Regression models with alpha

Introduction

Plotting ridge and LASSO regression on 2009 PISA exam dataset.

Goal

I would be running the Ridge and LASSO regression plots on Pisa2009 dataset. The analysis would be considering the entire Pisa2009 dataset and the final model obtained from Assignment 3.

I would also be running the residual analysis on the Pisa 2009 dataset. I would be plotting appropriate graphs for the residual analysis.

a - i) Ridge

Entire Dataset
Feature selection

Removing the variable X from Pisa2009 dataset. Creating a new dataset with name PisaEntire

```
PisaEntire <- Pisa2009[,c(-1)]
```

Dummy Variable

```
> Pisa2009$raceeth <- as.numeric(Pisa2009$raceeth)
```

In order to change the categorical variable into a numeric format, as.numeric function was applied on Pisa2009$raceeth

Preparation:

Creating x and y variable:

X variable will contain all the independent variable from column 1 to column 23

```
xentire <-as.matrix(PisaEntire[,1:23])
```

Y variable will contain the response variable from column 24 (readingScore)

```
yentire <- as.double(PisaEntire[,24])
```

Since we are using the cross validation function, we would be setting the seed.

```
set.seed(123)
```

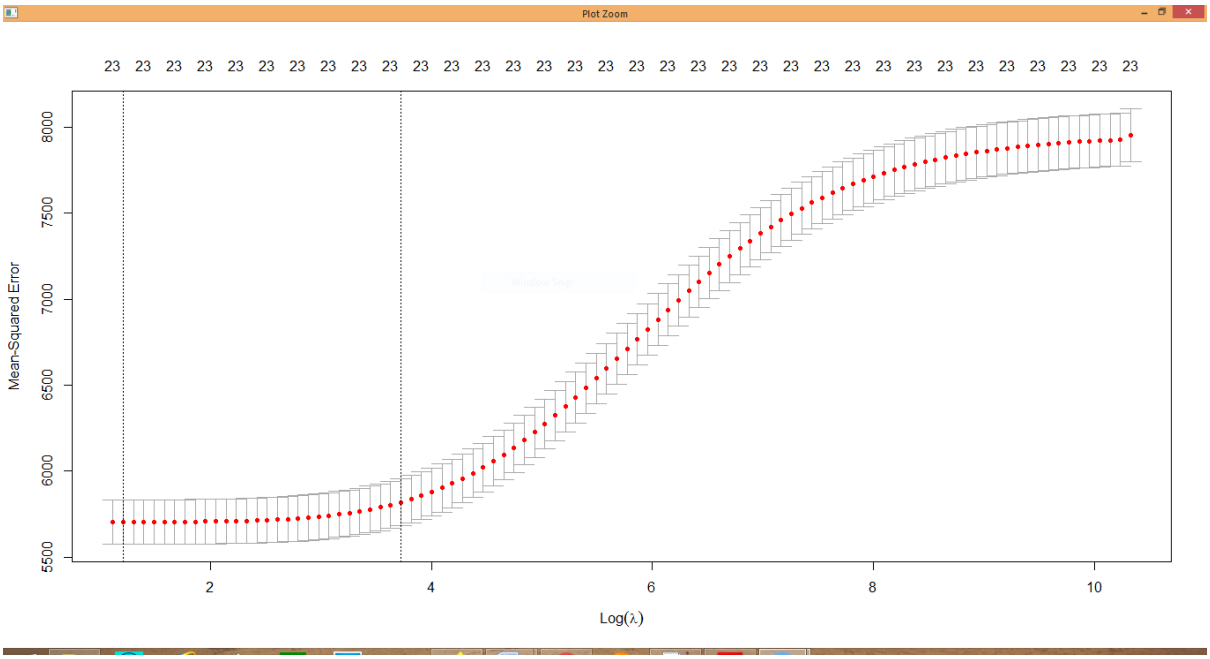
Ridge regression

Here we would be considering the entire Pisa2009 dataset.
Since we are using the alpha value of 0, the function will be using ridge regression model.

```
ridgeentire <-cv.glmnet(xentire,yentire,family="gaussian",alpha=0)
```

Plotting the ridge graph

```
plot(ridgeentire)
```



Minimum lambda value

The k fold cross validation has determined the value of lambda. The lambda has produced a minimum value of 3.0607 which minimizes the MSE.

```
ridgeentire$lambda.min
```

```
3.060793
```

Coefficient

The below plot determines the optimal values of betas for each variable.

```
coef(ridgeentire,s=ridgeentire$lambda.min)
```

```
24 x 1 sparse Matrix of class "dgCMatrix"
                                1
(Intercept)           104.795627976
grade                  26.625252430
male                  -12.421200721
raceeth                11.050302558
preschool              -0.757204068
expectBachelors        52.415732235
motherHS                4.329633587
motherBachelors        11.146311811
motherWork             -3.214957369
fatherHS               11.598706182
fatherBachelors        19.524737651
fatherWork              4.228829749
selfBornUS              0.153242908
motherBornUS          -12.740984510
fatherBornUS           -2.539214361
englishAtHome           9.651061708
computerForSchoolwork  21.922497707
read30MinsADay         32.753350588
minutesPerWeekEnglish   0.014337687
studentsInEnglish      -0.028304124
schoolHasLibrary       -1.057278830
publicSchool          -19.495016451
urban                  -2.790960002
schoolSize              0.006569603
```

Final Model from Assignment 3

<u>Feature selection</u>

Removing the variable X from Pisa2009 dataset. Creating a new dataset with name Pisa

```
Pisa <- Pisa2009[,c(-1)]
```

Dummy Variable

```
> Pisa2009$raceeth <- as.numeric(Pisa2009$raceeth)
```

In order to change the categorical variable into a numeric format, as.numeric function was applied on Pisa2009$raceeth

Final Model and variables:

```
Pisa <- Pisa[,-c(4,6,8,11,12,14,15,18,19,20,22)]
```

Preparation:

Creating x and y variable:

X variable will contain all the independent variable from column 1 to column 12

```
x <-as.matrix(Pisa [,1:12])
```

Y variable will contain the response variable from column 13 (readingScore)

```
y <- as.double(Pisa [,13])
```

Since we are using the cross validation function, we would be setting the seed.

```
set.seed(123)
```

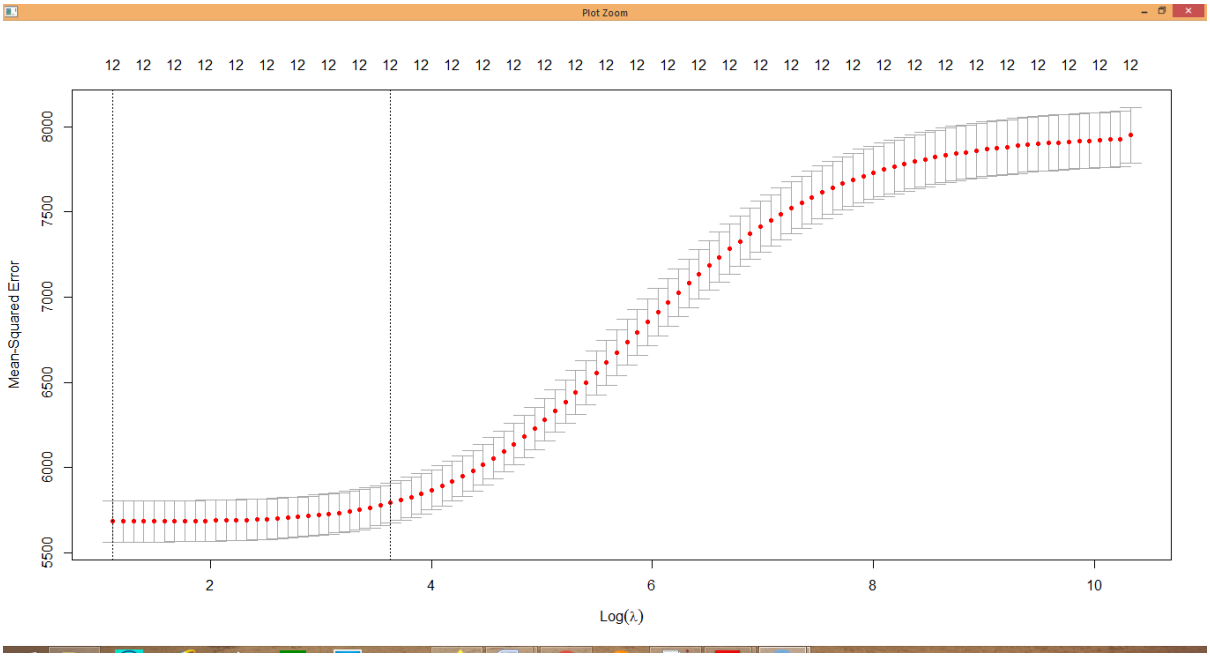
Ridge regression

Here we would be considering the Pisa2009 dataset using the final model from assignment 3. Since we are using the alpha value of 0, the function will be using ridge regression model.

```
ridge <-cv.glmnet(x,y,family="gaussian",alpha=0)
```

Plotting the ridge graph

```
plot(ridge)
```



Minimum lambda value

The k fold cross validation has determined the value of lambda. The lambda has produced a minimum value of 3.0607 which minimizes the MSE.

```
ridge$lambda.min
```

```
3.060793
```

Coefficient

The below plot determines the optimal values of betas for each variable.

```
coef(ridge,s=ridge$lambda.min)
13 x 1 sparse Matrix of class "dgCMatrix"
                                1
(Intercept)           107.161109216
grade                  26.796220998
male                  -12.303987359
raceeth                11.328621950
expectBachelors        52.496905918
motherBachelors        11.440155043
fatherHS               14.196860543
fatherBachelors        20.141408165
motherBornUS           -9.673849707
computerForSchoolwork  22.518037871
read30MinsADay         32.888412882
publicSchool          -17.389422078
schoolSize              0.005858202
```

Ridge regression handling multicollinearity:

Ridge regression is a technique developed for stabilizing the regression coefficients in the presence of multicollinearity.

The least squares estimates of the β coefficients may be subject to extreme roundoff error as well as inflated standard errors. Ridge regression is a modification of the method of least squares to allow biased estimators of the regression coefficients.

In presence of multicollinearity the variance of the least squares regression coefficients is quite large. The ridge regression introduces a small amount of bias in the ridge estimator, so that its mean square error is considerably smaller than the corresponding mean square error produced by least squares model. The ridge regression model produces a value of lambda (biasing constant).

As the value of lambda increases, the bias in the ridge estimates increases while the variance decreases. The idea is to choose lambda so that the total mean square error for the ridge estimators is smaller than the total mean square error for the least squares estimates.
The value of lambda can be determined using the ridge trace method. The smallest value of lambda is chosen so that the ridge estimates are stable.
If the optimal value of lambda has been selected, it will reduce variance and also solve other problems related to multicollinearity.

a - ii) Residuals

Using the linear regression model from Assignment 3.

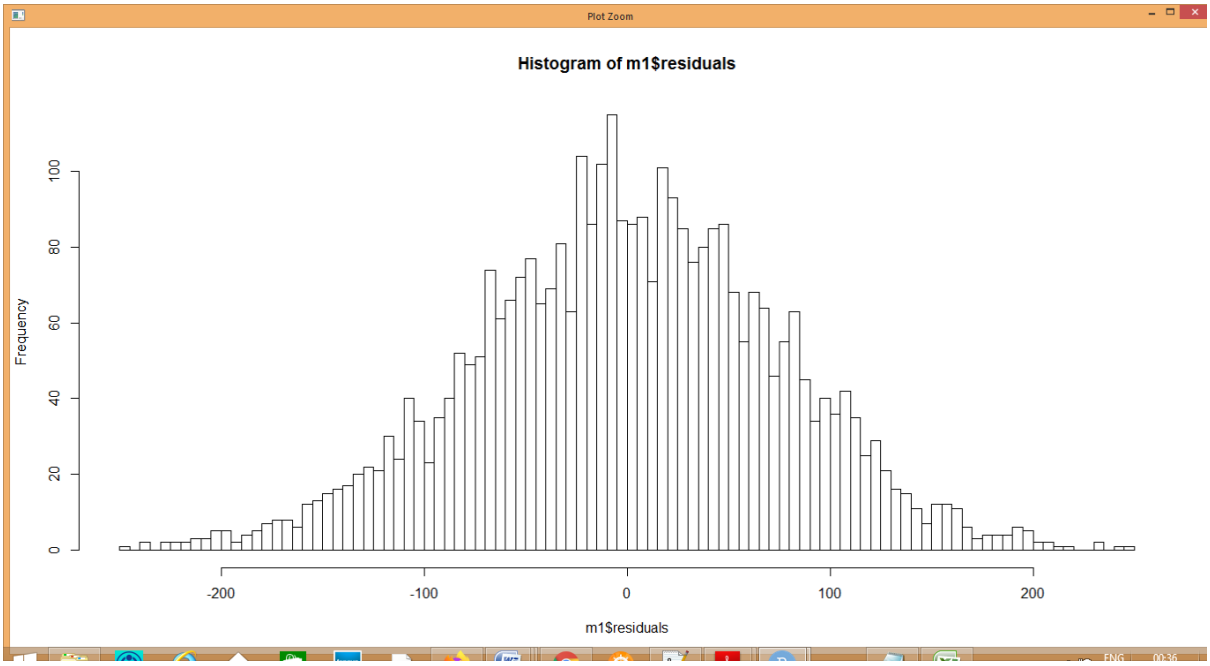
```
m1 <- lm(Pisa$readingScore ~ grade + male + raceeth  + expectBachelors  +
motherBachelors + fatherHS + fatherBachelors  + motherBornUS+ computerForSchoolwork +
read30MinsADay + publicSchool, data = Pisa)
```

The sum of all the residuals for the final model is ~ 0

```
sum(m1$residuals)
[1] 3.370082e-13
```

The histogram graph for the residuals shows a normal distribution pattern.

```
> hist(m1$residuals,breaks = 100)
```



Following are the general stats of the residuals.

Mean:

```
> mean(m1$residuals)
[1] 9.726173e-17
```

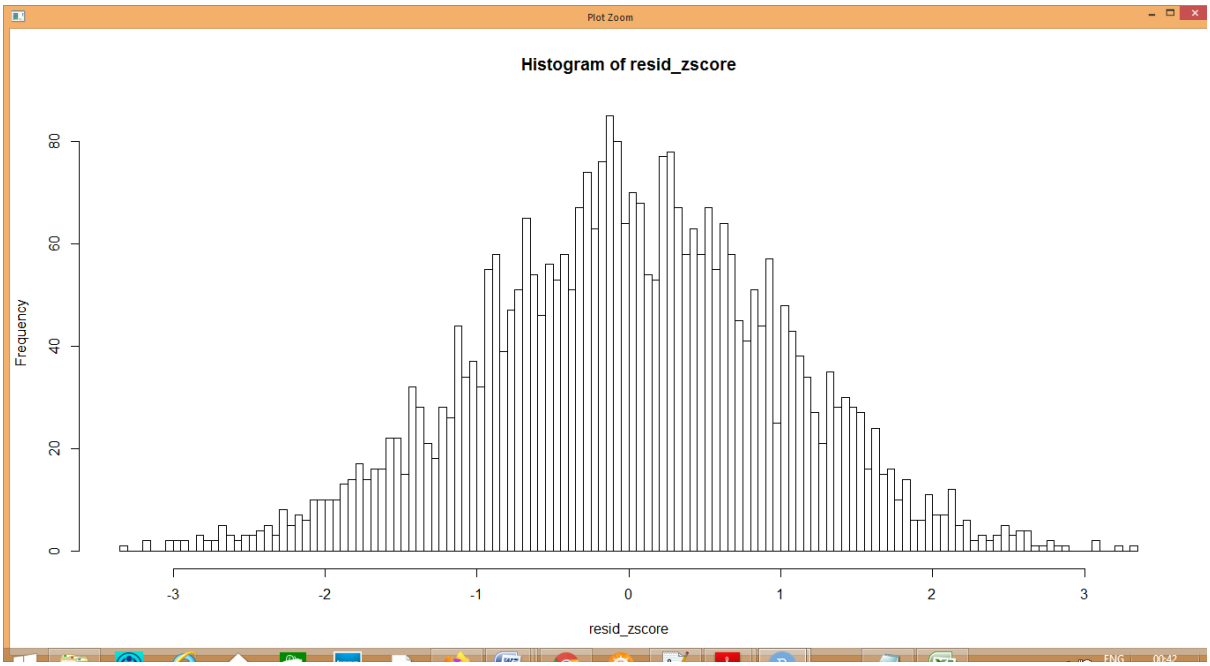
Standard Deviation:

```
> sd(m1$residuals)
[1] 75.24756
```

Histogram graph for z score normalisation.

```
> hist(resid_zscore,breaks = 100)
```

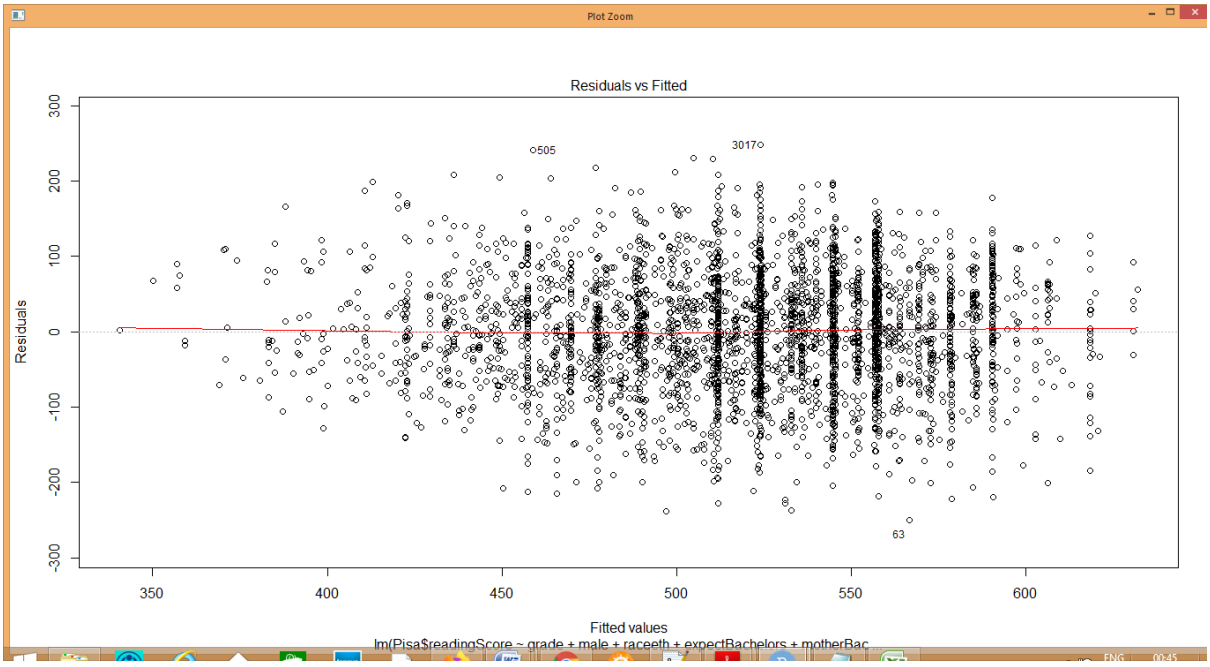
The histogram graph for the Z score shows a normal distribution pattern.




```
> plot(m1)
```

Residuals vs Fitted Plot

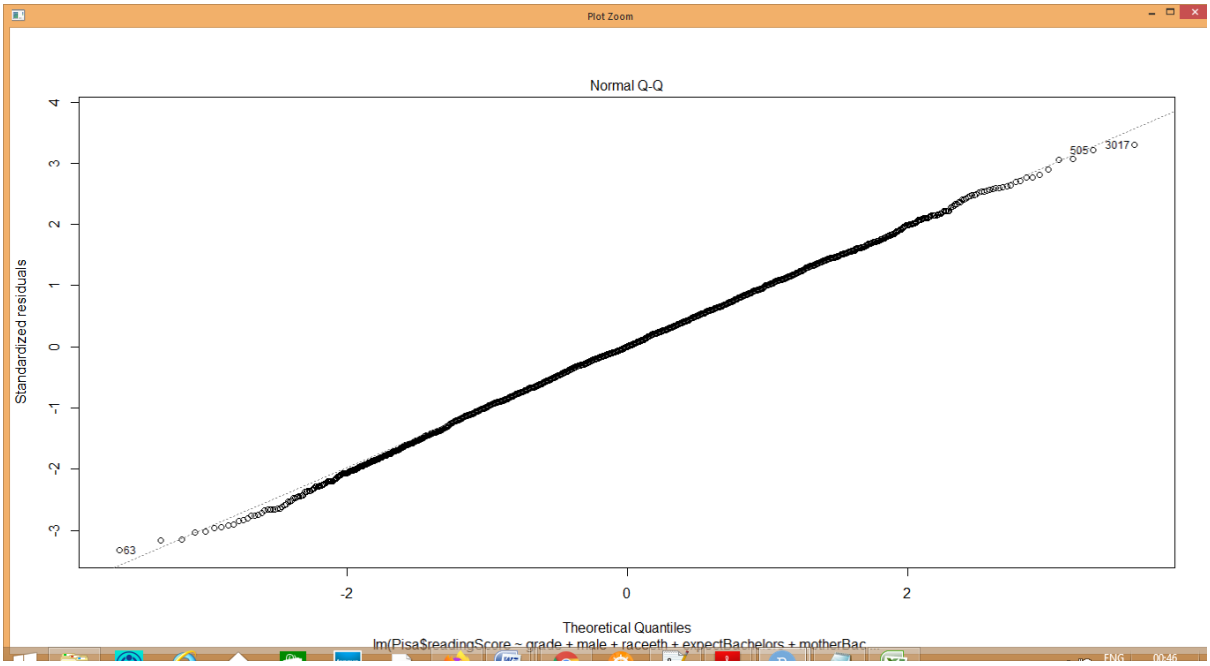
The residuals vs fitted graph shows normal formation.



Normal Q–Q (quantile-quantile) Plot

Residuals should be normally distributed. Since residuals follow close to a straight line on this plot, they are normally distributed.

The qq line plots all the data points across the line. 95% of the data is plotted on the line.

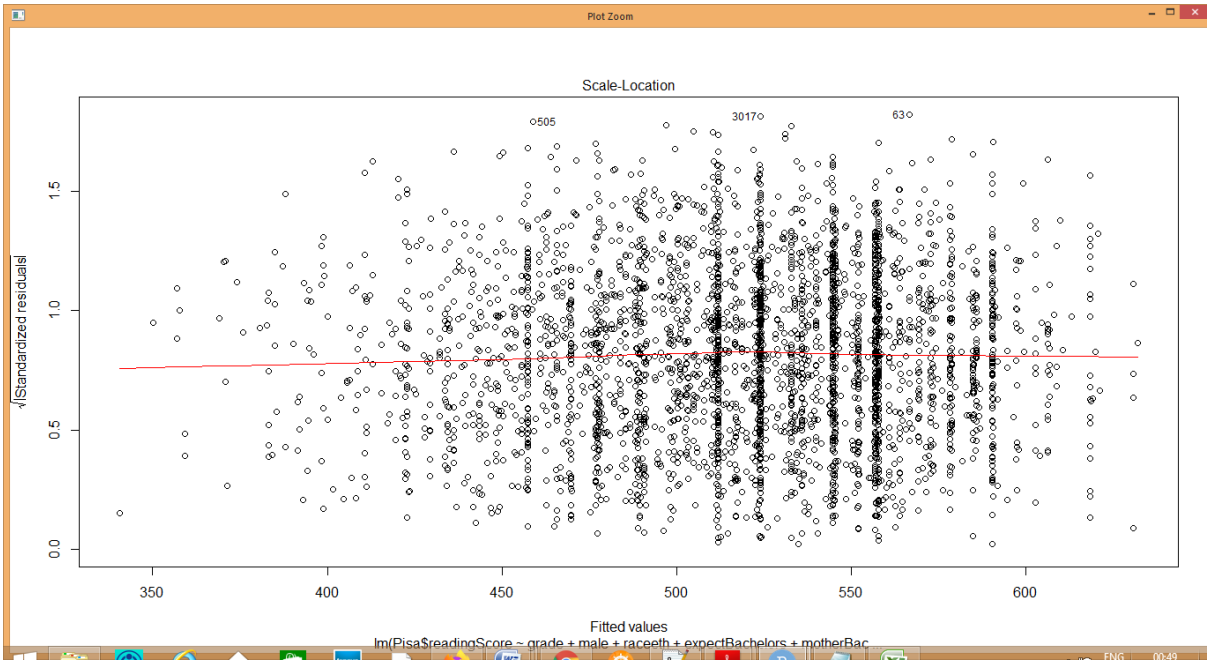


Standardised residuals vs Fitted values

This plot test the linear regression assumption of equal variance (homoscedasticity) i.e. that the residuals have equal variance along the regression line.

The residuals have equal variance (occupy equal space) above and below the line and along the length of the line.

The graph standardised residuals vs fitted values shows a normal variance.



Conclusion:

The sum of the squares of the errors is ~ 0.
The Residual vs. Fitted plot shows a linear pattern. This shows regression model is a good fit.
Around 95% of the data is plotted on the qq line.
The standardised residuals vs. fitted values shows homoscedasticity i.e equal variance across the line.
All the above plots show the linear regression model to predict the readingScore pattern in Pisa2009 dataset is a good fit.

```
m1 <- lm(Pisa$readingScore ~ grade + male + raceeth  + expectBachelors  +
motherBachelors + fatherHS + fatherBachelors  + motherBornUS+ computerForSchoolwork +
read30MinsADay + publicSchool, data = Pisa)
```

b) LASSO

Entire Dataset
Preparation:

Using the same variables xentire and yentire from Ridge model.

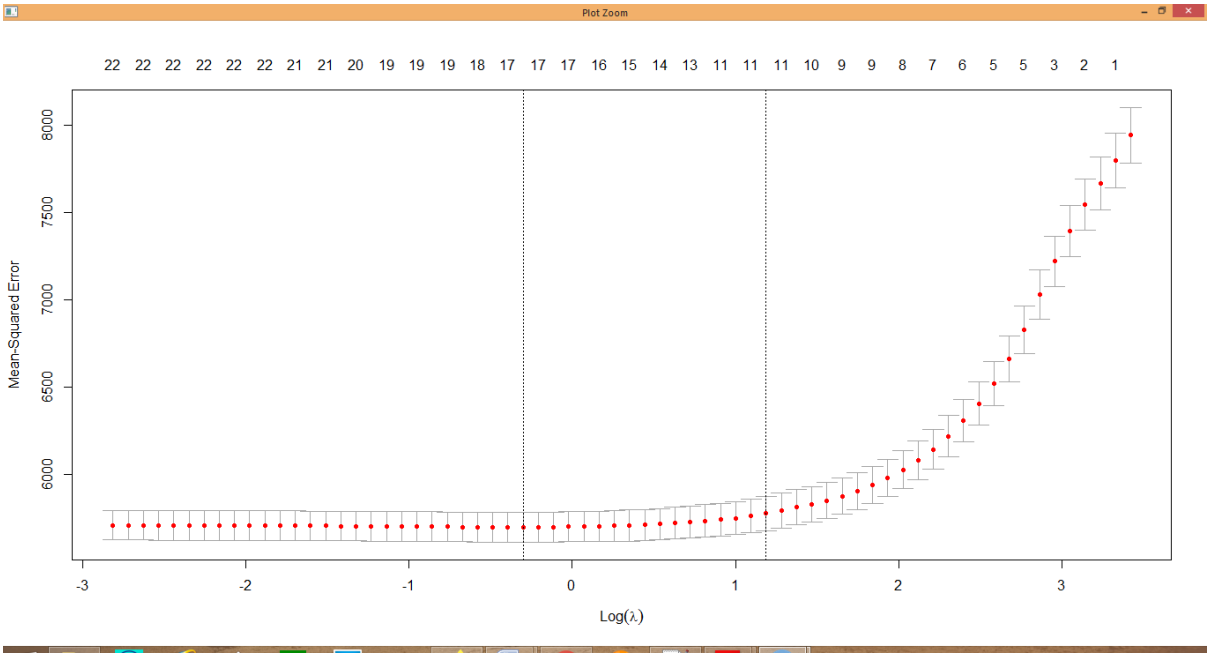
LASSO regression

Here we would be considering the entire Pisa2009 dataset.
Since we are using the alpha value of 1, the function will be using LASSO regression model.

```
lassoentire <-cv.glmnet(xentire,yentire,family="gaussian",alpha=1)
```

Plotting the LASSO graph

```
plot(lassoentire)
```



Minimum lambda value

The k fold cross validation has determined the value of lambda. The lambda has produced a minimum value of 0.7407 which minimizes the MSE.

```
lassoentire$lambda.min
```

```
0.740751
```

Coefficient

The below plot determines the value of betas for each variable. This can be used as a feature selection as the variables with no values can be dropped.

```
coef(lassoentire,s=lassoentire$lambda.min)
24 x 1 sparse Matrix of class "dgCMatrix"
                                    1
(Intercept)            104.220978244
grade                   26.588802139
male                   -11.138637393
raceeth                 11.003556493
preschool                .
expectBachelors         53.368399595
motherHS                 2.247083499
motherBachelors         10.303140609
motherWork              -1.133715110
fatherHS                10.538576635
fatherBachelors         20.139057518
fatherWork               2.356311594
selfBornUS               .
motherBornUS            -9.567331958
fatherBornUS             .
englishAtHome            4.281502373
computerForSchoolwork   21.067916048
read30MinsADay          32.458171021
minutesPerWeekEnglish    0.009529061
studentsInEnglish        .
schoolHasLibrary         .
publicSchool           -15.237055992
urban                    .
schoolSize               0.005176290
```

Final Model from Assignment 3

Feature selection

Removing the variable X from Pisa2009 dataset. Creating a new dataset with name Pisa

```
Pisa <- Pisa2009[,c(-1)]
```

Dummy Variable

```
> Pisa2009$raceeth <- as.numeric(Pisa2009$raceeth)
```

In order to change the categorical variable into a numeric format, as.numeric function was applied on Pisa2009$raceeth

Final Model and variables:

```
Pisa <- Pisa[,-c(4,6,8,11,12,14,15,18,19,20,22)]
```

Preparation:

Using the same variables x and y from Ridge model.

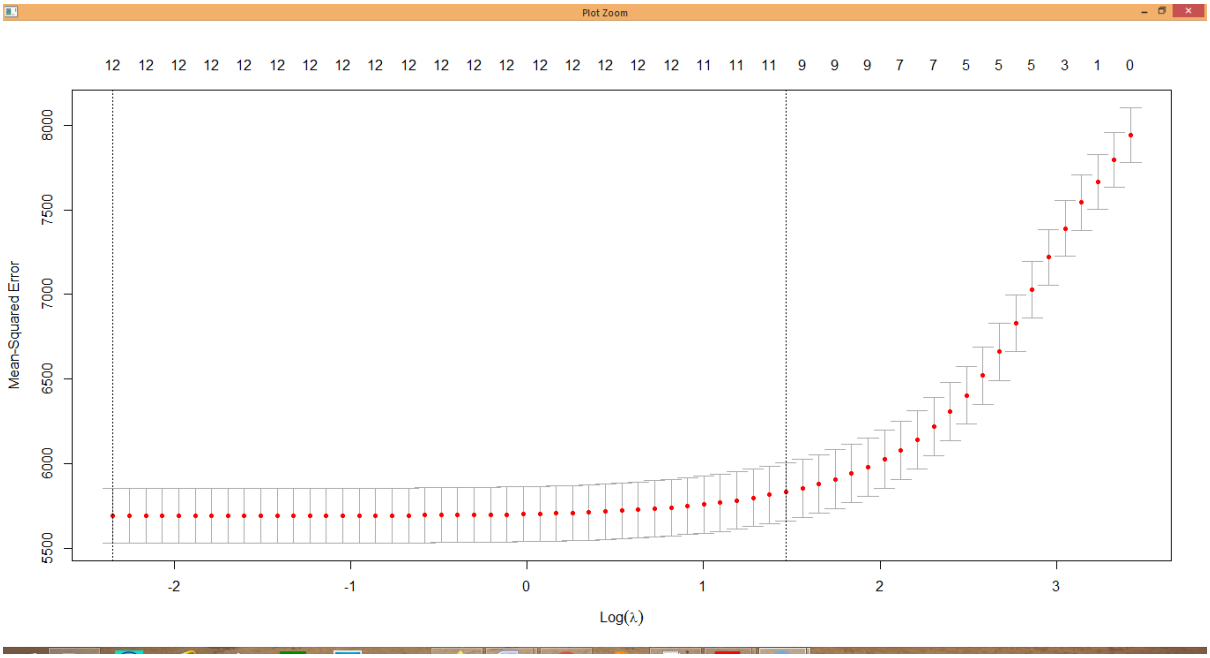
LASSO regression

Here we would be considering the entire Pisa2009 dataset.
Since we are using the alpha value of 1, the function will be using LASSO regression model.

```
lasso <-cv.glmnet(x,y,family="gaussian",alpha=1)
```

Plotting the LASSO graph

```
plot(lasso)
```



Minimum lambda value

The k fold cross validation has determined the value of lambda. The lambda has produced a minimum value of 0.7407 which minimizes the MSE.

```
lasso$lambda.min
```

```
0.09567168
```

Coefficient

The below plot determines the value of betas for each variable. This can be used as a feature selection as the variables with no values can be dropped.

```
coef(lasso,s=lasso$lambda.min)
13 x 1 sparse Matrix of class "dgCMatrix"
                                1
(Intercept)            98.166645502
grade                  27.381402249
male                  -12.261104667
raceeth                11.790055510
expectBachelors        53.829475498
motherBachelors        11.200072565
fatherHS               13.906851034
fatherBachelors        20.277156528
motherBornUS          -10.541107148
computerForSchoolwork  22.395067455
read30MinsADay         33.703275457
publicSchool          -17.404647522
schoolSize              0.005984522
```

LASSO: Feature Selection

The LASSO (Least Absolute Shrinkage and Selection Operator) is a regression method that involves penalizing the absolute size of the regression coefficients. Lasso regression performs L1 regularization that is it adds the penalty equivalent to the absolute value of the magnitude of the coefficients.

By penalizing (or equivalently constraining the sum of the absolute values of the estimates) you end up in a situation where some of the parameter estimates may be exactly zero. The larger the penalty applied, the further estimates are shrunk towards zero.

Therefore it can be used for feature selection.

c) Comparing the models produced by Ridge and LASSO

Entire dataset:

Comparing the ridge and LASSO model produced for the entire dataset with 23 independent variables and 1 response variable.
Ridge regression model introduces a bias to produce the model with minimum MSE. The model dete rmines the minimum value of lambda as 3.060793. The ridge regression model determines the new v alues for beta without dropping any of the variables.
LASSO regression model is used for feature selection. The model determines the minimum value of l ambda as 0.740751. The LASSO model has dropped 6 variables from the entire dataset. According to LASSO regression there are not relevant in the model.

Therefore the models produced by Ridge and LASSO models are not the same.

Final Model from Assignment 3

Comparing the ridge and LASSO model produced for the entire dataset with 12 independent variables and 1 response variable.
Ridge regression model introduces a bias to produce the model with minimum MSE. The model dete rmines the minimum value of lambda as 3.060793. The ridge regression model determines the new v alues for beta without dropping any of the variables.

LASSO regression model is used for feature selection. The model determines the minimum value of l ambda as 0.09567. The LASSO model has not dropped any variables from the dataset. Hence all the variables are relevant in the model.

The models and variables produced by Ridge and LASSO models are exactly the same.

Problem 2

Introduction

Plotting logistic model on REMISSION dataset.

Goal

I would be running the logistic model on REMISSION dataset.

a) Load REMISSION

Variable Transformation

```
remission$remiss <- factor(remission$remiss)
```

In order to change the numeric variable into a factor format, factor function was applied on remission$remiss.

Following is the summary of REMISSION dataset. There are 18 records having the remission$remiss value as 0 and 9 records having remission$remiss value as 1. Now the variable is transformed as factor variables with two values 0 and 1. (Highlighted in Yellow.)

```
> summary(remission)
 remiss       cell            smear           infil             li          
 0:18   Min.   :0.2000   Min.   :0.3200   Min.   :0.0800   Min.   :0.400  
 1: 9   1st Qu.:0.8250   1st Qu.:0.4300   1st Qu.:0.3350   1st Qu.:0.650  
        Median :0.9500   Median :0.6500   Median :0.6300   Median :0.900  
        Mean   :0.8815   Mean   :0.6352   Mean   :0.5707   Mean   :1.004  
        3rd Qu.:1.0000   3rd Qu.:0.8350   3rd Qu.:0.7400   3rd Qu.:1.250  
        Max.   :1.0000   Max.   :0.9700   Max.   :0.9200   Max.   :1.900  
     blast             temp       
 Min.   :0.0000   Min.   :0.980  
 1st Qu.:0.2275   1st Qu.:0.986  
 Median :0.5190   Median :0.990  
 Mean   :0.6889   Mean   :0.997  
 3rd Qu.:1.0625   3rd Qu.:1.005  
 Max.   :2.0640   Max.   :1.038  
```

b) Logistic Regression

```
> m <- glm(remiss ~ cell + smear + infil + li +  blast + temp, data = remission, family
= "binomial")
> summary(m)

Call:
glm(formula = remiss ~ cell + smear + infil + li + blast + temp,
    family = "binomial", data = remission)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.95165  -0.66491  -0.04372   0.74304   1.67069

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  58.0385    71.2364   0.815   0.4152
cell         24.6615    47.8377   0.516   0.6062
```

```
smear        19.2936    57.9500   0.333    0.7392
infil       -19.6013    61.6815  -0.318    0.7507
li            3.8960     2.3371   1.667    0.0955 .
blast         0.1511     2.2786   0.066    0.9471
temp        -87.4339    67.5735  -1.294    0.1957
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.372  on 26  degrees of freedom
Residual deviance: 21.751  on 20  degrees of freedom
AIC: 35.751

Number of Fisher Scoring iterations: 8
```

d) Explanation

The above model predicts remission (logical odds) based on the following independent variables.

- cell
- smear
- infil
- li
- blast
- temp

The Pr value in the above model can be interpreted in the same way as linear model.
Since the Pr (all variables) is above the level of significance (a = 0.05), **NONE** of the variables can be u sed to predict the value of remission.

Confidence Intervals

```
> confint(m)
Waiting for profiling to be done...
                 2.5 %     97.5 %
(Intercept)  -70.9683777 222.202990
cell         -27.7332544 138.404531
smear        -60.4544868 152.174139
infil       -159.7565104  67.536927
li             0.1944541   9.526820
blast         -4.5238625   4.715064
temp        -244.7720744  24.913187
There were 26 warnings (use warnings() to see them)
```

The `confint` function if used to predict the 95% interval for each variable.
Example: The value of cell variable for around 95% of the data will be between `-27.733254 and 138 .40453.`

Variable Probability

```
> exp(coef(m))-1
  (Intercept)          cell         smear         infil            li         blast
 1.606182e+25  5.133014e+10  2.393828e+08 -1.000000e+00  4.820343e+01  1.631040e-01
         temp
-1.000000e+00
```

The above function determines the probability of remission for each variable.

c) glm vs. lm

The generalized linear model (GLM) is a flexible generalization of ordinary linear regression that allo ws for response variables that have error distribution models other than a normal distribution. The G LM generalizes linear regression by allowing the linear model to be related to the response variable

via a link function and by allowing the magnitude of the variance of each measurement to be a functi on of its predicted value.
A GLM is a more general version of a linear model: the linear model is a special case of a Gaussian GL M with the identity link.
"glm" in that the relationship between the regressors and the target is not strictly linear, but under a transformation (the link function) they are.

- lm models (ordinary linear regression) in R are fit using ordinary least squares regression (OL S) which assumes the error terms of your model are normally distributed (i.e. family = gaussi an) with mean zero and a common variance. You cannot run a lm model using other link func tions. In lm fit model assumed the errors could take on any value on the real number line.

- glm uses the error terms in the model which are binomial using a logit link function. This ess entially constrains the model so that it assumes no constant error variance and it assumes th e error terms can only be 0 or 1 for each observation.
  glm accepts a named argument for family.
  Example: `glm(y ~ x, family = binomial)`

lm fits models of the form: `Y = XB + e` where `e~Normal( 0, s^2 ).`

glm fits models of the form `g(Y) = XB + e,` where the function `g()` and the sampling distributi on of `e` need to be specified. The function `'g'` is called the `"link function". The` default li nk function for glm is the "identity function" such that `g(Y) = Y,` and the default error distributio n is Normal.
Using these defaults glm is fitting the same model fit by lm.

The most common glm's are poisson regression and logistic regression.
Technically "normal" regression is the *most* common glm.