HOMEWORK#5 21
DSA 5103-001
Sujata Sahu
Bhavya Reddy kanuganti

Libraries used:
library(ggplot2)
library(caret)
library(tidyverse)
library(Metrics)
library(AppliedPredictiveModeling)
library(mice)
library(glmnet)
library(pls)
library(earth)
library(car)

Pre-Processing

1.Checking the total number of na values present in each column of the data by using the following command
colSums(is.na(housingData)):

```
> colSums(is.na(housingData))
          Id   MSSubClass     MSZoning  LotFrontage      LotArea        Alley     LotShape
           0            0            0          207            0          938            0
 LandContour    LotConfig    LandSlope Neighborhood   Condition1     BldgType   HouseStyle
           0            0            0            0            0            0            0
 OverallQual  OverallCond    YearBuilt YearRemodAdd    RoofStyle  Exterior1st  Exterior2nd
           0            0            0            0            0            0            0
  MasVnrType   MasVnrArea    ExterQual    ExterCond   Foundation     BsmtQual     BsmtCond
           4            4            0            0            0           31           31
BsmtExposure BsmtFinType1   BsmtFinSF1 BsmtFinType2   BsmtFinSF2    BsmtUnfSF  TotalBsmtSF
          32           31            0           32            0            0            0
     Heating    HeatingQC   CentralAir   Electrical    X1stFlrSF    X2ndFlrSF LowQualFinSF
           0            0            0            1            0            0            0
   GrLivArea BsmtFullBath BsmtHalfBath     FullBath     HalfBath BedroomAbvGr KitchenAbvGr
           0            0            0            0            0            0            0
 KitchenQual TotRmsAbvGrd   Functional   Fireplaces  FireplaceQu   GarageType  GarageYrBlt
           0            0            0            0          466           53           53
GarageFinish   GarageCars   GarageArea   GarageQual   GarageCond   PavedDrive   WoodDeckSF
          53            0            0           53           53            0            0
 OpenPorchSF   EncPorchSF     PoolArea       PoolQC        Fence  MiscFeature      MiscVal
           0            0            0          998          805          966            0
      MoSold       YrSold     SaleType    SalePrice
           0            0            0            0
```

2. Removed the columns with na values more than 50% and the column Id as it does not define any value in data using the following command:
For all the categorical variables related to garrage
hd <- subset(housingData, select = -c(Alley, PoolQC, Fence, MiscFeature,Id, FireplaceQu))

3.Replacing the na values of categorical variables with 0
##For all the categorial variables related to garage

```
hd <- hd %>%
  mutate(GarageType = ifelse(is.na(GarageType), "Not present", GarageType))
hd <- hd %>%
  mutate(GarageFinish = ifelse(is.na(GarageFinish), "Not present", GarageFinish))
hd <- hd %>%
  mutate(GarageQual = ifelse(is.na(GarageQual), "Not present", GarageQual))
hd <- hd %>%
  mutate(GarageCond = ifelse(is.na(GarageCond), "Not present", GarageCond))
hd <- hd %>%
  mutate(GarageYrBlt = ifelse(is.na(GarageYrBlt), YearBuilt, GarageYrBlt))

##For all the categorial variables related to basement
hd<- hd%>% mutate(BsmtQual = ifelse(is.na(BsmtQual), 'none', BsmtQual),
BsmtCond = ifelse(is.na(BsmtCond), 'none', BsmtCond),BsmtExposure =
ifelse(is.na(BsmtExposure), 'No', BsmtExposure),BsmtFinType1 =
ifelse(is.na(BsmtFinType1), 'Unf', BsmtFinType1),BsmtFinType2 =
ifelse(is.na(BsmtFinType2), 'Unf', BsmtFinType2))

#Electrical
summary(factor(hd$Electrical))
hd[is.na(hd$Electrical), ]$Electrical <- "SBrkr"

#MasVnrType
hd <- hd %>%
  mutate(MasVnrType = ifelse(is.na(MasVnrType), "None", MasVnrType))

#MasVnrArea
hd <- hd %>%
  mutate(MasVnrArea = ifelse(is.na(MasVnrArea), 0, MasVnrArea))
```

4.Imputing values for numeric variables

```
pmm_imp <- mice(hd, m = 1, method = "pmm")
hd <- complete(pmm_imp)
```

Again checking the data to see if any na variables sort(colSums(is.na(hd)))

```
> sort(colSums(is.na(hd)))
  MSSubClass     MSZoning  LotFrontage      LotArea     LotShape  LandContour    LotConfig 
           0            0            0            0            0            0            0 
   LandSlope Neighborhood   Condition1     BldgType   HouseStyle  OverallQual  OverallCond 
           0            0            0            0            0            0            0 
   YearBuilt YearRemodAdd    RoofStyle  Exterior1st  Exterior2nd   MasVnrType   MasVnrArea 
           0            0            0            0            0            0            0 
   ExterQual    ExterCond   Foundation     BsmtQual     BsmtCond BsmtExposure BsmtFinType1 
           0            0            0            0            0            0            0 
  BsmtFinSF1 BsmtFinType2   BsmtFinSF2    BsmtUnfSF  TotalBsmtSF      Heating    HeatingQC 
           0            0            0            0            0            0            0 
  CentralAir   Electrical    X1stFlrSF    X2ndFlrSF LowQualFinSF    GrLivArea BsmtFullBath 
           0            0            0            0            0            0            0 
BsmtHalfBath     FullBath     HalfBath BedroomAbvGr KitchenAbvGr  KitchenQual TotRmsAbvGrd 
           0            0            0            0            0            0            0 
  Functional   Fireplaces   GarageType  GarageYrBlt GarageFinish   GarageCars   GarageArea 
           0            0            0            0            0            0            0 
  GarageQual   GarageCond   PavedDrive   WoodDeckSF  OpenPorchSF  EncPorchSF     PoolArea 
           0            0            0            0            0            0            0 
     MiscVal       MoSold       YrSold     SaleType    SalePrice 
           0            0            0            0            0 
```

For this homework the challenge is to find the natural log of sales price
So first we convert the SalePrice column of the data frame to log(SalePrice) using the following command:
hd$SalePrice<-log(hd$SalePrice)

1(a)**OLS Model**

Splitting the data into two sets the first set has 100 observations and the second set as 900 observations using the following commands:
hd100<-head(hd,100)
hd900<-hd[101:1000,]
Ols model calculation
Linearols2 =
lm(SalePrice~OverallQual+GrLivArea+GarageArea+GarageCars+TotalBsmtSF+X1stFlrSF,hd900)
summary(Linearols2)

```
Call:
lm(formula = SalePrice ~ OverallQual + GrLivArea + GarageArea +
    GarageCars + TotalBsmtSF + X1stFlrSF, data = hd900)

Residuals:
     Min       1Q   Median       3Q      Max
-0.66416 -0.07507  0.01099  0.09728  0.52472

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.062e+01  2.474e-02 429.168  < 2e-16 ***
OverallQual  1.077e-01  5.364e-03  20.074  < 2e-16 ***
GrLivArea    2.478e-04  1.354e-05  18.307  < 2e-16 ***
GarageArea   1.696e-04  5.314e-05   3.192  0.00146 **
GarageCars   6.071e-02  1.524e-02   3.984 7.33e-05 ***
TotalBsmtSF  1.965e-04  1.989e-05   9.883  < 2e-16 ***
X1stFlrSF   -6.313e-06  2.362e-05  -0.267  0.78934
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1479 on 893 degrees of freedom
Multiple R-squared:  0.8348,	Adjusted R-squared:  0.8337
F-statistic: 752.3 on 6 and 893 DF,  p-value: < 2.2e-16
```

The above screenshots shows the coefficient estimate,p value,adjusted R squared

##For plotting
plot(Linearols2)
##making prediction on test data
Predict2<-predict(Linearols2,hd100)
Predict2
##Calculating the RMSE
RMSE2<-RMSE(Predict2,hd100$SalePrice)
RMSE2

```
> RMSE2
[1] 0.1512453
```

AIC(Linearols2)

```
> AIC(Linearols2)
[1] -877.5785
```

BIC(Linearols2)

```
> BIC(Linearols2)
[1] -839.1593
```

VIF(Linearols2)

```
> vif(Linearols2)
OverallQual   GrLivArea  GarageArea  GarageCars TotalBsmtSF   X1stFlrSF
   2.043697    1.848561    4.484401    4.754762    2.792432    2.940217
```

Linearols1=lm(SalePrice~.,hd900)

summary(Linearols1)

```
GarageArea               6.333e-05  3.729e-05   1.698 0.089834 .
GarageQualAvg           -3.754e-02  3.856e-02  -0.974 0.330531
GarageQualBelowAvg      -6.467e-02  4.447e-02  -1.454 0.146280
GarageQualNot present           NA         NA      NA       NA
GarageCondAvg            3.941e-02  4.114e-02   0.958 0.338314
GarageCondBelowAvg      -5.669e-03  4.660e-02  -0.122 0.903202
GarageCondNot present           NA         NA      NA       NA
PavedDriveP             -2.470e-02  2.347e-02  -1.052 0.292879
PavedDriveY              4.159e-03  1.519e-02   0.274 0.784298
WoodDeckSF               8.343e-05  2.667e-05   3.129 0.001816 **
OpenPorchSF              1.425e-04  5.318e-05   2.680 0.007514 **
EncPorchSF               2.093e-04  3.994e-05   5.241 2.02e-07 ***
PoolArea                 1.609e-04  1.071e-04   1.503 0.133203
MiscVal                  9.071e-06  1.763e-05   0.514 0.607089
MoSold                  -1.216e-03  1.153e-03  -1.055 0.291836
YrSold                  -1.270e-03  2.370e-03  -0.536 0.591981
SaleTypeWD              -8.668e-03  1.782e-02  -0.486 0.626758
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.08741 on 846 degrees of freedom
Multiple R-squared:  0.9509,    Adjusted R-squared:  0.9421
F-statistic: 107.2 on 153 and 846 DF,  p-value: < 2.2e-16
```

##making prediction on test data

Predict1<-predict(Linearols1,hd100)

##Calculating the RMSE

RMSE1<-RMSE(Predict1,hd100$SalePrice)

RMSE1

```
> RMSE1<-RMSE(Predict1,hd100$SalePrice)
> RMSE1
[1] 0.08075406
```

AIC(Linearols1)

BIC(Linearols1)

vif(Linearols1)

```
> AIC(Linearols1)
[1] -1893.261
> BIC(Linearols1)
[1] -1132.559
> vif(Linearols1)
Error in vif.default(Linearols1) :
  there are aliased coefficients in the model
```

(ii)**Residuals vs Fitted**-when we used the plot function the first plot is Residuals vs Fitted plot.

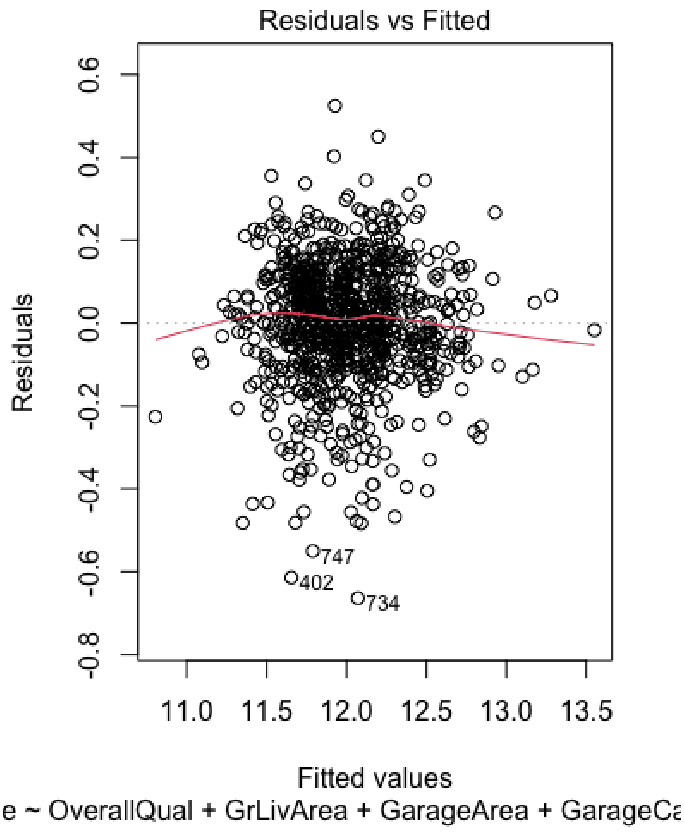
This residual vs fitted values plot shows independent variables in the x axis and residuals on the y axis.The residuals bounce randomly around 0 line and it is not randomly dispersed.Few residuals standout from the basic random pattern so we can say we have few outliers

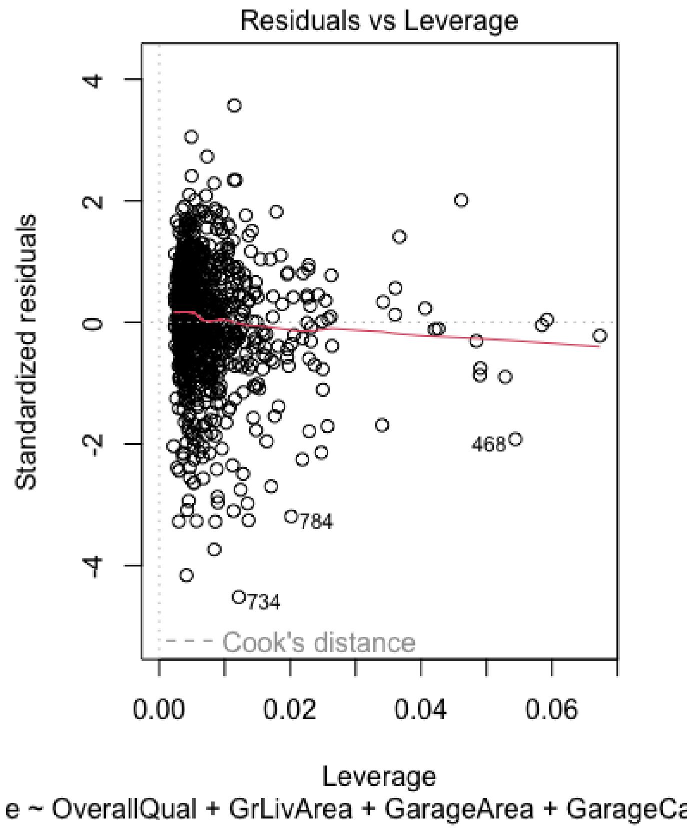
Residuals vs Leverage-
The red line is almost horizontal but the points are not evenly spread. Also we have number of outliers

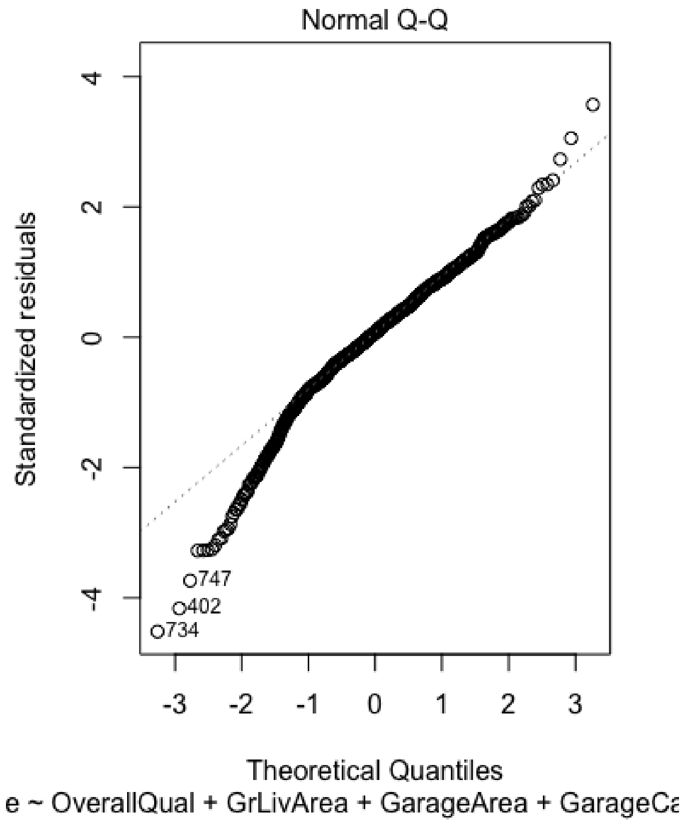
Normal Q-Q-
The normal Q-Qplot shows independent variables in the x axis and residuals on the y axis.The distribution is more widely spread around a central value than the normally distributed data.There are more outliers and the tail of distribution is fatter

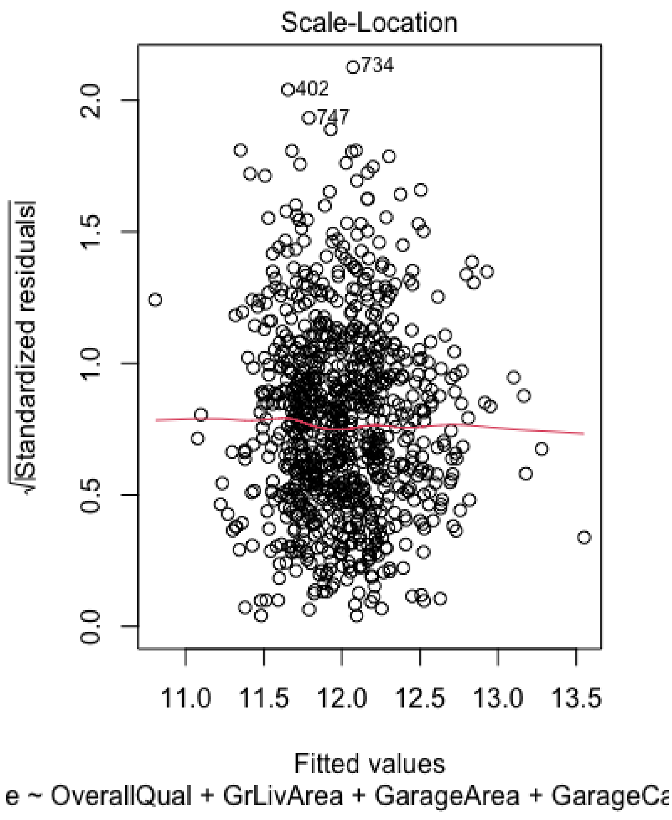
Scale location residual
Looking at the graph we can see the red line is roughly across the horizontal so the average magnitude of the standardised residuals is not changing much as a function of fitted values.Also,there is no clear pattern of the residuals it is randomly scattered around which means the variance are not equal

By checking all the residual pattern i feel none of the plot actually support as the residuals are not evenly spread also only normal Q-A we can see the residuals are not evenly spread across the horizontal line while in others we see there are many outliers and the residuals are closer to each other that means there are correlated to each other also i can't see any kind of pattern in the relation. So i feel i will try some other model where the the residuals fit or try to reduce the components that will fit.


Residuals vs Fitted
Residuals
Fitted values
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
11.0
11.5
12.0
12.5
13.0
13.5
747
402
734
e ~ OverallQual + GrLivArea + GarageArea + GarageCa


Residuals vs Leverage
Standardized residuals
4
2
0
-2
-4
0.00
0.02
0.04
0.06
Leverage
e ~ OverallQual + GrLivArea + GarageArea + GarageCa
468
784
734
Cook's distance


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
747
402
734
-3
-2
-1
0
1
2
3
Theoretical Quantiles
e ~ OverallQual + GrLivArea + GarageArea + GarageCa

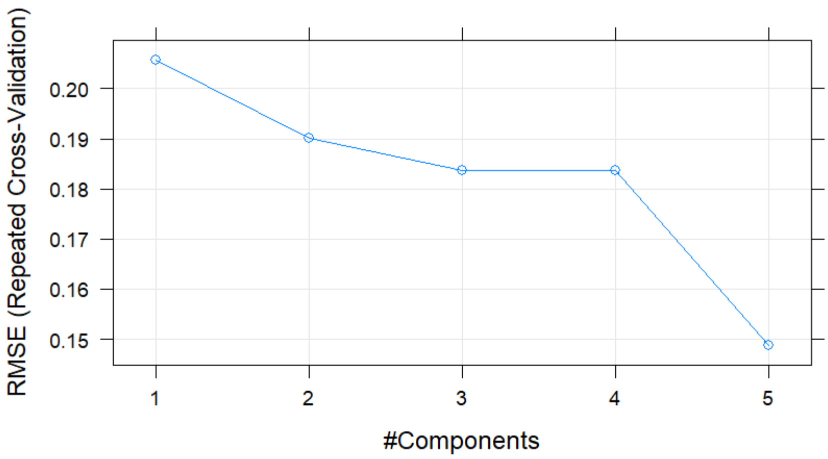

Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
734
402
747
11.0
11.5
12.0
12.5
13.0
13.5
Fitted values
e ~ OverallQual + GrLivArea + GarageArea + GarageCa

(b) **PLS Model**

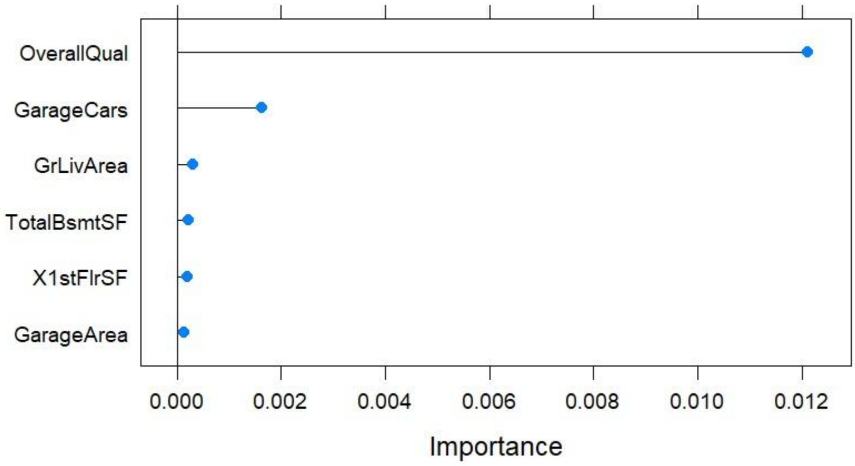
set.seed(123)
custom <- trainControl(method = "repeatedcv", number = 5, repeats = 5, verboseIter = T)
plsmodel<-train(SalePrice~OverallQual+GrLivArea+GarageArea+GarageCars+TotalBsmtSF+X1stFlrSF,data=hd,method="pls",trControl = custom,tuneLength = 10)
plsmodel$results
predictpls <- predict(plsmodel, hd)
RMSE(hd$SalePrice,predictpls)
summary(plsmodel)

```
> RMSE(hd$SalePrice,predictpls)
[1] 0.1481264
> plsmodel$results
  ncomp      RMSE  Rsquared       MAE     RMSESD RsquaredSD       MAESD
1     1 0.2057982 0.6789757 0.1581887 0.01506594 0.06023180 0.010765230
2     2 0.1901122 0.7261350 0.1425299 0.01554353 0.05127389 0.010946536
3     3 0.1838613 0.7428168 0.1370364 0.01431048 0.05069865 0.009914311
4     4 0.1838407 0.7428985 0.1370770 0.01421042 0.05056730 0.009822806
5     5 0.1486611 0.8322497 0.1119536 0.01165874 0.02969828 0.008421704
> summary(plsmodel)
Data:    X dimension: 1000 6
         Y dimension: 1000 1
Fit method: oscorespls
Number of components considered: 5
TRAINING: % variance explained
          1 comps  2 comps  3 comps  4 comps  5 comps
X           65.90    79.93    95.04   100.00   100.00
.outcome    67.87    72.68    74.47    74.48    83.35
```

The number of components is 5 and the CV RMSE is 0.148
plot(plsmodel)



plot(varImp(plsmodel, scale = F))



(c) **LASSO MODEL**

lasso <-
train(SalePrice~OverallQual+GrLivArea+GarageArea+GarageCars+TotalBsmtSF+X
1stFlrSF,
data = hd, method = 'glmnet',
tuneGrid = expand.grid(alpha = 1, lambda = seq(0.0001,1,length=5)),
trControl = custom)
lassopred <- predict(lasso, hd)
RMSE(hd$SalePrice,lassopred)
lasso$results

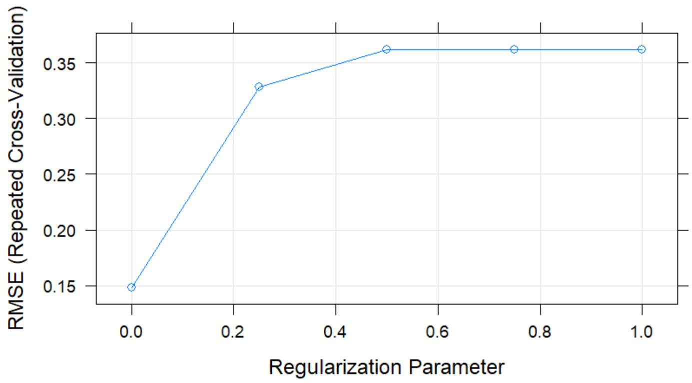
```
> lassopred <- predict(lasso, hd)
> RMSE(hd$SalePrice,lassopred)
[1] 0.1476324
> lasso$results
  alpha   lambda      RMSE  Rsquared       MAE      RMSESD RsquaredSD       MAESD
1     1 0.000100 0.1484545 0.8322688 0.1111145 0.008389624 0.02157999 0.006074676
2     1 0.250075 0.3290268 0.6536674 0.2561230 0.016701493 0.03611337 0.010577344
3     1 0.500050 0.3627555       NaN 0.2839877 0.015215653         NA 0.009179432
4     1 0.750025 0.3627555       NaN 0.2839877 0.015215653         NA 0.009179432
5     1 1.000000 0.3627555       NaN 0.2839877 0.015215653         NA 0.009179432
```

The coefficient values are reported below:
coef(lasso$finalModel)

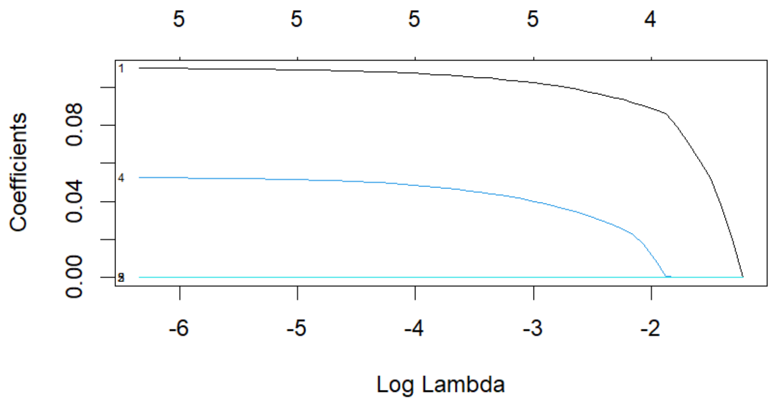
```
(Intercept) 1.102561e+01 1.098802e+01 1.095384e+01 1.092263e+01 1.089419e+01 1.086834e+01
OverallQual 9.640017e-02 9.763621e-02 9.874878e-02 9.977555e-02 1.007117e-01 1.015506e-01
GrLivArea   1.601569e-04 1.674821e-04 1.741434e-04 1.802255e-04 1.857678e-04 1.908051e-04
GarageArea  5.084001e-05 6.311245e-05 7.395947e-05 8.416212e-05 9.347395e-05 1.016184e-04
GarageCars  3.006806e-02 3.208973e-02 3.402279e-02 3.569744e-02 3.721911e-02 3.869810e-02
TotalBsmtSF 8.646728e-05 9.652172e-05 1.057141e-04 1.140602e-04 1.216634e-04 1.286228e-04
X1stFlrSF   .            .            .            .            .            .

(Intercept) 1.084473e+01 1.082321e+01 1.080367e+01 1.078581e+01 1.076952e+01 1.075475e+01
OverallQual 1.023283e-01 1.030376e-01 1.036695e-01 1.042586e-01 1.047964e-01 1.052718e-01
GrLivArea   1.954067e-04 1.996004e-04 2.034098e-04 2.068914e-04 2.100647e-04 2.129464e-04
GarageArea  1.093581e-04 1.164310e-04 1.225317e-04 1.284058e-04 1.337854e-04 1.383428e-04
GarageCars  3.995898e-02 4.110224e-02 4.223763e-02 4.318616e-02 4.404304e-02 4.491784e-02
TotalBsmtSF 1.349342e-04 1.406831e-04 1.459529e-04 1.507256e-04 1.550717e-04 1.590627e-04
X1stFlrSF   .            .            .            .            .            .
```

The RMSE value of final model is 0.14763

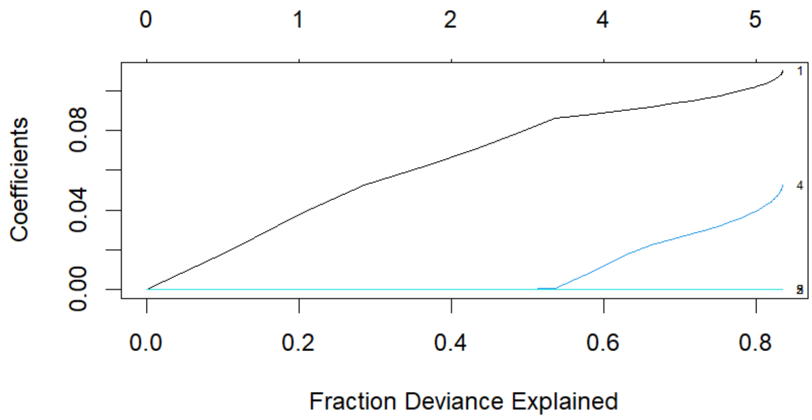
plot(lasso)



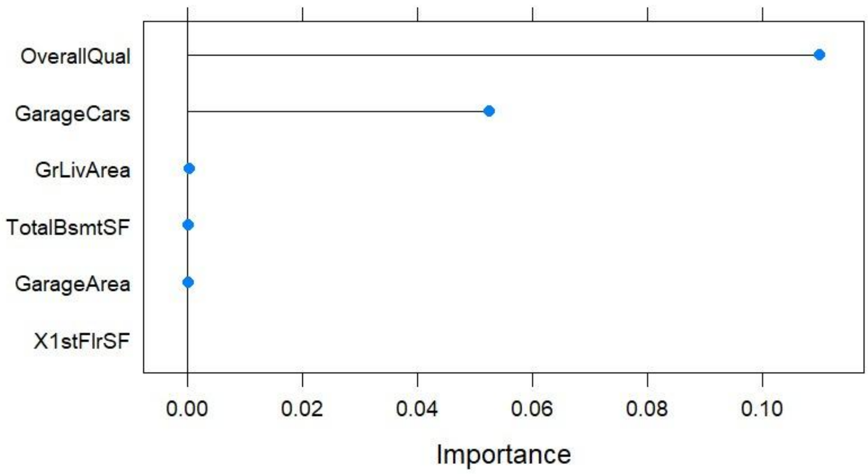
plot(lasso$finalModel, xvar = "lambda", label = T)



plot(lasso$finalModel, xvar = "dev", label = T)



plot(varImp(lassomodel, scale = F))



(d)

Ridge Regression Model

```
set.seed(1234)
ridge <-
train(SalePrice~OverallQual+GrLivArea+GarageArea+GarageCars+TotalBsmtSF+X
1stFlrSF,
            data = hd, method = 'glmnet',
            tuneGrid = expand.grid(alpha = 0, lambda = seq(0.0001,1,length=5)),
trControl = custom)
ridge <-
train(SalePrice~OverallQual+GrLivArea+GarageArea+GarageCars+TotalBsmtSF+X
1stFlrSF,
            data = hd, method = 'glmnet',
            tuneGrid = expand.grid(alpha = 0, lambda = 0.0001), trControl = custom)
ridge$results
```

ridgepred <- predict(ridge, hd)
RMSE(hd$SalePrice,ridgepred)

```
> ridge$results
  alpha lambda      RMSE  Rsquared      MAE      RMSESD RsquaredSD       MAESD
1     0  1e-04 0.1489729 0.8316366 0.111119 0.008611129 0.02268195 0.006284706
> ridgepred <- predict(ridge, hd)
> RMSE(hd$SalePrice,ridgepred)
[1] 0.148305
```

MARS Model

set.seed(1234)
marsFit <-
earth(SalePrice~OverallQual+GrLivArea+GarageArea+GarageCars+TotalBsmtSF+X1stFlrSF,
data = hd,
degree=2,nk=49,pmethod="cv",nfold=5,ncross=5)
plot(marsFit)
predmars<- predict(marsFit, hd)
predmars
RMSEmars<-RMSE(hd$SalePrice,pred)
RMSEmars
marsFit

```
> RMSEmars<-RMSE(hd$SalePrice,pred)
> RMSEmars
[1] 0.1439063
> marsFit
Selected 8 of 31 terms, and 6 of 6 predictors (pmethod="cv")
Termination condition: RSq changed by less than 0.001 at 31 terms
Importance: OverallQual, GrLivArea, TotalBsmtSF, GarageArea, GarageCars, X1stFlrSF
Number of terms at each degree of interaction: 1 5 2
GRSq 0.8315462  RSq 0.8373963  mean.oof.RSq 0.8270965 (sd 0.0215)

pmethod="backward" would have selected:
    16 terms 6 preds,  GRSq 0.8456815  RSq 0.8570495  mean.oof.RSq 0.8131934
```

PCR Model

pcrmodel <-
pcr(SalePrice~OverallQual+GrLivArea+GarageArea+GarageCars+TotalBsmtSF+X1stFlrSF, data = hd900, scale=TRUE,5)
pcr_pred <- predict(model,hd100, ncomp = 4)
summary(model)
RMSEPcr<-RMSE(pcr_pred,hd100$SalePrice)
RMSEPcr

```
> summary(pcrmodel)
Data: 	X dimension: 900 6
	Y dimension: 900 1
Fit method: svdpc
Number of components considered: 5
TRAINING: % variance explained
           1 comps  2 comps  3 comps  4 comps  5 comps
X            58.93    76.03    86.94    94.90    97.99
SalePrice    76.87    76.87    81.54    83.17    83.48
> RMSEPcr<-RMSE(pcr_pred,hd100$SalePrice)
> RMSEPcr
[1] 0.1508738
```

Elastic Net Model

```
set.seed(1234)
en<-
train(SalePrice~OverallQual+GrLivArea+GarageArea+GarageCars+TotalBsmtSF+X
1stFlrSF,
        data = hd, method = 'glmnet',
        tuneGrid = expand.grid(alpha = seq(0,1,length=10), lambda =
seq(0.0001,0.2,length=5)), trControl = custom)

en <-
train(SalePrice~OverallQual+GrLivArea+GarageArea+GarageCars+TotalBsmtSF+X
1stFlrSF,
        data = hd, method = 'glmnet',
        tuneGrid = expand.grid(alpha = 0.8, lambda = 0.0001))

summary(en)
en$results
enpred <- predict(en, hd)
RMSE(hd$SalePrice,enpred)
```

```
> en$results
  alpha lambda      RMSE  Rsquared      MAE      RMSESD RsquaredSD       MAESD
1   0.8  1e-04 0.1484876 0.8327452 0.110782 0.006182281 0.01707976 0.004650063
```

Summary of Model performance with 5 fold cv

Model	Notes	Hyperparameter	CV RMSE	CV R^2
OLS	lm	N/A	0.08075	0.9421
OLS	Lm+2 way	N/A	0.15121	0.8337
PLS	Plsr method and plsr package	ncomp=5	0.1486	0.8322

LASSO	glmnet <-method/package	fraction-0.0001	0.14845	0.8322
Mars	Cv method, earth package	Degree=2	0.1439	0.857
Ridge Regression	glmnet <-method/package	fraction-0.0001	0.1489	0.8316
Elastic net	glmnet <-method/package	fraction-0.0001,0	0.1484	0.8327
pcr	pcr	N/A	0.1508	NULL