STAT E-109 Project - Predicting Student Performance

Isha Goyal, Hillel Landman & Rahul Sharma

1 Introduction

In this study, we built a model to predict final grades of Portuguese students in Mathematics. In Portugal, students receive a final grade using a 20-point scale, where 20 is the highest score. The dataset contained 30 predictor variables including students' demographics family background and behaviors. We decided to exclude the first and second trimester scores ("G1" and "G2" variables)in order to narrow down the prediction to factors outside of scoring itself. (*Dataset details are available in Appendix I*)

The model predicts student performance based on two different scales:
1. Final Grade (G3-Continuous) using Least Squares and Regularization Regression techniques
2. Pass/Fail (G3-Binary) using Logistic regression & NeuralNets

2 Approach

Exploratory Data Analysis

We initaited our data analysis by plotting data and visually identifying abnormalities, performing tests for multi-collianity and non-constant variance.(*Appendix II and Appendix III*) A handful of observations showed positive "G1" and "G2" scores but final "G3" scores of zero. Since it is very unlikely that students with positive first and second trimester scores would be awarded with final scores of zero, we believe those observations are incorrect. We exculded those observations from the dataset.

The highest VIF value was 5.1 (Fjob_other), so there was no concern of multicollinearity. The p-value for the NCV test was high, so we can assume constant variance.

Data Transformation

We started off with a model without transformation and performed the Ramsey RESET test to verify the model specifications. The test showed a low p-value for the null hypothesis that no transformations were needed. We proceeded to transform X variables followed by the Y variable, continuing the process until the RESET test passed.

The primary data transformations (transforming appropriate X variables to nominal or dummy variables) were based on visual data analysis. However, we used powerTransform (Box Cox) to analyze the data further. Most of the variables were either binary or nominal. There were only 2 dense continuos variables (age and absences) while the other were numeric (traveltime, studytime etc.) and similar to ordinal variables. We decided to treat them as numeric, since transformaing them into dummy variables did not add any value to model. We validated the model specifications using the RESET test.

We detected that we could produce the most predictive model by transforming Y to $\sqrt{Y}$. After modeling $\sqrt{Y}$ as a function of each X variable, the p value for the RESET test was still low. We therefore decided to experiment with various polynomial transformations until our model passed the RESET test.

The model that passed the RESET used the following transformations (*See Appendix IV for more details*):
1. *Square Root* of Y variable
2. *3 degree polynomial* for absences (X variable)
3. *2 degree polynomial* for failure, studytime and goout (X variables)

Model Building

To begin, we constructed a full linear regression model using all of the independent variables and ran an overall F-test. We then built a reduced model by dropping all variables with high p-values and ran a partial F-test to see if the reduced model was viable.

Overall F-test

An overall F test was conducted using the following hypotheses:

H_0: Independent variables is not needed to explain G3.

H_a: At Least one independent variable is needed to explain G3.

Since the F-statistic had a very low p-value, we rejected H_0, meaning at least one independent variable is needed to explain G3.

Partial F-test

A partial F-test was conducted to determine if all X variables with high p-values could be removed. We used the following hypotheses:

H_0: We can remove all X variables with high p values.

H_a: We cannot remove all X variables with high p values.

For the reduced model, we regressed Y on X variables with p-values below 0.05 (absences, studytime, failures, schoolsup_no, famsup_yes, goout, Fjob_other, Fjob_services, Fjob_health, health_1).

We calculated the anova for the full vs. reduced models, and the p-value of its F statistic came out to 0.0068, meaning we could reject H_0.

Other Models

Once we were certain that we could not remove all the variables at once, we used the following techniques to build models and compare their performances:

1. Full Model without any transformations (*failed Ramsey RESET test*).
2. Model with only Y Variable transformed to $\sqrt{Y}$ (*failed Ramsey RESET test*).

- This model outperformed the model with X variables transformed so we included it in our analysis.

3. Full Transformed Model (*passed Ramsey RESET test*).

- This model included all transformations suggested in the Data Transformation step.

Variable Selection

To get an even stronger model, we performed backward stepwise variable selection on the models with Y transformed and with the model that passed the RESET test. We performed stepwise using both AIC and BIC criteria, adding the following to our list of potential models:

5. Model with only Y variable transformed - Stepwise AIC
6. Model with only Y variable transformed - Stepwise BIC
7. Full Transformed Model - Stepwise AIC
8. Full Transformed Model - Stepwise BIC

Regularization

We also used the tranformed data model variables as inputs and created the following two additional models using Ridge and Lasso mechanisms:

9. Ridge Model
10. Lasso Model

Model Comparison

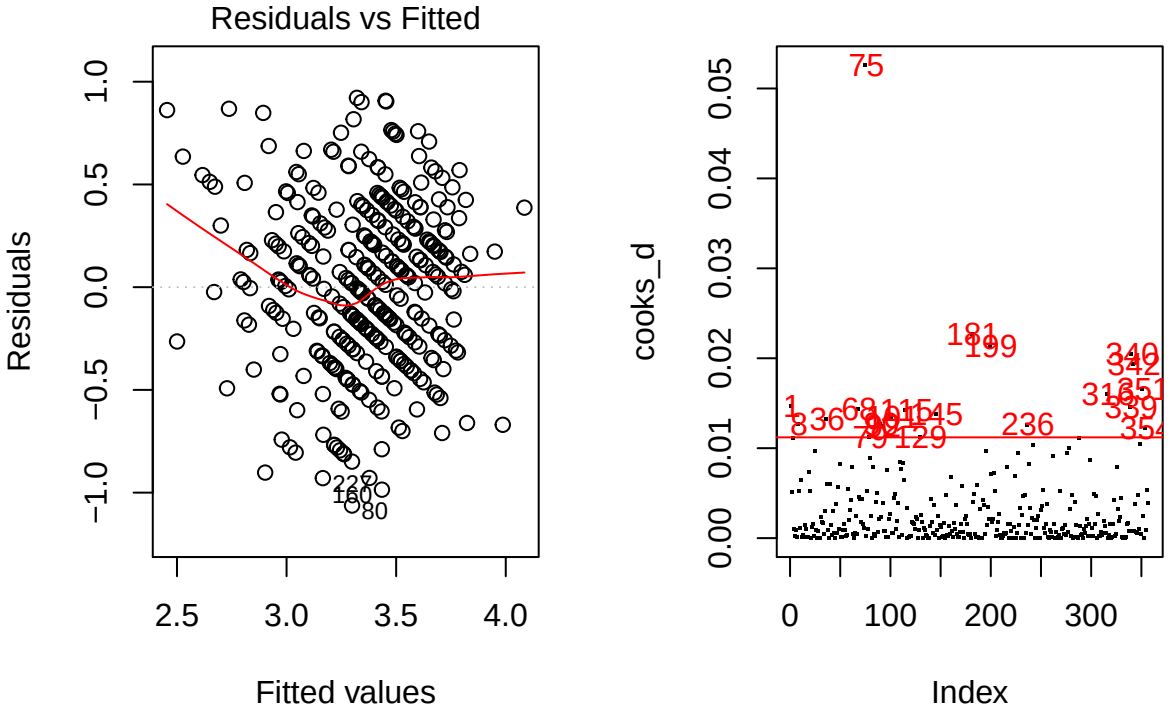
We ran 1000 simulations, splitting our data into 80% train/20% test data, and used the mean resulting RMSE values to compare our models.

We observed the lowest RMSE values for the Ridge Model, Lasso Model and Model with only Y variable transformed - Stepwise AIC, each with RMSE values between 0.42 and 0.46.

```
##    Y_Trans      Y_AIC      Y_BIC full_Trans   full_AIC   full_BIC
##  0.5549658  0.4467793  0.4536375  0.4896640  0.4957032  0.5034461
##      Ridge      Lasso
##  0.4316049  0.4378869
```

Diagnostics

We ran residulas plots on the Model with only Y variable transformed - Stepwise AIC, one of the models with the lowest RMSE. We also looked for influentual points by calculating the Cook's Distance. We decided to remove only the most extreme influential point (point #75) because we had limited visibity to the rationale we could use to remove points with only slightly high Cook's distances. When we attempted to remove more points and rerun the model, more variables would move above the Cook's distance threshold. We therefore determined that it would be best to only remove point #75.



Rerunning Simulations on Selected Models

After removing the extreme influential point, We reran the simulation on our three best models. We discovered that even without point #75, the RMSE for the Ridge, Lasso and Model with only Y variable transformed - Stepwise AIC did not change significantly.

```
##     Y_AIC     Ridge     Lasso
## 0.4475645 0.4309384 0.4361086
```

Binary Models - Logistic and Neural Networks

We also attempted to develop a model to predict if students would pass or fail. In Portugal, students fail if they receive a score below 10, and pass with a 10 or higher. We used Logistic Regression and Neural Networks to develop the model, applying these methods to the tranformed data set that passed the RESET test.

However, the accuracy of the model remained under 74%. We used an 80% train/20% test data cross-validation to validate the models. The Logistic model performed better than the neural network, with accuracy between 70-74% in the 1000 simulations.

```
##      logistic_Regress Neural_Nets
## [1,]           0.7125   0.6569444
```

Conclusion

We first began by using R^2 to compare our models but we never observed R^2 values higher than 0.35. We noticed that R^2 was decreasing when we added variables, leading us to belive that the data quality may have been poor. We then decided to use RMSE as criteria to compare the predictive ability of our models, since the RMSE values seemed satisfactory (*between 0.4 and 0.6*). We noticed the performance of the Ridge and Lasso with transformed variables (based on RESET test) performed similarly or better than regular regression models, with RMSE values between 0.4 and 0.5. The other models also produced RMSE values between 0.4 and 0.6.

When testing binary models, we observed accuracy levels between 65% and 74%. Logistic regression performed better than the Neural Network. The Neural Network performance did not improve even after increasing the intermediate steps.

Challenges

We faced a few challenges while performing the data analysis and model building processes. Key factors included lack of domain knowledge and lack of visibility as to how data was collected and which factors may have driven the data values. We noticed that our results were not improving even after removing outliers and rerunning simulations. The overall quality of the data and the nature of variables also seemed to suggest that this may not be a perfect dataset for linear analysis.

Next Steps - Enhancement

We believe the models that we developed might be further improved on by looking into the relationships between variables, understanding the data collection mechanisms, and adding more interaction terms. One other way to improve these models further could be the use of other statistical techniques such as Support Vector Machines, Gradient Boosting etc. Those were out of scope for our analysis.

References

http://www3.dsi.uminho.pt/pcortez/student.pdf

http://archive.ics.uci.edu/ml/datasets/student+performance

Appendix I - Data Variable Definitions

Dataset Link - *https://archive.ics.uci.edu/ml/datasets/Student+Performance*

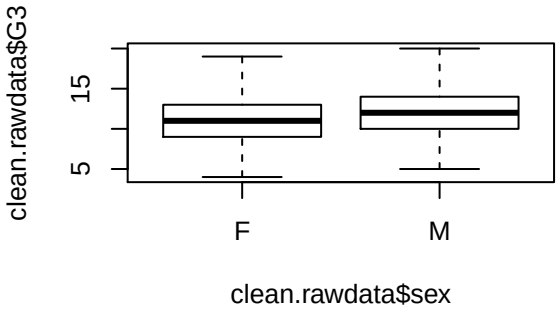
- school - student's school (binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira)
- sex - student's sex (binary: "F" - female or "M" - male)
- age - student's age (numeric: from 15 to 22)
- address - student's home address type (binary: "U" - urban or "R" - rural)
- famsize - family size (binary: "LE3" - less or equal to 3 or "GT3" - greater than 3)
- Pstatus - parent's cohabitation status (binary: "T" - living together or "A" - apart)
- Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
- Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
- Mjob - mother's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
- Fjob - father's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
- reason - reason to choose this school (nominal: close to "home", school "reputation", "course" preference or "other")
- guardian - student's guardian (nominal: "mother", "father" or "other")
- traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
- studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
- failures - number of past class failures (numeric: n if 1<=n<3, else 4)
- schoolsup - extra educational support (binary: yes or no)
- famsup - family educational support (binary: yes or no)
- paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
- activities - extra-curricular activities (binary: yes or no)
- nursery - attended nursery school (binary: yes or no)
- higher - wants to take higher education (binary: yes or no)
- internet - Internet access at home (binary: yes or no)
- romantic - with a romantic relationship (binary: yes or no)
- famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
- freetime - free time after school (numeric: from 1 - very low to 5 - very high)
- goout - going out with friends (numeric: from 1 - very low to 5 - very high)
- Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
- Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
- health - current health status (numeric: from 1 - very bad to 5 - very good)
- absences - number of school absences (numeric: from 0 to 93)

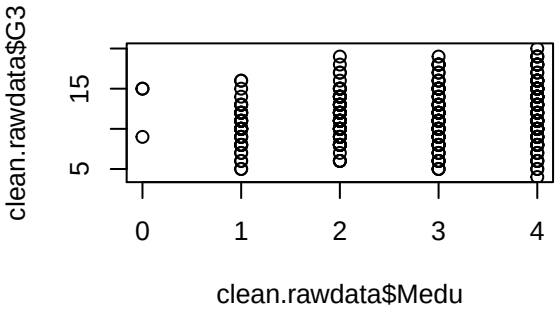
Appendix II - Data Cleaning & Transformations

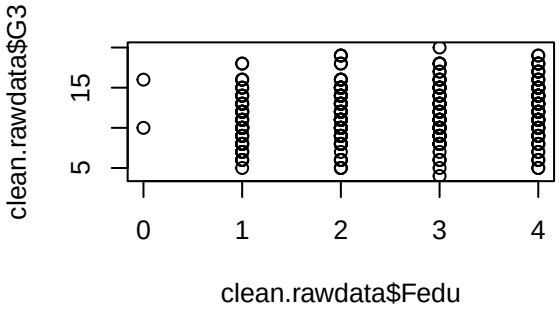
```
library(fastDummies)

# Load raw data
rawdata <- read.csv("math.csv")

# Remove all rows with Final Grade=0 to clean the data
clean.rawdata = subset(rawdata, rawdata$G3 > 0)

# Remove G1(First Period Grade) and G2(Second Period Grade) from data
datawo_g1g2 <- clean.rawdata[,!(colnames(clean.rawdata) %in% c("G1","G2"))]

# Convert binary variables to 0/1 values and Nominal Variables to Dummy Variables
dummycolconversion <- dummy_cols(datawo_g1g2, select_columns = c("school", "sex", "address", "famsize",

# final_data is our cleaned data with minor transformations to be used for continuous(Y) variable analy
final_data <- dummycolconversion[,!names(dummycolconversion) %in% c("school", "sex", "address", "famsiz

# New dataset "binary_data", where we changed variable G3 to 1 or 0 values for pass/fail analysis. Scor
binary_data <- final_data
binary_data$G3 <- replace(final_data$G3>9,1,0)
```

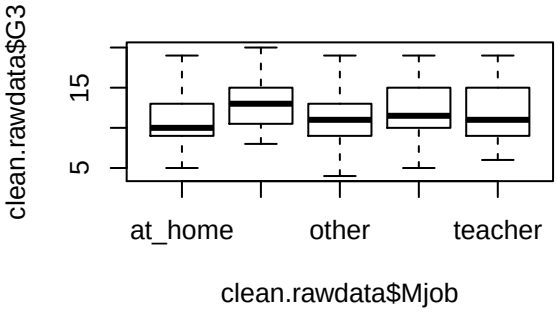
Appendix III - Data Plots, VIF & VCF Validations

```
clean.rawdata = subset(rawdata, rawdata$G3 > 0)
par(mfrow=c(2,2))
plot(clean.rawdata$G3~clean.rawdata$sex)
plot(clean.rawdata$G3~clean.rawdata$Medu)
plot(clean.rawdata$G3~clean.rawdata$Fedu)
plot(clean.rawdata$G3~clean.rawdata$Mjob)
```

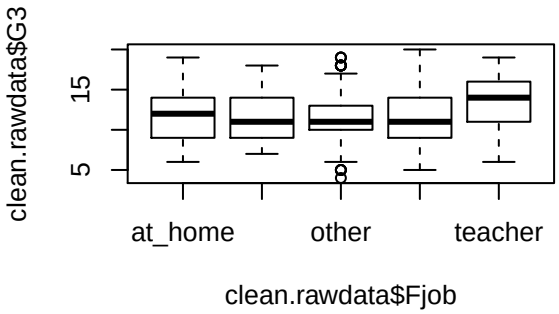


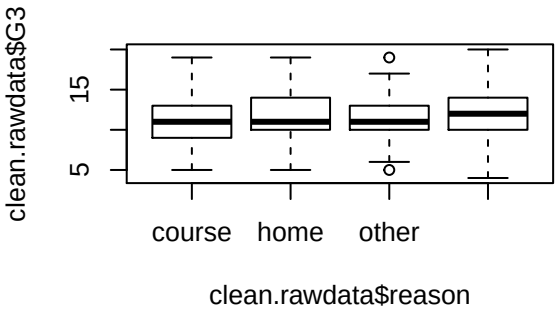


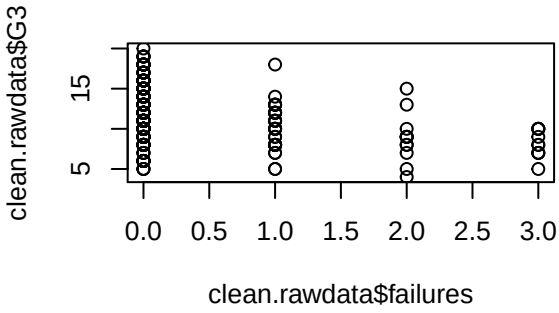




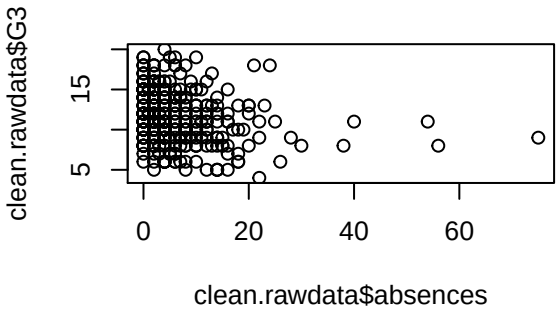

```
plot(clean.rawdata$G3~clean.rawdata$Fjob)
plot(clean.rawdata$G3~clean.rawdata$reason)
plot(clean.rawdata$G3~clean.rawdata$failures)
boxplot(G3~goout,data=clean.rawdata, xlab="G3", ylab="goout")
```

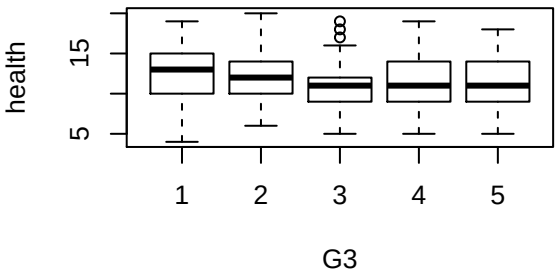


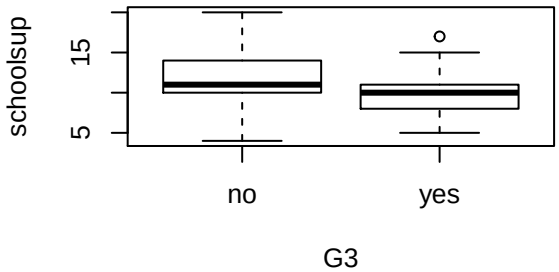


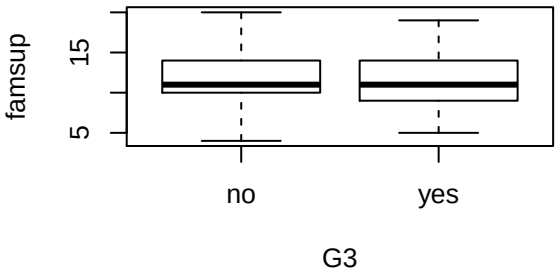



```
plot(clean.rawdata$G3~clean.rawdata$absences)
boxplot(G3~health,data=clean.rawdata, xlab="G3", ylab="health")
boxplot(G3~schoolsup,data=clean.rawdata, xlab="G3", ylab="schoolsup")
boxplot(G3~famsup,data=clean.rawdata, xlab="G3", ylab="famsup")
```










```
library(car)
raw_model <- lm(G3~.,data=final_data)
```

```r
vif(raw_model)
```

```
##                age               Medu               Fedu         traveltime
##           1.826378           2.821878           2.098839           1.294027
##          studytime           failures             famrel           freetime
##           1.377723           1.505239           1.137900           1.344215
##              goout               Dalc               Walc             health
##           1.533713           2.087345           2.471893           1.175365
##           absences          school_MS              sex_M          address_R
##           1.345099           1.532317           1.460015           1.368631
##        famsize_LE3          Pstatus_T       schoolsup_no         famsup_yes
##           1.155079           1.160783           1.206347           1.272158
##           paid_yes     activities_yes         nursery_no          higher_no
##           1.293614           1.200822           1.170381           1.315169
##       internet_yes       romantic_yes        Mjob_health        Mjob_other
##           1.242816           1.156895           2.406054           2.953776
##      Mjob_services       Mjob_teacher         Fjob_other      Fjob_services
##           3.092861           3.305712           5.097561           4.384414
##        Fjob_health       Fjob_at_home       reason_other        reason_home
##           1.825893           1.871210           1.342226           1.480049
## reason_reputation    guardian_father     guardian_other
##           1.572830           1.224853           1.537455
```

```r
ncvTest(raw_model)
```

```
## Non-constant Variance Score Test
## Variance formula: ~fitted.values
## Chisquare = 0.002390205, Df = 1, p = 0.96101
```

Appendix IV - Data Tranformations

```
library(moderndive)
# Full model
model_full <- lm(G3~.,data=final_data)
summary(model_full)
```

```
##
## Call:
## lm(formula = G3 ~ ., data = final_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4890 -1.8303  0.0622  1.6753  6.9214
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       14.53827    2.87622   5.055 7.30e-07 ***
## age               -0.18688    0.15724  -1.189 0.235522
## Medu               0.17882    0.22658   0.789 0.430574
## Fedu               0.14723    0.19717   0.747 0.455802
## traveltime         0.06489    0.24467   0.265 0.791011
## studytime          0.48298    0.20820   2.320 0.020990 *
## failures          -0.98106    0.26951  -3.640 0.000318 ***
## famrel             0.11372    0.17772   0.640 0.522722
## freetime           0.07386    0.16912   0.437 0.662628
## goout             -0.43928    0.16754  -2.622 0.009163 **
## Dalc               0.07366    0.23176   0.318 0.750836
## Walc              -0.15796    0.17915  -0.882 0.378614
## health            -0.22989    0.11406  -2.016 0.044684 *
## absences          -0.06254    0.02090  -2.992 0.002990 **
## school_MS         -0.60474    0.56615  -1.068 0.286254
## sex_M              0.51895    0.35634   1.456 0.146287
## address_R         -0.30548    0.41718  -0.732 0.464555
## famsize_LE3        0.32023    0.34568   0.926 0.354960
## Pstatus_T         -0.19808    0.50894  -0.389 0.697384
## schoolsup_no       2.42149    0.46635   5.192 3.72e-07 ***
## famsup_yes        -0.69942    0.34131  -2.049 0.041260 *
## paid_yes          -0.42560    0.33535  -1.269 0.205339
## activities_yes     0.12094    0.32296   0.374 0.708296
## nursery_no         0.32147    0.39938   0.805 0.421460
## higher_no         -0.06856    0.87059  -0.079 0.937276
## internet_yes       0.60860    0.44534   1.367 0.172717
## romantic_yes      -0.08457    0.34158  -0.248 0.804624
## Mjob_health        1.24197    0.80015   1.552 0.121618
## Mjob_other        -0.45820    0.52900  -0.866 0.387058
## Mjob_services      0.86213    0.58840   1.465 0.143853
## Mjob_teacher      -0.68915    0.74774  -0.922 0.357413
## Fjob_other        -1.87059    0.66861  -2.798 0.005461 **
## Fjob_services     -1.91470    0.68710  -2.787 0.005647 **
## Fjob_health       -1.83863    0.90999  -2.021 0.044172 *
## Fjob_at_home      -1.16429    0.94652  -1.230 0.219583
## reason_other      -0.14831    0.57410  -0.258 0.796319
```

```
## reason_home        0.27233    0.40299   0.676 0.499679
## reason_reputation  0.07426    0.41281   0.180 0.857355
## guardian_father   -0.11326    0.38771  -0.292 0.770381
## guardian_other     0.70359    0.69103   1.018 0.309367
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.784 on 317 degrees of freedom
## Multiple R-squared:  0.3375, Adjusted R-squared:  0.256
## F-statistic:  4.14 on 39 and 317 DF,  p-value: 5.965e-13
```

```
library(lmtest)

# Testing model with transformed x variables
resettest(model_full)
```

```
##
##  RESET test
##
## data:  model_full
## RESET = 7.7764, df1 = 2, df2 = 315, p-value = 0.0005052
```

```
# Power Transformation test for the Y variable to detect transformation
summary(powerTransform(model_full))
```

```
## bcPower Transformation to Normality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1    0.6686         0.5       0.3728       0.9644
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                            LRT df       pval
## LR test, lambda = (0) 20.59349  1 5.6789e-06
##
## Likelihood ratio test that no transformation is needed
##                            LRT df     pval
## LR test, lambda = (1) 4.699124  1 0.030178
```

```
# Fully-transformed model
Y_model_trans = lm(sqrt(G3)~., data=final_data)

get_regression_table(Y_model_trans)
```

```
## # A tibble: 40 x 7
##    term       estimate std_error statistic p_value lower_ci upper_ci
##    <chr>         <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
##  1 intercept     3.78      0.431     8.77    0        2.93     4.62
##  2 age          -0.026     0.024    -1.11    0.268   -0.072    0.02
##  3 Medu          0.016     0.034     0.484   0.629   -0.05     0.083
##  4 Fedu          0.027     0.03      0.9     0.369   -0.031    0.085
##  5 traveltime    0.016     0.037     0.445   0.657   -0.056    0.088
##  6 studytime     0.073     0.031     2.35    0.019    0.012    0.135
##  7 failures     -0.157     0.04     -3.89    0       -0.236   -0.078
##  8 famrel        0.017     0.027     0.622   0.535   -0.036    0.069
##  9 freetime      0.008     0.025     0.305   0.76    -0.042    0.058
## 10 goout        -0.07      0.025    -2.80    0.005   -0.119   -0.021
```

```
## # ... with 30 more rows
```

```
# Reset test on fully transformed model
resettest(Y_model_trans)
```

```
##
##  RESET test
##
## data:  Y_model_trans
## RESET = 5.8063, df1 = 2, df2 = 315, p-value = 0.00334
```

Appendix V - Overall F Test for Dropping all Variable at Once

```
# Full Model with no transformations
model_full <- lm(G3~.,data=final_data)
summary(model_full)
```

```
##
## Call:
## lm(formula = G3 ~ ., data = final_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4890 -1.8303  0.0622  1.6753  6.9214
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       14.53827    2.87622   5.055 7.30e-07 ***
## age               -0.18688    0.15724  -1.189 0.235522
## Medu               0.17882    0.22658   0.789 0.430574
## Fedu               0.14723    0.19717   0.747 0.455802
## traveltime         0.06489    0.24467   0.265 0.791011
## studytime          0.48298    0.20820   2.320 0.020990 *
## failures          -0.98106    0.26951  -3.640 0.000318 ***
## famrel             0.11372    0.17772   0.640 0.522722
## freetime           0.07386    0.16912   0.437 0.662628
## goout             -0.43928    0.16754  -2.622 0.009163 **
## Dalc               0.07366    0.23176   0.318 0.750836
## Walc              -0.15796    0.17915  -0.882 0.378614
## health            -0.22989    0.11406  -2.016 0.044684 *
## absences          -0.06254    0.02090  -2.992 0.002990 **
## school_MS         -0.60474    0.56615  -1.068 0.286254
## sex_M              0.51895    0.35634   1.456 0.146287
## address_R         -0.30548    0.41718  -0.732 0.464555
## famsize_LE3        0.32023    0.34568   0.926 0.354960
## Pstatus_T         -0.19808    0.50894  -0.389 0.697384
## schoolsup_no       2.42149    0.46635   5.192 3.72e-07 ***
## famsup_yes        -0.69942    0.34131  -2.049 0.041260 *
## paid_yes          -0.42560    0.33535  -1.269 0.205339
## activities_yes     0.12094    0.32296   0.374 0.708296
## nursery_no         0.32147    0.39938   0.805 0.421460
## higher_no         -0.06856    0.87059  -0.079 0.937276
## internet_yes       0.60860    0.44534   1.367 0.172717
## romantic_yes      -0.08457    0.34158  -0.248 0.804624
## Mjob_health        1.24197    0.80015   1.552 0.121618
## Mjob_other        -0.45820    0.52900  -0.866 0.387058
## Mjob_services      0.86213    0.58840   1.465 0.143853
## Mjob_teacher      -0.68915    0.74774  -0.922 0.357413
## Fjob_other        -1.87059    0.66861  -2.798 0.005461 **
## Fjob_services     -1.91470    0.68710  -2.787 0.005647 **
## Fjob_health       -1.83863    0.90999  -2.021 0.044172 *
## Fjob_at_home      -1.16429    0.94652  -1.230 0.219583
## reason_other      -0.14831    0.57410  -0.258 0.796319
## reason_home        0.27233    0.40299   0.676 0.499679
```

```
## reason_reputation  0.07426    0.41281   0.180 0.857355
## guardian_father   -0.11326    0.38771  -0.292 0.770381
## guardian_other     0.70359    0.69103   1.018 0.309367
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.784 on 317 degrees of freedom
## Multiple R-squared:  0.3375, Adjusted R-squared:  0.256
## F-statistic:  4.14 on 39 and 317 DF,  p-value: 5.965e-13
```

```
# Reduced model after dropping all the variables at once
model_reduced = lm(G3~absences+studytime+failures+schoolsup_no+famsup_yes
                   +goout+Fjob_other+Fjob_services+Fjob_health+health, data = final_data)
get_regression_table(model_reduced)
```

```
## # A tibble: 11 x 7
##    term          estimate std_error statistic p_value lower_ci upper_ci
##    <chr>            <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
##  1 intercept       13.0       0.916     14.2    0       11.2     14.8
##  2 absences        -0.063     0.019     -3.33   0.001   -0.1     -0.026
##  3 studytime        0.346     0.19       1.82   0.069   -0.027    0.72
##  4 failures        -1.07      0.235     -4.56   0       -1.54    -0.611
##  5 schoolsup_no     2.26      0.446      5.07   0        1.38     3.14
##  6 famsup_yes      -0.47      0.322     -1.46   0.145   -1.10     0.163
##  7 goout           -0.415     0.141     -2.93   0.004   -0.693   -0.137
##  8 Fjob_other      -1.51      0.488     -3.09   0.002   -2.46    -0.546
##  9 Fjob_services   -1.28      0.531     -2.41   0.017   -2.32    -0.234
## 10 Fjob_health     -0.882     0.82      -1.08   0.283   -2.50     0.731
## 11 health          -0.169     0.11      -1.53   0.126   -0.385    0.048
```

```
anova(model_full, model_reduced)
```

```
## Analysis of Variance Table
##
## Model 1: G3 ~ age + Medu + Fedu + traveltime + studytime + failures +
##     famrel + freetime + goout + Dalc + Walc + health + absences +
##     school_MS + sex_M + address_R + famsize_LE3 + Pstatus_T +
##     schoolsup_no + famsup_yes + paid_yes + activities_yes + nursery_no +
##     higher_no + internet_yes + romantic_yes + Mjob_health + Mjob_other +
##     Mjob_services + Mjob_teacher + Fjob_other + Fjob_services +
##     Fjob_health + Fjob_at_home + reason_other + reason_home +
##     reason_reputation + guardian_father + guardian_other
## Model 2: G3 ~ absences + studytime + failures + schoolsup_no + famsup_yes +
##     goout + Fjob_other + Fjob_services + Fjob_health + health
##   Res.Df    RSS  Df Sum of Sq     F   Pr(>F)
## 1    317 2457.3
## 2    346 2868.9 -29   -411.62 1.831 0.006784 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Appendix VI - Generalized Models based on transformations

```
# Full Transfromed model based on passed RESET test
full_model_trans <- lm(sqrt(G3)~.+I(poly(absences,3))-absences
                       +I(poly(failures,2))-failures+I(poly(studytime,2))
                       -studytime+I(poly(goout,2))-goout,data=final_data)
get_regression_table(full_model_trans)
```

```
## # A tibble: 45 x 7
##    term       estimate std_error statistic p_value lower_ci upper_ci
##    <chr>         <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
##  1 intercept     3.55      0.443     8.00    0        2.68     4.42
##  2 age          -0.018     0.024    -0.779   0.437   -0.065    0.028
##  3 Medu          0.012     0.034     0.355   0.722   -0.055    0.079
##  4 Fedu          0.027     0.03      0.907   0.365   -0.032    0.085
##  5 traveltime    0.006     0.037     0.16    0.873   -0.067    0.079
##  6 famrel        0.01      0.027     0.377   0.707   -0.043    0.063
##  7 freetime      0.006     0.025     0.231   0.817   -0.044    0.056
##  8 Dalc          0.016     0.035     0.466   0.641   -0.052    0.084
##  9 Walc         -0.023     0.027    -0.852   0.395   -0.076    0.03
## 10 health       -0.035     0.017    -2.02    0.044   -0.068   -0.001
## # ... with 35 more rows
```

```
# Model based on the backward stepwise and AIC approach by transforming only Y variable
Y_model_trans_AIC = step(Y_model_trans,direction="backward", trace = 0)
get_regression_table(Y_model_trans_AIC)
```

```
## # A tibble: 16 x 7
##    term          estimate std_error statistic p_value lower_ci upper_ci
##    <chr>            <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
##  1 intercept        3.43      0.15      22.9    0        3.14     3.73
##  2 Fedu             0.037     0.023      1.64   0.102   -0.007    0.082
##  3 studytime        0.076     0.028      2.69   0.007    0.021    0.132
##  4 failures        -0.15      0.035     -4.26   0       -0.22    -0.081
##  5 goout           -0.077     0.02      -3.77   0       -0.117   -0.037
##  6 health          -0.037     0.016     -2.31   0.022   -0.068   -0.005
##  7 absences        -0.009     0.003     -3.44   0.001   -0.015   -0.004
##  8 school_MS       -0.149     0.071     -2.10   0.036   -0.288   -0.01
##  9 sex_M            0.096     0.047      2.04   0.043    0.003    0.189
## 10 schoolsup_no     0.331     0.065      5.07   0        0.202    0.459
## 11 famsup_yes      -0.115     0.047     -2.43   0.016   -0.208   -0.022
## 12 Mjob_health      0.277     0.081      3.42   0.001    0.117    0.436
## 13 Mjob_services    0.206     0.052      3.93   0        0.103    0.308
## 14 Fjob_other      -0.197     0.073     -2.70   0.007   -0.339   -0.054
## 15 Fjob_services   -0.203     0.077     -2.63   0.009   -0.355   -0.051
## 16 Fjob_health     -0.21      0.12      -1.76   0.08    -0.446    0.025
```

```
# Model based on the backward stepwise and BIC approach by transforming only Y variable
n = nrow(final_data)
Y_model_trans_BIC = step(Y_model_trans,direction="backward",k=log(n), trace = 0)
get_regression_table(Y_model_trans_BIC)
```

```
## # A tibble: 7 x 7
##   term          estimate std_error statistic p_value lower_ci upper_ci
##   <chr>            <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
```

```
## 1 intercept        3.35      0.089     37.7    0        3.18     3.53
## 2 failures        -0.187     0.035     -5.42   0       -0.255   -0.119
## 3 goout           -0.074     0.021     -3.52   0       -0.115   -0.033
## 4 absences        -0.01      0.003     -3.51   0.001   -0.015   -0.004
## 5 schoolsup_no     0.324     0.065      4.97   0        0.196    0.452
## 6 Mjob_health      0.238     0.081      2.94   0.003    0.079    0.397
## 7 Mjob_services    0.194     0.053      3.70   0        0.091    0.297
```

```
# Model based on the backward stepwise and AIC approach for full transformed model
full_model_trans_AIC = step(full_model_trans,direction="backward", trace = 0)
get_regression_table(full_model_trans_AIC)
```

```
## # A tibble: 23 x 7
##    term          estimate std_error statistic p_value lower_ci upper_ci
##    <chr>            <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
##  1 intercept        3.20      0.129     24.8    0        2.95     3.45
##  2 Fedu             0.035     0.023      1.54   0.124   -0.01     0.08
##  3 health          -0.037     0.016     -2.33   0.021   -0.069   -0.006
##  4 school_MS       -0.148     0.071     -2.08   0.039   -0.288   -0.008
##  5 sex_M            0.083     0.048      1.74   0.083   -0.011    0.177
##  6 schoolsup_no     0.331     0.065      5.08   0        0.203    0.459
##  7 famsup_yes      -0.116     0.047     -2.45   0.015   -0.209   -0.023
##  8 internet_yes     0.086     0.062      1.40   0.163   -0.035    0.207
##  9 Mjob_health      0.272     0.081      3.36   0.001    0.113    0.432
## 10 Mjob_services    0.21      0.053      3.98   0        0.106    0.314
## # ... with 13 more rows
```

```
# Model based on the backward stepwise and AIC approach for full transformed model
full_model_trans_BIC = step(full_model_trans,direction="backward",k=log(n), trace = 0)
get_regression_table(full_model_trans_BIC)
```

```
## # A tibble: 6 x 7
##   term                estimate std_error statistic p_value lower_ci upper_ci
##   <chr>                  <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept              3.02      0.065     46.6    0        2.89     3.15
## 2 schoolsup_no           0.313     0.067      4.65   0        0.181    0.445
## 3 Mjob_health            0.232     0.083      2.79   0.006    0.068    0.395
## 4 Mjob_services          0.194     0.054      3.57   0        0.087    0.301
## 5 I(poly(failures, ~    -2.80      0.444     -6.30   0       -3.67    -1.92
## 6 I(poly(failures, ~     0.535     0.441      1.21   0.226   -0.332    1.40
```

Appendix VII - Regularization Models(Ridge and Lasso)

```
# Regularuization Models - Ridge and Lass on Full Transformed Model
library(glmnet)
library(dplyr)


X <- model.matrix(full_model_trans)
X <- X[,-1]
y <- as.matrix(sqrt(final_data$G3))


# Setting alpha = 0 implements ridge regression
ridge_cv <- cv.glmnet(X, y, alpha = 0)

# Fit final model, get its sum of squared residuals and multiple R-squared
ridge_model <- glmnet(X, y, alpha = 0, lambda = ridge_cv$lambda.min, standardize = TRUE)
coef(ridge_model)
```

```
## 45 x 1 sparse Matrix of class "dgCMatrix"
##                                    s0
## (Intercept)             3.376379456
## age                    -0.012279762
## Medu                    0.021404617
## Fedu                    0.025057497
## traveltime             -0.001987569
## famrel                  0.005360643
## freetime                0.002263523
## Dalc                   -0.004904858
## Walc                   -0.022781876
## health                 -0.021557250
## school_MS              -0.062506119
## sex_M                   0.064607020
## address_R              -0.055447902
## famsize_LE3             0.034039115
## Pstatus_T              -0.034996880
## schoolsup_no            0.241478358
## famsup_yes             -0.068609960
## paid_yes               -0.040780018
## activities_yes          0.023317545
## nursery_no              0.036795786
## higher_no              -0.073298191
## internet_yes            0.062259227
## romantic_yes           -0.001427092
## Mjob_health             0.118132558
## Mjob_other             -0.069142637
## Mjob_services           0.081310854
## Mjob_teacher           -0.034434675
## Fjob_other             -0.054884385
## Fjob_services          -0.059937571
## Fjob_health            -0.060721488
## Fjob_at_home            0.030411372
## reason_other            0.009081658
```

```
## reason_home             0.029402171
## reason_reputation       0.015207703
## guardian_father         0.004150853
## guardian_other          0.050880080
## I(poly(absences, 3))1  -1.115125756
## I(poly(absences, 3))2   0.594212748
## I(poly(absences, 3))3  -0.332457646
## I(poly(failures, 2))1  -1.323429381
## I(poly(failures, 2))2   0.174248199
## I(poly(studytime, 2))1  0.710099868
## I(poly(studytime, 2))2  0.503077856
## I(poly(goout, 2))1     -0.888973198
## I(poly(goout, 2))2      0.156858183
```

```
# Setting alpha = 1 implements lasso regression
lasso_cv <- cv.glmnet(X, y, alpha = 1)

# Fits the Lasso model
lasso_model <- glmnet(X, y, alpha = 1, lambda = lasso_cv$lambda.min, standardize = TRUE)
coef(lasso_model)
```

```
## 45 x 1 sparse Matrix of class "dgCMatrix"
##                                  s0
## (Intercept)             3.270054568
## age                    -0.007102499
## Medu                    0.012943977
## Fedu                    0.024528185
## traveltime              .
## famrel                  .
## freetime                .
## Dalc                    .
## Walc                   -0.017790009
## health                 -0.025969296
## school_MS              -0.076550683
## sex_M                   0.059361926
## address_R              -0.048362191
## famsize_LE3             0.018197032
## Pstatus_T              -0.018608366
## schoolsup_no            0.304966984
## famsup_yes             -0.070695463
## paid_yes               -0.041560084
## activities_yes          0.015347390
## nursery_no              0.013876154
## higher_no              -0.008105569
## internet_yes            0.059334451
## romantic_yes            .
## Mjob_health             0.162381399
## Mjob_other             -0.035906060
## Mjob_services           0.135254165
## Mjob_teacher            .
## Fjob_other             -0.064162831
## Fjob_services          -0.075193930
## Fjob_health            -0.032088087
## Fjob_at_home            .
## reason_other            .
```

```
## reason_home             0.012924807
## reason_reputation       .
## guardian_father         .
## guardian_other          0.051683461
## I(poly(absences, 3))1  -1.383755173
## I(poly(absences, 3))2   0.594404375
## I(poly(absences, 3))3  -0.277424220
## I(poly(failures, 2))1  -1.725470396
## I(poly(failures, 2))2   .
## I(poly(studytime, 2))1  0.759579238
## I(poly(studytime, 2))2  0.438276239
## I(poly(goout, 2))1     -1.087565283
## I(poly(goout, 2))2      0.022558729
```

Appendix VIII - Simulation of General an regularization Models

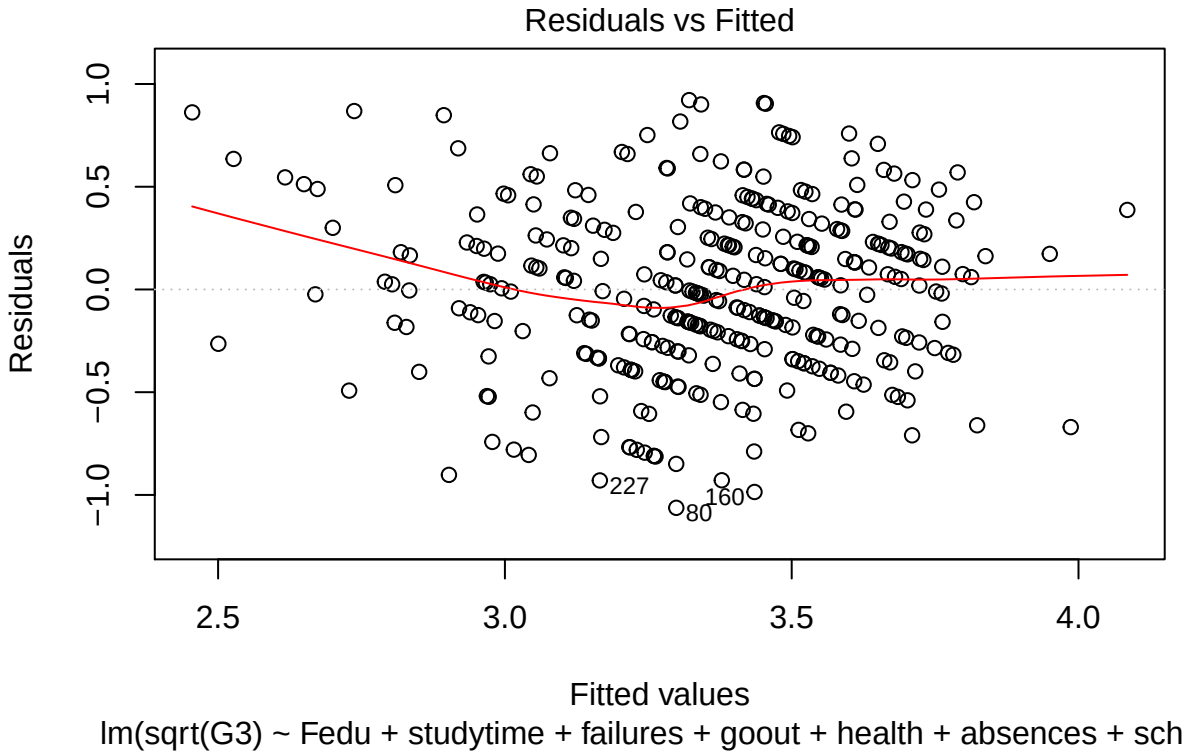
```
library(Metrics)

# Number of simulations - we used 1000 for analysis
nsims = 10

sse_Y_trans = sse_Y_AIC = sse_Y_BIC = sse_full_trans = sse_full_AIC = sse_full_BIC = sse_Ridge = sse_Las

# LOOCV RMSE function
calc_loocv_rmse = function(model) {
sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}

for (i in 1:nsims) {
# Build train and test
sample <- sample.int(n=nrow(final_data),size=floor(0.8*nrow(final_data)))
final_data.train <- final_data[sample,]
final_data.test <- final_data[-sample,]

# Y-Transform model
Y_model_trans_train = lm(sqrt(G3)~., data=final_data.train)

# Full transform model
full_model_trans_train <- lm(sqrt(G3)~.+I(poly(absences,3))-absences+I(poly(failures,2))-failures+I(poly

# Stepwise on Y transform
Y_model_trans_AIC_train = step(Y_model_trans_train,direction="backward", trace = 0)

Y_model_trans_BIC_train = step(Y_model_trans_train,direction="backward",k=log(n), trace = 0)

# Stepwise on full transform
full_model_trans_AIC_train = step(full_model_trans_train,direction="backward", trace = 0)

full_model_trans_BIC_train = step(full_model_trans_train,direction="backward",k=log(n), trace = 0)


# Ridge/lasso data
X <- model.matrix(Y_model_trans_train)
X <- X[,-1]
y <- as.matrix(sqrt(final_data.train$G3))

Y_model_trans_test = lm(sqrt(G3)~., data=final_data.test)
X_new <-model.matrix(Y_model_trans_test)
X_new <- X_new[,-1]

ridge_model_train <- glmnet(X, y, alpha = 0, lambda = ridge_cv$lambda.min, standardize = TRUE)

lasso_model_train <- glmnet(X, y, alpha = 1, lambda = lasso_cv$lambda.min, standardize = TRUE)


# RMSE Formula
sse_Y_trans[i] = rmse(sqrt(final_data.test$G3), predict(Y_model_trans_train, data=final_data.test, type=
```

```
sse_Y_AIC[i] = rmse(sqrt(final_data.test$G3), predict(Y_model_trans_AIC_train, newdata=final_data.test,
sse_Y_BIC[i] = rmse(sqrt(final_data.test$G3), predict(Y_model_trans_BIC_train, newdata=final_data.test,
sse_full_trans[i] = rmse(sqrt(final_data.test$G3), predict(full_model_trans_train, newdata=final_data.te
sse_full_AIC[i] = rmse(sqrt(final_data.test$G3), predict(full_model_trans_AIC_train, newdata=final_data
sse_full_BIC[i] = rmse(sqrt(final_data.test$G3), predict(full_model_trans_BIC_train, newdata=final_data
sse_Ridge[i] = rmse(sqrt(final_data.test$G3), predict(ridge_model_train, newx = X_new, type="response"))
sse_Lasso[i] = rmse(sqrt(final_data.test$G3), predict(lasso_model_train, newx = X_new, type="response"))
}

# Table with means of RMSE values
rmse_values <- c(Y_Trans = mean(sse_Y_trans), Y_AIC = mean(sse_Y_AIC), Y_BIC = mean(sse_Y_BIC), full_Tra
rmse_values
```

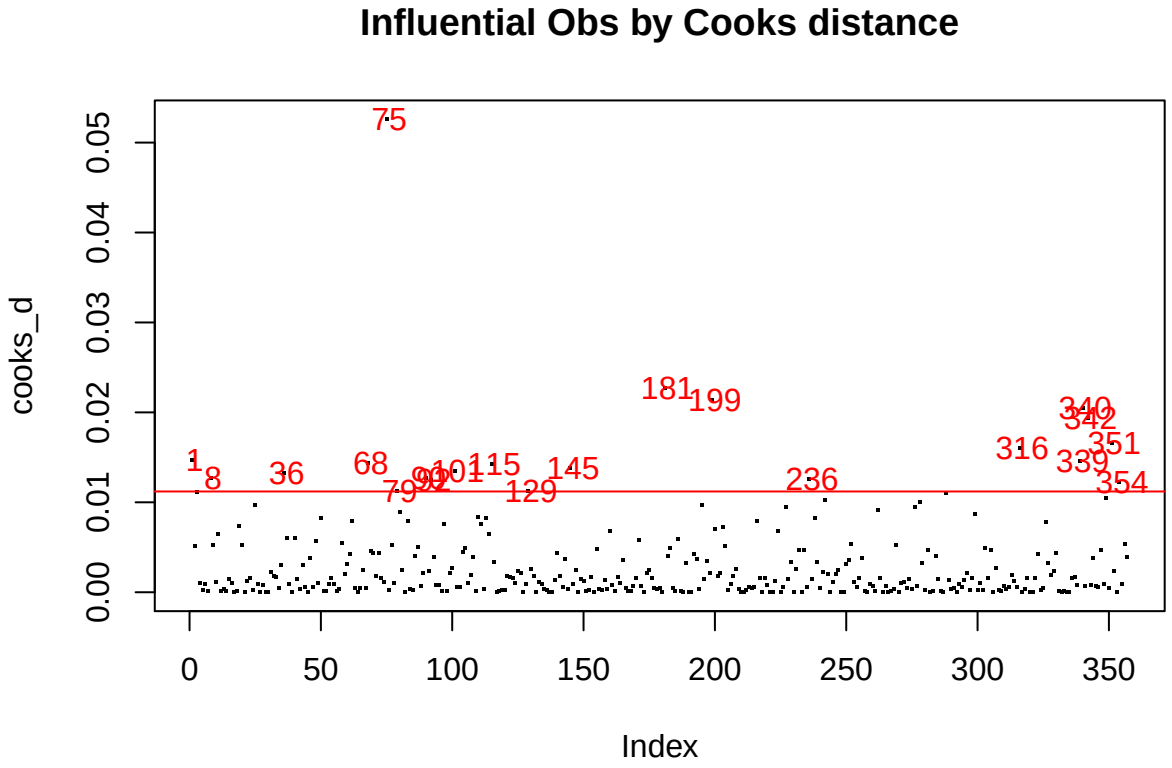
```
##    Y_Trans      Y_AIC      Y_BIC full_Trans   full_AIC   full_BIC
##  0.5508958  0.4549660  0.4445437  0.5339823  0.5278513  0.5121246
##      Ridge      Lasso
##  0.4392805  0.4415221
```

Appendix IX - Model Diagnostics

```
# Residuals plot for model with transformed Y and stepback AIC
plot(Y_model_trans_AIC, which=1)
```




```
# Cook's distance plot
n_obsv = nrow(final_data)
cooks_d = cooks.distance(Y_model_trans_AIC)
plot(cooks_d, pch=".", cex=2, main="Influential Obs by Cooks distance")
abline(h = 4/n_obsv, col="red")  # add cutoff line
text(x=1:length(cooks_d)+1, y=cooks_d, labels=ifelse(cooks_d>4/n_obsv, names(cooks_d),""), col="red")
```




```
# Dataset with influential point removed
data_noInf <- final_data[-75, ]
```

Appendix X - Re-simulation of Regular Models

```
# Number of simulations - we used 1000 for analysis
nsims = 10

sse_Y_AIC_NI = sse_Ridge_NI = sse_Lasso_NI = vector()


for (i in 1:nsims) {
# Build train and test
sample <- sample.int(n=nrow(data_noInf),size=floor(0.65*nrow(data_noInf)))
data_noInf.train <- data_noInf[sample,]
data_noInf.test <- data_noInf[-sample,]

# Y-Transform model
Y_trans_train_NI = lm(sqrt(G3)~., data=data_noInf.train)

# Stepwise on Y transform
Y_trans_AIC_train_NI = step(Y_trans_train_NI,direction="backward", trace = 0)

# Ridge/lasso data
X_NI <- model.matrix(Y_trans_train_NI)
X_NI <- X_NI[,-1]
y_NI <- as.matrix(sqrt(data_noInf.train$G3))

Y_trans_test_NI = lm(sqrt(G3)~., data=data_noInf.test)
X_NI_new <-model.matrix(Y_trans_test_NI)
X_NI_new <- X_NI_new[,-1]

ridge_train_NI <- glmnet(X_NI, y_NI, alpha = 0, lambda = ridge_cv_NI$lambda.min, standardize = TRUE)

lasso_train_NI <- glmnet(X_NI, y_NI, alpha = 1, lambda = lasso_cv_NI$lambda.min, standardize = TRUE)


# RMSE Formula
sse_Y_AIC_NI[i] = rmse(sqrt(data_noInf.test$G3), predict(Y_trans_AIC_train_NI, newdata=data_noInf.test,
sse_Ridge_NI[i] = rmse(sqrt(data_noInf.test$G3), predict(ridge_train_NI, newx = X_NI_new, type="respons
sse_Lasso_NI[i] = rmse(sqrt(data_noInf.test$G3), predict(lasso_train_NI, newx = X_NI_new, type="respons
}

# Table with means of RMSE values
rmse_values_NI <- c(Y_AIC = mean(sse_Y_AIC_NI), Ridge = mean(sse_Ridge_NI), Lasso = mean(sse_Lasso_NI))
rmse_values_NI
```

```
##     Y_AIC     Ridge     Lasso
## 0.4447094 0.4303020 0.4324649
```

Appendix XI - Simulation of Binary Models(Logistic and Neural Nets)

```
library(MASS)
library(neuralnet)
binary_data <- final_data

#We changed variable G3 to 1/0 values for pass/fail analysis. Score greater than 9 was set to pass(1) a

binary_data$G3 <- replace(final_data$G3>9,1,0)


# Instantiate "normalize" function
 normalize = function(x){
     return((x-min(x))/(max(x)-min(x)) )
 }

logistic_accuracy = nn_accuracy = rep(NA,10)

# We used 1000 simulations for analysis
for (i in 1:10){
sample <- sample.int(n=nrow(binary_data),size=floor(0.8*nrow(binary_data)), replace =F )
binary_data.train <- binary_data[sample,]
binary_data.test <- binary_data[-sample,]


logistic_model = glm(G3~.+I(poly(absences,3))-absences+I(poly(failures,2))-failures
                     +I(poly(studytime,2))-studytime+I(poly(goout,2))-goout, family = binomial(link = "
logistic_model_step = step(logistic_model,trace=F)
logisticpredict = round(predict(logistic_model_step,newdata=binary_data.test,type="response"))

# Running the Neural Netwwork
nn_model <- neuralnet(G3~., data=binary_data.train, hidden = c(6,1), linear.output = FALSE, threshold =
npredict = round(predict(nn_model, newdata=binary_data.test, type="response"))


logistic_accuracy[i] = sum(diag(table(binary_data.test$G3,logisticpredict)))/sum(table(binary_data.test
nn_accuracy[i] = sum(diag(table(binary_data.test$G3, npredict)))/sum(table(binary_data.test$G3, npredic

}

mean_lr_accuracy <- mean(logistic_accuracy)
mean_nn_accuracy <- mean(nn_accuracy)
binary_results <- cbind(logistic_Regress=mean_lr_accuracy,Neural_Nets = mean_nn_accuracy)
binary_results
```

```
##      logistic_Regress Neural_Nets
## [1,]              0.7   0.6111111
```

```
plot(nn_model)
```