Among those 14 potential explanatory variables,Non-dummy variables are:

- Size: 2nd column in the dataset
- Land: 14th column in the dataset
- Bed.Rooms: 5th column in the dataset
- Fireplace: 7th column in the dataset
- Full.Bath: 8th column in the dataset
- Half.Bath: 9th column in the dataset
- Basement..Total.: 10th column in the dataset
- Finished.Bsmt: 11th column in the dataset
- there are 8 non-dummy explanatory variables

1. First step standardize all non-dummy variables via scale function in R.

2. There are 17 columns in the original dataset. A new data frame with 15 columns, which means get rid of two columns, Bsmt.Full.Bath (12th column) and Bsmt.Half.Bath (13th column) in the original dataset

3.

- **OLS output:**

```
> summary(db_outOLS)

Call:
lm(formula = Y ~ X)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0195 -0.2615 -0.0887  0.3133  1.4471

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.67655    0.25962  -2.606   0.0110 *
Size               0.19169    0.10481   1.829   0.0712 .
Garage             0.37643    0.25025   1.504   0.1366
Mutiple.Car        0.05479    0.12755   0.430   0.6687
Bed.Rooms          0.01534    0.06686   0.230   0.8191
Central.Air        0.25323    0.13507   1.875   0.0646 .
Fireplace          0.29755    0.06701   4.440 2.92e-05 ***
Full.Bath          0.13949    0.07783   1.792   0.0770 .
Half.Bath          0.17423    0.06624   2.630   0.0103 *
Basement..Total.   0.19056    0.06580   2.896   0.0049 **
Finished.Bsmt     -0.08167    0.06939  -1.177   0.2427
Land               0.24628    0.05947   4.141 8.67e-05 ***
Style..2.Story.    0.18122    0.13460   1.346   0.1821
Zone..Town.Center. -0.07348    0.12016  -0.611   0.5427
Bsmt.Bath          0.44730    0.17379   2.574   0.0120 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4964 on 78 degrees of freedom
Multiple R-squared: 0.7911,    Adjusted R-squared: 0.7536
F-statistic:  21.1 on 14 and 78 DF,  p-value: < 2.2e-16
```

From OLS output, the highlight predictor variables have significant coefficient.

- **Lasso**

```
> coef(db_outlasso, s = "lambda.1se")
15 x 1 sparse Matrix of class "dgCMatrix"
                          1
(Intercept)      -0.02245842
Size              0.36893049
Garage            .
Mutiple.Car       .
Bed.Rooms         .
Central.Air       0.03071519
Fireplace         0.24988475
Full.Bath         .
Half.Bath         .
Basement..Total.  .
Finished.Bsmt     .
Land              0.08746876
Style..2.Story.   .
Zone..Town.Center.  .
Bsmt.Bath         .
```

There are four predictor variables via LASSO procedure: size, central air, fire place, land

- **Ridge**

```
> coef(db_outridge, s = "lambda.1se")
15 x 1 sparse Matrix of class "dgCMatrix"
                          1
(Intercept)      -0.33450708
Size              0.12442424
Garage            0.15329889
Mutiple.Car       0.12332425
Bed.Rooms         0.04848359
Central.Air       0.16260614
Fireplace         0.12142357
Full.Bath         0.07364743
Half.Bath         0.06218625
Basement..Total.  0.07689481
Finished.Bsmt     0.02855760
Land              0.10100580
Style..2.Story.   0.10333037
Zone..Town.Center. -0.08929170
Bsmt.Bath         0.13362168
```

The first four predictors with largest coefficients are: central air, size, garage, basement bath

- **Elastic Net with \alpha = 0.5**

```
> coef(db_outEnet, s = "lambda.1se")
15 x 1 sparse Matrix of class "dgCMatrix"
                          1
(Intercept)      -0.0006082502
Size              0.2673322330
Garage            .
Mutiple.Car       .
Bed.Rooms         .
Central.Air       0.0008318716
Fireplace         0.1798513807
Full.Bath         .
Half.Bath         .
Basement..Total.  .
Finished.Bsmt     .
Land              0.0565254238
Style..2.Story.   .
```

```
Zone..Town.Center.   .
Bsmt.Bath            .
```

There are four predictor variables via LASSO procedure: size, central air, fire place, land

Ps: OLS will give more relatively complexity model.

The four predictors that be considered in the following analysis are: **size, central air, fire place, Bsmt.Bath**

The highest correlation are between lasso and Elastic Net.

Here is the R code:

```
rm(list = ls())

## read data

db = read.csv("F:././././ISU_course/stat342/hw8/328 Final 03 Data.csv",stringsAsFactors = FALSE)

## print out the columns names of the data set

print(colnames(db))


## Among those 14 potential explanatory variables,Non-dummy variables are:

## Size: 2nd column in the dataset

## Land: 14th column in the dataset

## Bed.Rooms: 5th column in the dataset

## Fireplace: 7th column in the dataset

## Full.Bath: 8th column in the dataset

## Half.Bath: 9th column in the dataset

## Basement..Total.: 10th column in the dataset

## Finished.Bsmt: 11th column in the dataset

## there are 8 non-dummy explanatory variables


##  standardize all those eight explanatory variables using scale function in R

db_standardized = db

db_standardized$Size = scale(db$Size)

db_standardized$Land = scale(db$Land)

db_standardized$Bed.Rooms = scale(db$Bed.Rooms)

db_standardized$Fireplace = scale(db$Fireplace)

db_standardized$Full.Bath = scale(db$Full.Bath)

db_standardized$Half.Bath = scale(db$Half.Bath)

db_standardized$Basement..Total. = scale(db$Basement..Total.)

db_standardized$Finished.Bsmt = scale(db$Finished.Bsmt)

db_standardized$Price = scale(db$Price)
```

```
## there are 17 columns in the original dataset

## a new data frame with 15 columns, which means get rid of two columns, Bsmt.Full.Bath (12th column) and Bsmt.Half.Bath (13th column) in the original dataset

db_standardized_15 = as.data.frame(db_standardized[,c(-12,-13)])


Y = db_standardized_15$Price

X = db_standardized_15[,c(-1)]

X = as.matrix(X)


## OLS

db_outOLS  = lm(Y~X)

summary(db_outOLS)

coef(db_outOLS)


# lasso fit

library(glmnet)

db_outlasso = cv.glmnet(X,Y,alpha=1)

summary(db_outlasso)

x11()

plot(db_outlasso)

## get the coefficeinet for lasso regression

coef(db_outlasso, s = "lambda.1se")


#Next a ridge fit


db_outridge = cv.glmnet(X,Y,alpha=0)

summary(db_outridge)

x11()

plot(db_outridge)
```

```
## get the coefficeinet for ridge fit regression

coef(db_outridge, s = "lambda.1se")



#Then an alpha=.5 elastic net fit


db_outEnet = cv.glmnet(X,Y,alpha=.5)

summary(db_outEnet)

x11()

plot(db_outEnet)


## get the coefficeinet for elastic net fit

coef(db_outEnet, s = "lambda.1se")


library(FNN)

## choose k with largest R2-Predict value.

## ps: P2-Predict here represents predicted R-square. So larger is better

## only chose size, central air, fire place, garage four variables

#x_new = X[,c("Size", "Central.Air", "Fireplace", "Garage")]

x_new = X[,c("Size", "Central.Air", "Fireplace", "Bsmt.Bath")]

db_outknn = knn.reg(x_new,test=NULL,Y,k=4)

print(db_outknn)


#db_outknn$pred


#Now try random forest fitting

## use default value

library(randomForest)

db_outrf = randomForest(Y~x_new[,1]+x_new[,2]+x_new[,3]+x_new[,4],type="regression")
```

```
##predict(db_outrf)



## N-W kernel regression

library(regpro)


nwRegr = function(x){

 ## input each point, apply N-W kernel regression

 ## return predicted value for that point

 y_pred = kernesti.regr(x,x_new,Y,h = 3)

 return(y_pred)

}


db_outnw = rep(0,nrow(x_new))

for(i in 1:nrow(X)){

 db_outnw = nwRegr(x_new[i,])

}


#Now some local regression smoothing

db_outloess1 =
loess(Y~x_new,as.data.frame(cbind(x_new,Y)),control=loess.control("direct"),family="gaussian",degree=2,span=
0.75,normalize=FALSE)


#Now cross-validate local regression smmothing


library(bisoreg)


db_loesscv = loess.wrapper(x_new, Y, span.vals = seq(.25, 1, by = 0.5), folds = 5)
```

```
#Now some neural nets and averages of neural nets

require(nnet)

db_outnnet = nnet(X,Y,size=1,decay=.001,linout=TRUE,maxit=1000)

var(residuals(db_outnnet))

fitted.values(db_outnnet)


require(caret)

db_outavnnet = avNNet(X,Y,repeats=8,size=1,decay=.001,linout=TRUE,maxit=1000)


Y_OLS = predict(db_outOLS,as.data.frame(X))

Y_ridge = predict(db_outridge,X,s="lambda.1se")

Y_lasso = predict(db_outlasso,X,s="lambda.1se")

Y_Enet = predict(db_outEnet,X,s="lambda.1se")

Y_knn = db_outknn$pred

Y_loess1 = predict(db_outloess1)

Y_loesscv = predict(db_loesscv)

Y_rf = predict(db_outrf)

Y_nw = db_outnw

Y_nnet = fitted.values(db_outnnet)

Y_avnnet = predict(db_outavnnet)


Ypreds = as.data.frame(cbind(Y_OLS,Y_ridge,Y_lasso,Y_Enet,Y_knn,Y_loess1,Y_loesscv,Y_rf,Y_nnet,Y_avnnet))

colnames(Ypreds)<-c("Y_OLS","Y_ridge","Y_lasso","Y_enet","Y_knn","Y_loess1","Y_loesscv","Y_rf","Y_nnet","Y_avnnet")

x11()

pairs(Ypreds,pch = 20)

cor(Ypreds)
```