of the indices (rows) of the
# dataset. This will be the indices of the training set
set.seed(1) # set random seed

inTrain <- sample(1:nrow(gpaData), 0.9 * nrow(gpaData))

# extract training set
gpa_traindata <- gpaData[inTrain,]

# extract testing set
gpa_testdata <- gpaData[-inTrain,]

# build regression tree to predict gpa
# using our predictors
tree.gpa <- tree(GPA~HSGPA+SATM+SATV,gpa_traindata)

# plot the tree
plot(tree.gpa)

# display the node labels
text(tree.gpa)
```



```
# predict output on the test set
gpa_pred <- predict(tree.gpa, gpa_testdata)
gpa_rmse <- rmse(gpa_testdata[,1],gpa_pred)

cat('RMSE of Regression Tree with on test set: ',gpa_rmse)
```

```
## RMSE of Regression Tree with on test set:  0.3435008
```

State your conclusion on which methods work well on your data set.

From our various methods of research, we found that our scatterplots show a positive relationship between all 3 of our predictors which suggests that they do indeed affect first year college GPA. Upon further investigation, our predictors were found to be suitable given the residual plot and hypothesis tests that suggest that all 3 of our predictors were good predictors for this data set. Given all of our methods, the lowest RMSE we found was knn at k=3 being 0.327532865187263, but the decision tree wasn't too far off either with an RMSE of 0.3435008