Homework_4

Sai Saketh Boyanapalli

October 23, 2017

```
library(mlbench)
library(car)
library(EnvStats)
library(asbio)
library(MASS)
library(outliers)
library(ggplot2)
library(reshape2)
library(Amelia)
library(mice)
library(HSAUR2)
library(VIM)
library(dplyr)
library(e1071)
library(tidyr)
library(fitdistrplus)
library(stats)
library(robustbase)
library(gridExtra)
library(memisc)
library(pls)
library(lars)
library(glmnet)
library(caret)
library(elasticnet)
library(lattice)
```

Question 1 Predicting House Prices

```
housedata<- read.csv("housingData2.CSV") # reading data into r
```

(a)

Modelling using Step wise variable selection.

Formula for the model using STEP.

```
ols_step <-  lm(formula(model_step), data = trainx)
```

i)

The values of AIC, BIC, Adjusted R - squared, RMSE, VIF and Coefficeints for the best fitter model.

```
## AIC is  -600.7616

##
## BIC is -110.9173

##
## Adjusted R squared is  0.943182

##
## root mean square error 0.1497486

##
##  Average value of VIF is 22.78117

##
## value of VIF's
##  9.404026 24.12889 2.495081 1.527062 2.924208 2460.064 2.393517 57.3186
4.005421 2.001415 8.54787 1.641763 11.80586 4.108487 2.02449 10.6177 1.805131
2.866861 12.12014 7.481999 9.606891 5.923958 5.008503 2.122718 2.05235
2.363583 5.877905 4.862137 1.341375 2.25471 3.187851 2.468365 3.783411
2.913648 1.799004 5.860848 5.651844 1.787858 1.406022 1.331511 1.317934
1.098818 1 3 1 3 2 17 5 4 1 1 1 2 7 2 2 3 2 3 5 1 5 1 1 2 1 3 1 1 1 1 1 1 1 5
1 1 1 2 1 1 1 1 3.066598 1.699898 1.579583 1.073106 1.307682 1.258154
1.091198 1.658773 2.001355 1.414714 2.923674 1.131951 1.19283 1.423707
1.192831 1.482535 1.159117 1.191885 1.283367 2.735324 1.253887 2.433918
2.237969 1.207044 1.432602 1.154149 2.424439 2.205025 1.158178 1.501569
1.785455 1.571103 1.945099 1.112868 1.34127 2.420919 2.377361 1.156334
1.185758 1.153911 1.148013 1.048245
```

RMSE

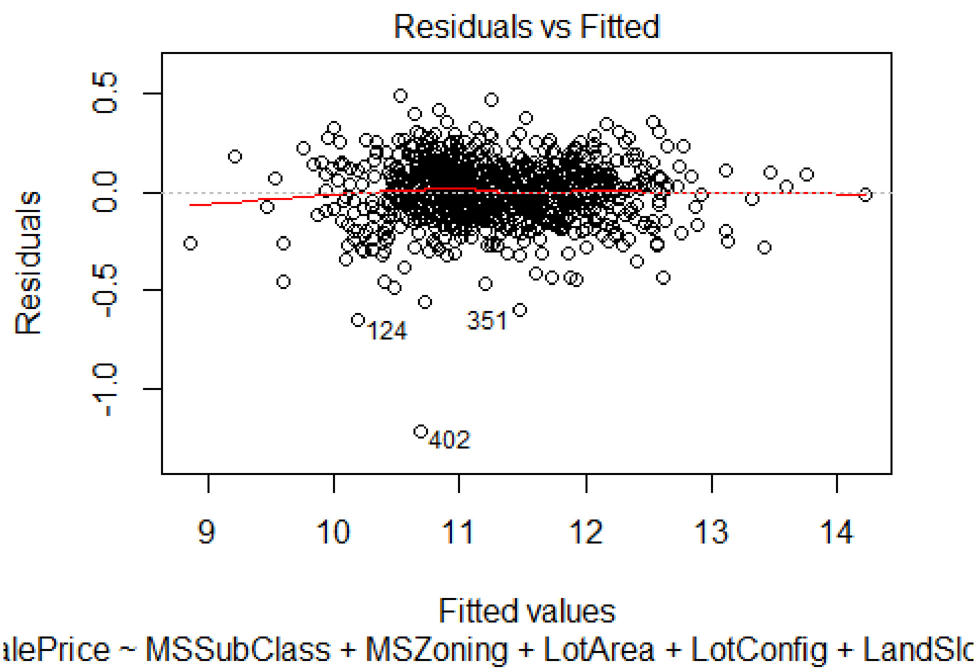
```
## RMSE test of Best OLS Model in test data is 0.02271333
```

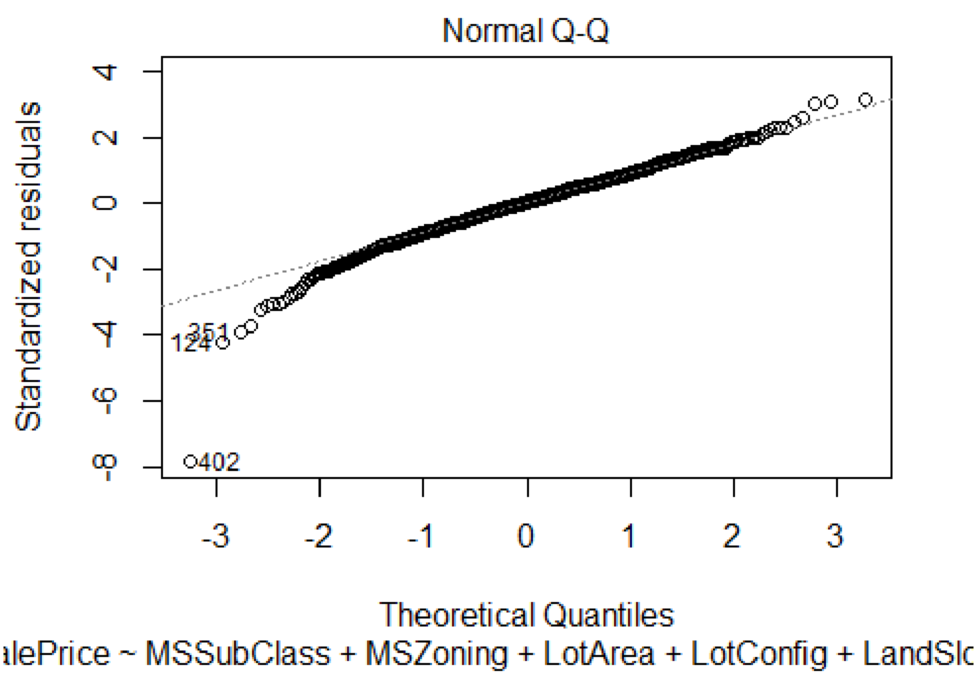
Regression Coefficient

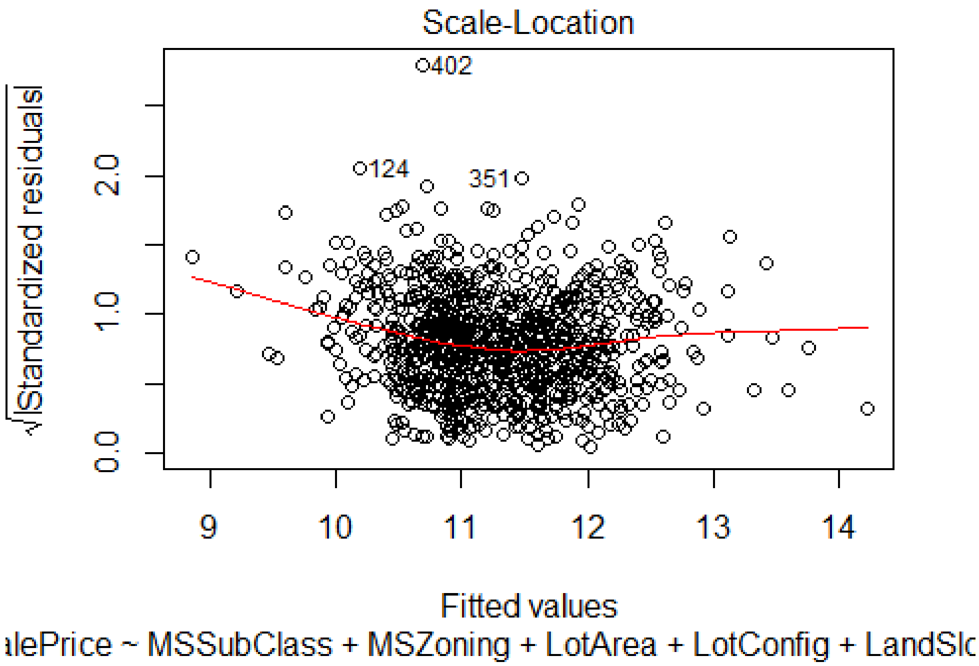
```
##
## Coefficients are  -0.6606962 -0.0006413371 -0.05297478 0.01184218 -
0.1375932 4.904634e-06 0.04865682 -0.01771564 -0.04318647 -0.01549592 -
0.2401414 -0.05137778 -0.1380103 0.1706909 -0.1750943 -0.1528789 0.06585326 -
0.1930265 -0.1283806 -0.1189768 0.01862615 -0.146595 -0.1360071 -0.1119907 -
0.1548541 -0.171919 0.005943804 -0.06945889 0.01892565 0.07831834 -0.04850595
-0.007801132 -0.01723258 0.1398117 -0.0284001 -0.1311081 -0.02244679
0.09213841 0.091555 0.004878111 0.02856318 0.1663936 -0.1165134 -0.1223739 -
0.09772987 -0.06080256 -0.1264465 -0.08080759 -0.1220562 -0.01588021 -
0.2790688 0.05615667 0.08653359 0.03345625 -0.0305428 0.07132025 -0.03692958
-0.09313566 0.06934646 -0.03701382 -0.03116245 -0.04108469 0.02864525 -
0.05804796 -0.04769372 -0.01687636 0.0002809975 -0.07859491 0.09369669 -
0.0614917 -0.05233371 -0.006066574 0.0002985147 0.0001514008 -0.04062405 -
```

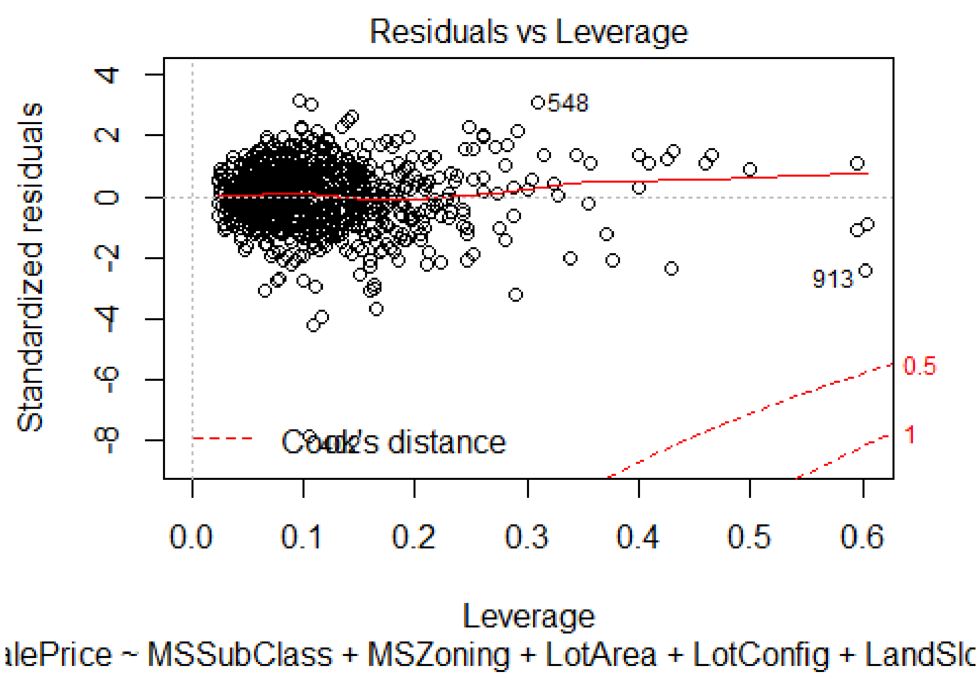
```
0.03827448 0.07209214 -0.1332036 -0.1247816 -0.009894607 0.0005486908
0.0005613421 0.0003549716 0.0389655 0.0279809 -0.02147694 -0.1025884 -
0.1701369 0.06313656 0.1472239 0.05152773 0.1755734 0.0449436 0.06810895
0.0001014062 0.04443946 -0.05467154 0.000153155 0.0003504378 0.0004101481
0.0004141263
```

ii)


Residuals vs Fitted
Residuals
0.5
0.0
-0.5
-1.0
124
351
402
9
10
11
12
13
14
Fitted values
alePrice ~ MSSubClass + MSZoning + LotArea + LotConfig + LandSlc


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
-6
-8
351
124
402
-3
-2
-1
0
1
2
3
Theoretical Quantiles
alePrice ~ MSSubClass + MSZoning + LotArea + LotConfig + LandSlc





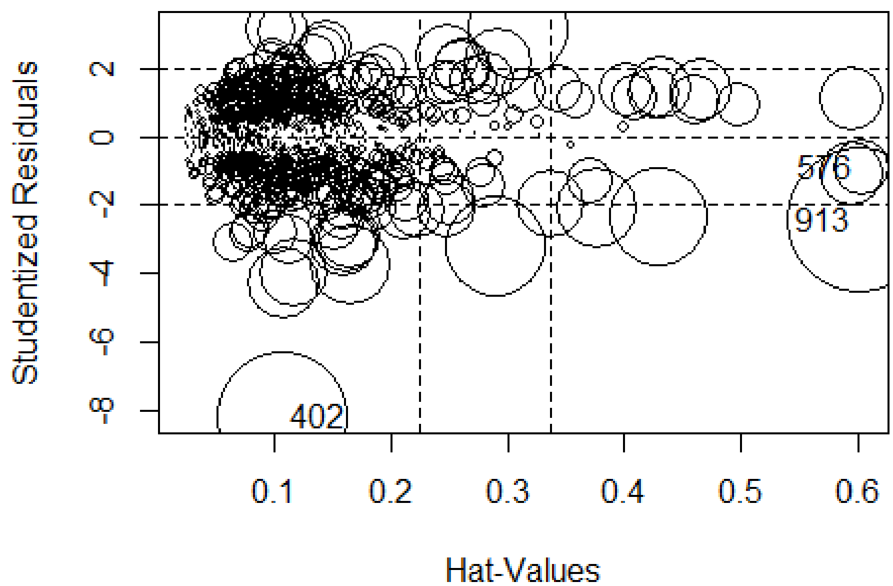
From the above plots

We can see that residuals pattern seems to be random.

In case of our Normal QQ Plot except the few indicated outliers it also looks to follow normal distribution.

Standardized Residual vs Fitted shows observation no 402 as outlier.

Residual vs leverage also shows 402 as possible outlier

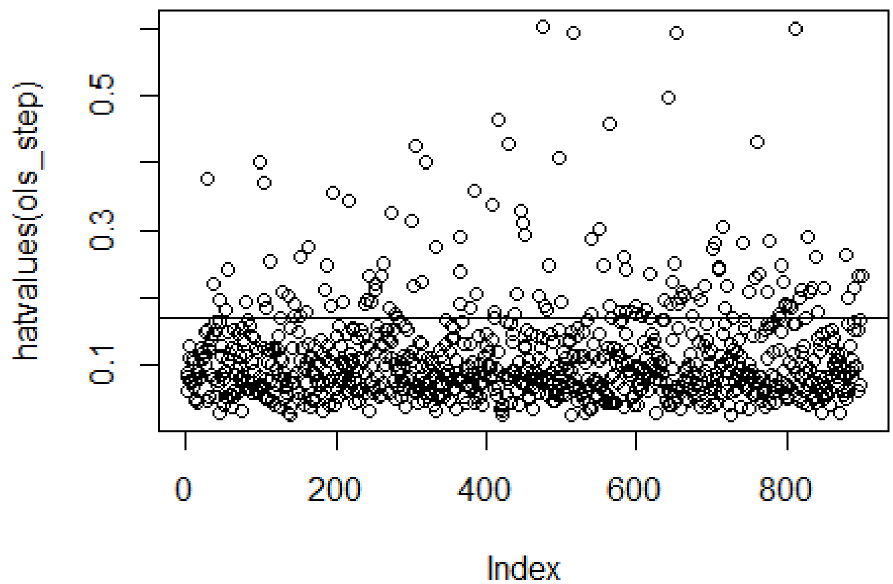



```
##         StudRes       Hat      CookD
## 402 -8.1512112 0.1056354 0.07181719
## 576 -0.9206129 0.6038929 0.01279567
## 913 -2.4255113 0.6017357 0.08747289
```

From the above plot we can see that the observations 402 might be an outlier and 913, 576 high hat values. combined these these points might be infuential to our model.

generally the points with hat values above 2(p+1)/n are considered to have leverage with the fitted model.

for our model the hat value is 2(p+1)/n = 0.15

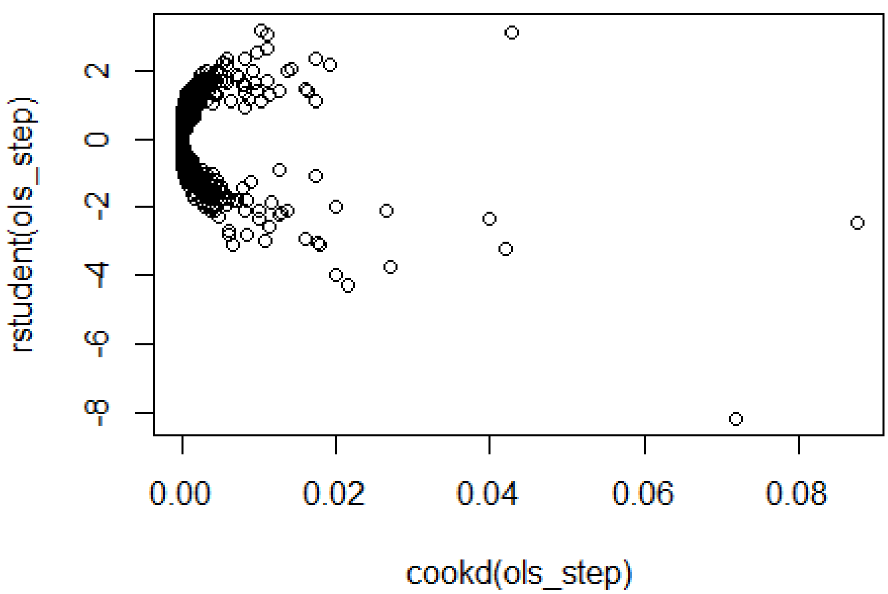


From the above plot we can see there are quite a few observations which are above the line. we might want to look at them. they can be good/bad leverage points.

```
##       128       137       145       154       155       179       199 
## 0.3772558 0.2212801 0.1953247 0.1802924 0.2420808 0.1937013 0.4008485 
##       203       205       207       213       228       237       244 
## 0.1961819 0.3704101 0.1828649 0.2520845 0.2092121 0.2031109 0.1901600 
##       246       253       254       261       264       284       289 
## 0.1711702 0.2606234 0.1719218 0.1765638 0.2758328 0.2099721 0.2460149 
##       292       295       310       318       338       342       343 
## 0.1882376 0.3546761 0.1937605 0.3446389 0.1909709 0.1959800 0.2310125 
##       346       352       353       361       364       372       373 
## 0.1918051 0.2194270 0.2096233 0.2316954 0.2510288 0.1851043 0.3242703 
##       380       401       403       405       415       419       434 
## 0.1737027 0.3146835 0.2179711 0.4248757 0.2222886 0.3998872 0.2733353 
##       464       465       466       479       485       488       508 
## 0.2370939 0.2881753 0.1904091 0.1850377 0.3574812 0.2043187 0.3378706 
##       510       515       530       538       540       547       548 
## 0.1763940 0.4648460 0.4284859 0.1735502 0.2054045 0.3286718 0.3096568 
##       550       570       576       577       580       584       597 
## 0.2916693 0.2011865 0.6038929 0.1874006 0.1814851 0.2478760 0.4084434 
##       600       617       639       640       651       656       664 
## 0.1915933 0.5948319 0.2878405 0.1745808 0.3014537 0.2462720 0.1868491 
##       665       679       684       686       687       695       698 
## 0.4596983 0.1788834 0.2600443 0.1783467 0.2424559 0.1782165 0.1750755 
##       700       708       714       719       738       741       744 
## 0.1880305 0.1705741 0.1797815 0.2355652 0.1947429 0.4986770 0.1884301
```

```
##       747       750       754       757       758       761       764 
## 0.2225406 0.2516039 0.5948319 0.1908194 0.1986072 0.2079017 0.2023712 
##       774       779       784       790       802       805       809 
## 0.1717835 0.2062262 0.1968728 0.2175615 0.2713773 0.2800816 0.2425871 
##       810       814       818       819       841       850       858 
## 0.2442808 0.3050880 0.1840720 0.2162618 0.2799392 0.2078802 0.2288885 
##       861       862       875       876       890       894       895 
## 0.4313249 0.2345339 0.2093091 0.2831489 0.1745080 0.2463304 0.2232549 
##       897       898       903       906       913       917       919 
## 0.1895583 0.1862888 0.1870555 0.1852604 0.6017357 0.1980204 0.2118545 
##       927       930       933       939       942       950       979 
## 0.2905050 0.2076961 0.2127404 0.2607222 0.1778845 0.2134293 0.2613358 
##       982       991       996      1000 
## 0.1982848 0.2126046 0.2307197 0.2316198
```

These are the observations with large leverage.



We can see from the above graph there are no influential points. since cook's d is less than 1.

```
##       rstudent unadjusted p-value Bonferonni p
## 402 -8.151211         1.3904e-15   1.2514e-12
## 124 -4.280384         2.0925e-05   1.8832e-02
```

Outlier test also indicates observation 402 as an outlier.

Now that we see some influential points in the data indicated by outlierTest, Hat values and studentized Residual, We will remove those points from the train data and see if there is any improvement with our model.

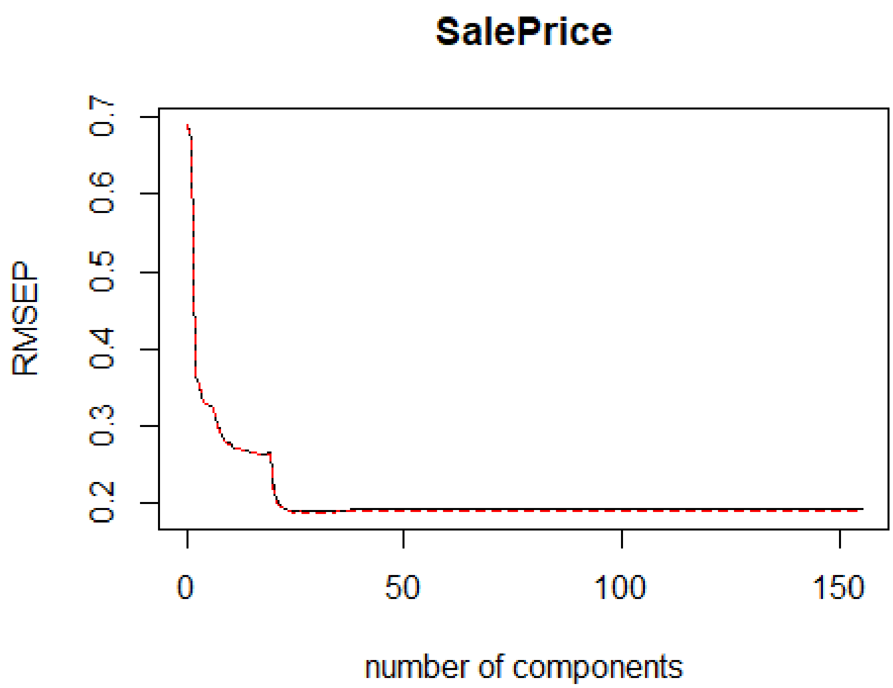
```
newcompletedData <- compdata[-c(402, 913, 576),]
newtrainx<-newcompletedData[101:997,]
newvalx<-newcompletedData[1:100,]
newy<-newcompletedData[101:997,69]

ols_step_modified <-  lm(formula(model_step), data = newtrainx)
ols_step_sum_modified <- summary(ols_step_modified)

ols_step_modified.pred <- predict(ols_step_modified, newvalx)
cat("RMSE test of Best OLS Model By removing outliers in test data
is",sqrt(mean(ols_step_modified.pred - (valx$SalePrice))^2))

## RMSE test of Best OLS Model By removing outliers in test data is 0.02198

cat("Adjusted R - Squared is", ols_step_sum_modified$adj.r.squared)

## Adjusted R - Squared is 0.9472412
```

b) PLS Model

PLS model is done on the best model found in part (a)

```
## RMSE of PLS in train data is  0.1825895
```

This graph shows RMSE vs Number of Components.



Final PLS Model

```
## RMSE test of PLS in test data is 0.001351722
```

We have taken 20 principal components to predict the value of Sale Price cause these 20 components explains the most variance and found that the value of

CV RMSE is 0.00110

Which is very good and better than that we found for OLS.

(c) LASSO Model

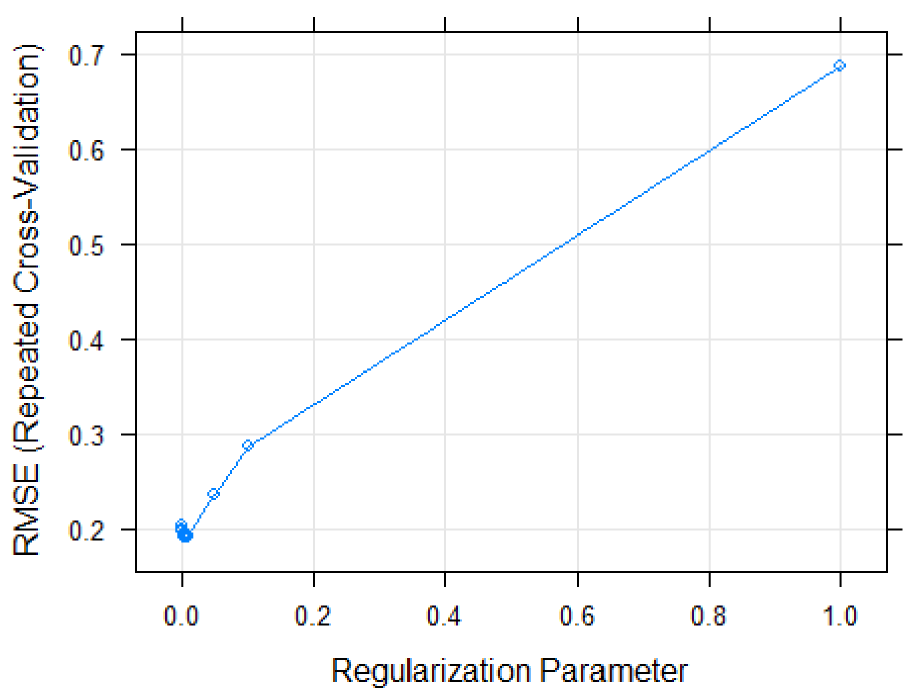
Penality value with RMSE

The parameter tuning with RMSE Chart

```
## glmnet
##
## 900 samples
##  68 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 721, 721, 719, 720, 719, 720, ...
## Resampling results across tuning parameters:
```

```
##
##   lambda   RMSE       Rsquared   MAE
##   0.00010  0.2026318  0.9148473  0.1466719
##   0.00050  0.1990426  0.9174964  0.1447688
##   0.00075  0.1973754  0.9187268  0.1439008
##   0.00100  0.1961298  0.9196373  0.1433012
##   0.00200  0.1928312  0.9220859  0.1417337
##   0.00300  0.1911689  0.9233090  0.1406803
##   0.00400  0.1902379  0.9239645  0.1402344
##   0.00500  0.1896857  0.9243645  0.1399217
##   0.00600  0.1895100  0.9244969  0.1397474
##   0.00700  0.1897275  0.9243550  0.1398814
##   0.00800  0.1902120  0.9240390  0.1401905
##   0.00900  0.1908819  0.9236029  0.1406474
##   0.01000  0.1916647  0.9230839  0.1411848
##   0.05000  0.2353939  0.8937224  0.1728258
##   0.10000  0.2872802  0.8621243  0.2128747
##   1.00000  0.6881096        NaN  0.5421191
##
## Tuning parameter 'alpha' was held constant at a value of 1
## RMSE was used to select the optimal model using  the smallest value.
## The final values used for the model were alpha = 1 and lambda = 0.006.
```

Plot of RMSE Vs Regulation Parameter.



Except the first few the RMSE is increasing with the Regularization Parameter.

variables with non - zero lasso_coefficeintsw

```
## 156 x 1 sparse Matrix of class "dgCMatrix"
##                                1
## (Intercept)          2.977195e-01
## MSSubClass          -2.471513e-04
## MSZoningRH          -6.255267e-02
## MSZoningRL           .
## MSZoningRM          -1.281979e-01
## LotFrontage          .
## LotArea              3.932376e-06
## LotShapeIR2          .
## LotShapeIR3         -8.271001e-03
## LotShapeReg         -6.657054e-04
## LandContourHLS       .
## LandContourLow       .
## LandContourLvl       .
## LotConfigCulDSac     3.709897e-02
## LotConfigInside      .
## LotConfigother       .
## LandSlopeMod         .
## LandSlopeSev         .
## NeighborhoodClearCr  5.021723e-02
## NeighborhoodCollgCr  .
## NeighborhoodCrawfor  2.279617e-01
## NeighborhoodEdwards -4.622111e-02
## NeighborhoodGilbert  .
## NeighborhoodIDOTRR   8.087881e-02
## NeighborhoodMitchel -1.517487e-02
## NeighborhoodNAmes    .
## NeighborhoodNoRidge  .
## NeighborhoodNridgHt  1.035576e-01
## NeighborhoodNWAmes   .
## NeighborhoodOldTown -3.713379e-02
## Neighborhoodother    .
## NeighborhoodSawyer  -5.226839e-03
## NeighborhoodSawyerW -9.918036e-03
## NeighborhoodSomerst  8.683523e-02
## NeighborhoodTimber   1.084054e-02
## Condition1Feedr      .
## Condition1Norm       7.194048e-02
## Condition1PosA       .
## Condition1PosN       .
## Condition1RR         .
## BldgType2fmCon       .
## BldgTypeDuplex      -5.120482e-02
## BldgTypeTwnhs       -1.131944e-01
## BldgTypeTwnhsE      -2.220355e-02
## HouseStyle1.5Unf     .
## HouseStyle1Story     .
```

```
## HouseStyle2.5Fin    -8.441911e-02
## HouseStyle2.5Unf    -2.522185e-03
## HouseStyle2Story     .
## HouseStyleSFoyer     .
## HouseStyleSLvl       .
## OverallQual          1.024510e-01
## OverallCond          7.923947e-02
## YearBuilt            3.695631e-03
## YearRemodAdd         6.068669e-04
## RoofStyleHip         1.612974e-02
## RoofStyleother       1.045931e-01
## Exterior1stCemntBd   .
## Exterior1stHdBoard   .
## Exterior1stMetalSd   .
## Exterior1stother     .
## Exterior1stPlywood  -7.853116e-03
## Exterior1stVinylSd   .
## Exterior1stWd Sdng  -1.483169e-02
## Exterior2ndCmentBd   .
## Exterior2ndHdBoard  -2.040737e-02
## Exterior2ndMetalSd   .
## Exterior2ndother     .
## Exterior2ndPlywood  -1.147384e-02
## Exterior2ndVinylSd   5.115342e-03
## Exterior2ndWd Sdng   .
## Exterior2ndWd Shng   .
## MasVnrTypeBrkFace    .
## MasVnrTypeNone       .
## MasVnrTypeStone      .
## MasVnrArea           .
## ExterQualAvg        -1.179440e-02
## ExterQualBelowAvg   -9.438818e-02
## ExterCondAvg         1.219318e-02
## ExterCondBelowAvg    .
## FoundationCBlock     .
## Foundationother      .
## FoundationPConc      5.151533e-02
## BsmtQualAvg          .
## BsmtQualBelowAvg     .
## BsmtCondAvg          .
## BsmtCondBelowAvg    -7.819113e-02
## BsmtExposureGd       7.645481e-02
## BsmtExposureMn       .
## BsmtExposureNo      -1.967026e-02
## BsmtFinType1BLQ      .
## BsmtFinType1GLQ      2.820133e-02
## BsmtFinType1LwQ      .
## BsmtFinType1Rec     -8.841997e-03
## BsmtFinType1Unf      .
## BsmtFinSF1           1.228805e-04
```

```
## BsmtFinType2BLQ     -2.456004e-02
## BsmtFinType2GLQ      5.163495e-02
## BsmtFinType2LwQ      .
## BsmtFinType2Rec      .
## BsmtFinType2Unf      .
## BsmtFinSF2           1.728392e-05
## BsmtUnfSF            .
## TotalBsmtSF          1.829436e-04
## Heatingother         .
## HeatingQCAvg        -3.037015e-02
## HeatingQCBelowAvg    .
## CentralAirY          6.042200e-02
## ElectricalFuseF     -7.911306e-02
## ElectricalFuseP     -1.452082e-02
## ElectricalSBrkr      .
## X1stFlrSF            .
## X2ndFlrSF            .
## LowQualFinSF        -2.694386e-05
## GrLivArea            5.202221e-04
## BsmtFullBath         3.808607e-02
## BsmtHalfBath         .
## FullBath             .
## HalfBath             .
## BedroomAbvGr         .
## KitchenAbvGr        -5.515930e-02
## KitchenQualAvg      -2.266568e-02
## KitchenQualBelowAvg -2.476052e-02
## TotRmsAbvGrd         .
## FunctionalMaj2      -2.162871e-01
## FunctionalMin1       .
## FunctionalMin2       .
## FunctionalMod        .
## FunctionalTyp        7.792587e-02
## Fireplaces           5.441629e-02
## FireplaceQuAvg       .
## FireplaceQuBelowAvg -5.388664e-03
## GarageTypeAttchd     .
## GarageTypeBasment    .
## GarageTypeBuiltIn    .
## GarageTypeCarPort    .
## GarageTypeDetchd    -2.454855e-03
## GarageYrBlt          .
## GarageFinishRFn      .
## GarageFinishUnf     -2.654040e-02
## GarageCars           6.533527e-02
## GarageArea           1.389243e-04
## GarageQualAvg        .
## GarageQualBelowAvg  -1.275267e-02
## GarageCondAvg        5.904543e-02
## GarageCondBelowAvg  -4.553377e-03
```

```
## PavedDriveP         -1.374579e-02
## PavedDriveY          .
## WoodDeckSF           1.012606e-04
## OpenPorchSF          2.560045e-04
## EncPorchSF           2.890965e-04
## PoolArea             8.491602e-05
## MiscVal              .
## MoSold               .
## YrSold               .
## SaleTypeWD           .
```

CV RMSE LASSO Model

```
## [1] 0.18951
```

(d) ELASTIC NET & LASSO (GLMNET)

These are the two models that we have used to predict the sale prices in the competion, The model with lasso and method glmnet gave us the best results.

Our Lasso Model

This model is the one where we remove outliers that were specified in part (a)