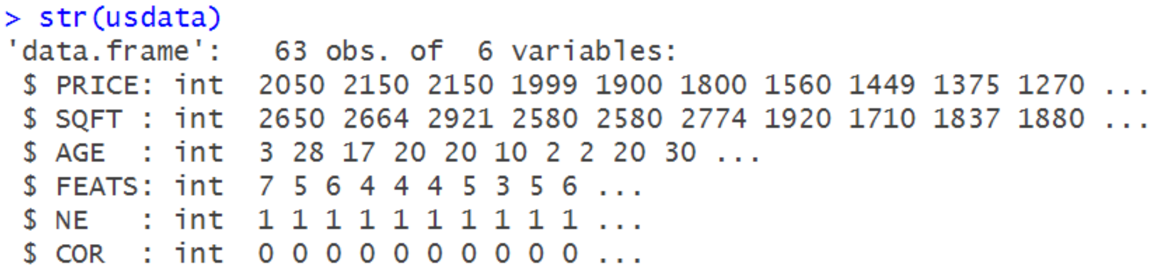
Assignment 2

FOR LAB ON CONSTRUCTING MODELS

Panagiotis Vaidomarkakis | Statistics for Business Analytics I (Part Time) | 22.12.2022

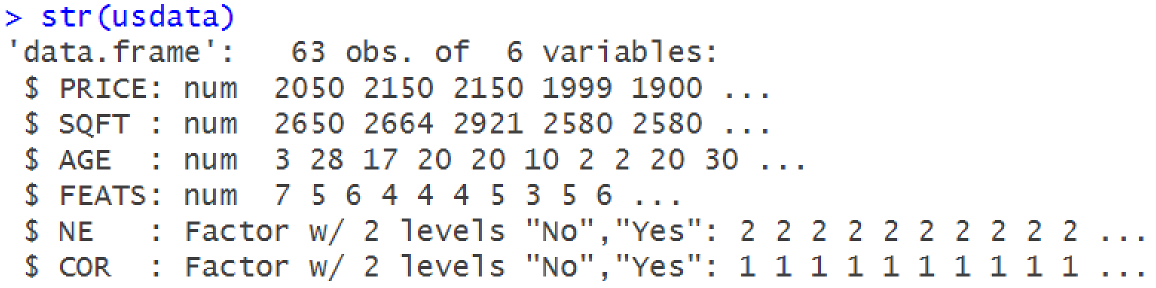
Question 1: Read the “usdata” dataset and use str() to understand its structure.



```
> str(usdata)
'data.frame':	63 obs. of  6 variables:
 $ PRICE: int  2050 2150 2150 1999 1900 1800 1560 1449 1375 1270 ...
 $ SQFT : int  2650 2664 2921 2580 2580 2774 1920 1710 1837 1880 ...
 $ AGE  : int  3 28 17 20 20 10 2 2 20 30 ...
 $ FEATS: int  7 5 6 4 4 4 5 3 5 6 ...
 $ NE   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ COR  : int  0 0 0 0 0 0 0 0 0 0 ...
```

First variables seem to be set correctly as they are numeric. NE & Cor are set as numeric, but do not seem to be so, since they have factorial meaning.

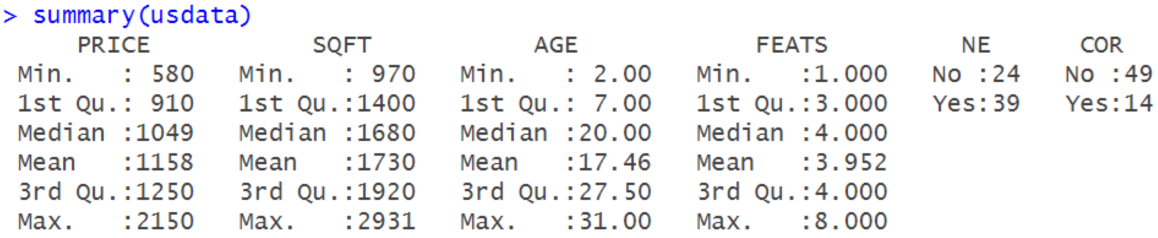
Question 2: Convert the variables PRICE, SQFT, AGE, FEATS to be numeric variables and NE, COR to be factors.



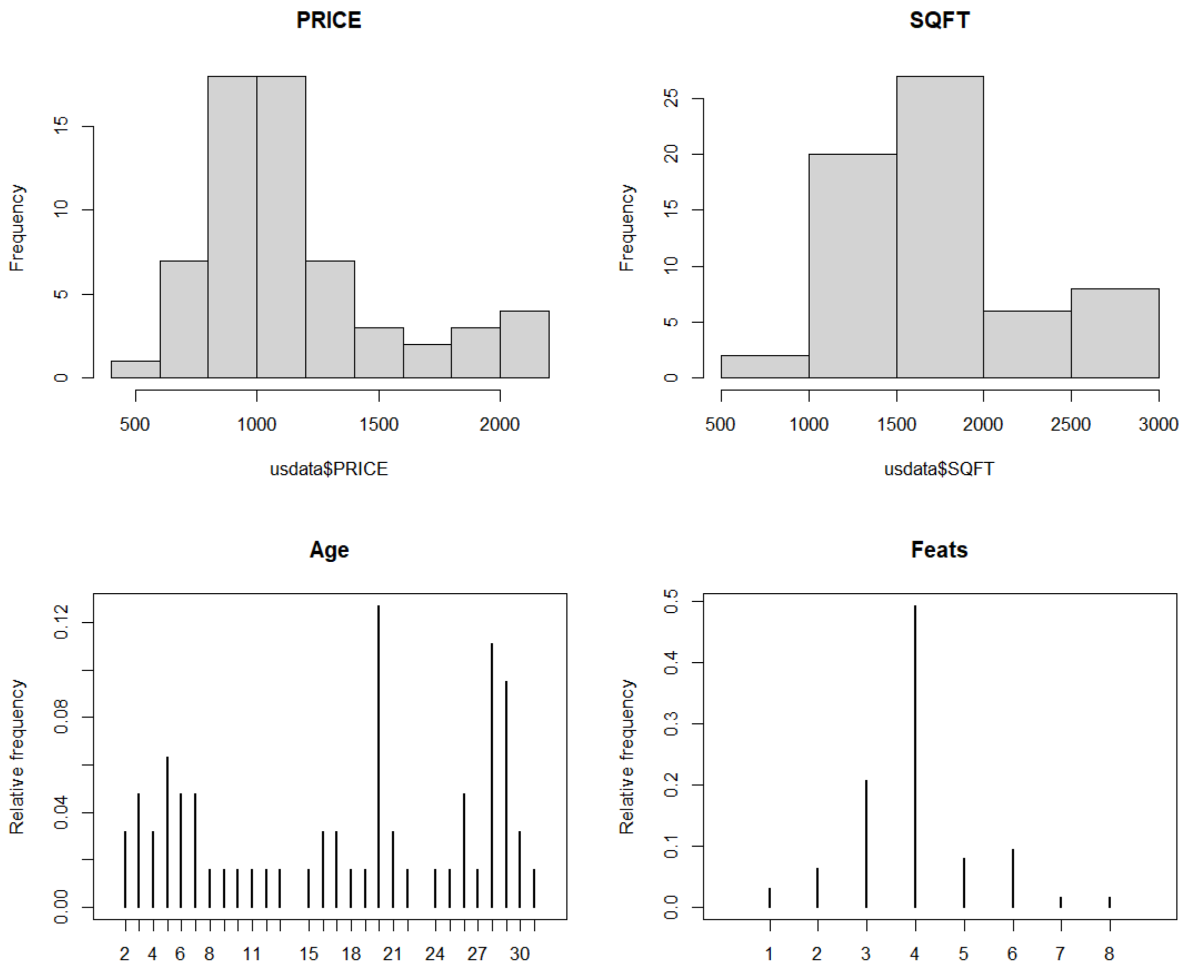
```
> str(usdata)
'data.frame':	63 obs. of  6 variables:
 $ PRICE: num  2050 2150 2150 1999 1900 ...
 $ SQFT : num  2650 2664 2921 2580 2580 ...
 $ AGE  : num  3 28 17 20 20 10 2 2 20 30 ...
 $ FEATS: num  7 5 6 4 4 4 5 3 5 6 ...
 $ NE   : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ COR  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

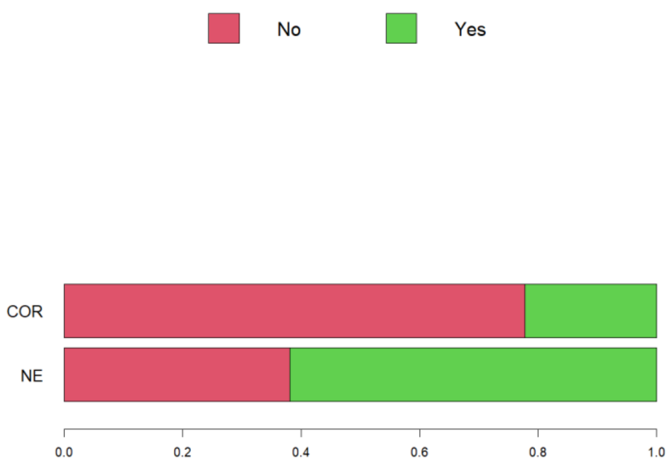
As you can see, I have made the necessary changes in order to have the correct type of variables.

Question 3: Perform descriptive analysis and visualization for each variable to get an initial insight of what the data looks like. Comment on your findings.



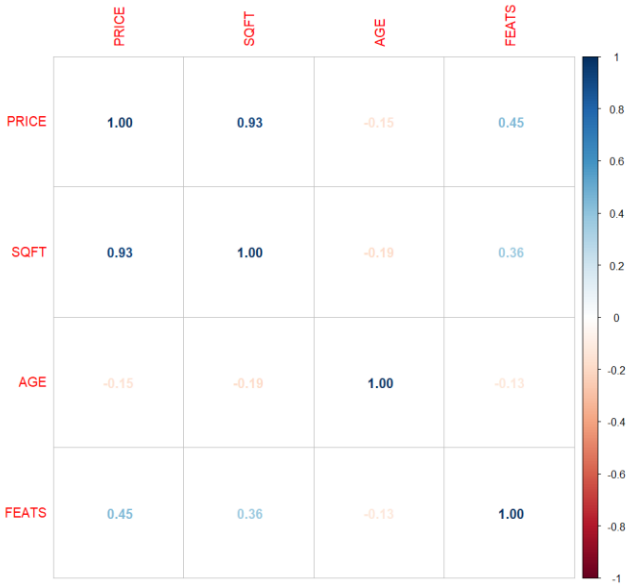
```
> summary(usdata)
     PRICE           SQFT           AGE             FEATS         NE        COR    
 Min.   : 580   Min.   : 970   Min.   : 2.00   Min.   :1.000   No :24   No :49  
 1st Qu.: 910   1st Qu.:1400   1st Qu.: 7.00   1st Qu.:3.000   Yes:39   Yes:14  
 Median :1049   Median :1680   Median :20.00   Median :4.000                    
 Mean   :1158   Mean   :1730   Mean   :17.46   Mean   :3.952                    
 3rd Qu.:1250   3rd Qu.:1920   3rd Qu.:27.50   3rd Qu.:4.000                    
 Max.   :2150   Max.   :2931   Max.   :31.00   Max.   :8.000                    
```


PRICE
Frequency
0
5
10
15
500
1000
1500
2000
usdata$PRICE
SQFT
Frequency
0
5
10
15
20
25
500
1000
1500
2000
2500
3000
usdata$SQFT
Age
Relative frequency
0.00
0.04
0.08
0.12
2
4
6
8
11
15
18
21
24
27
30
Feats
Relative frequency
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5
6
7
8

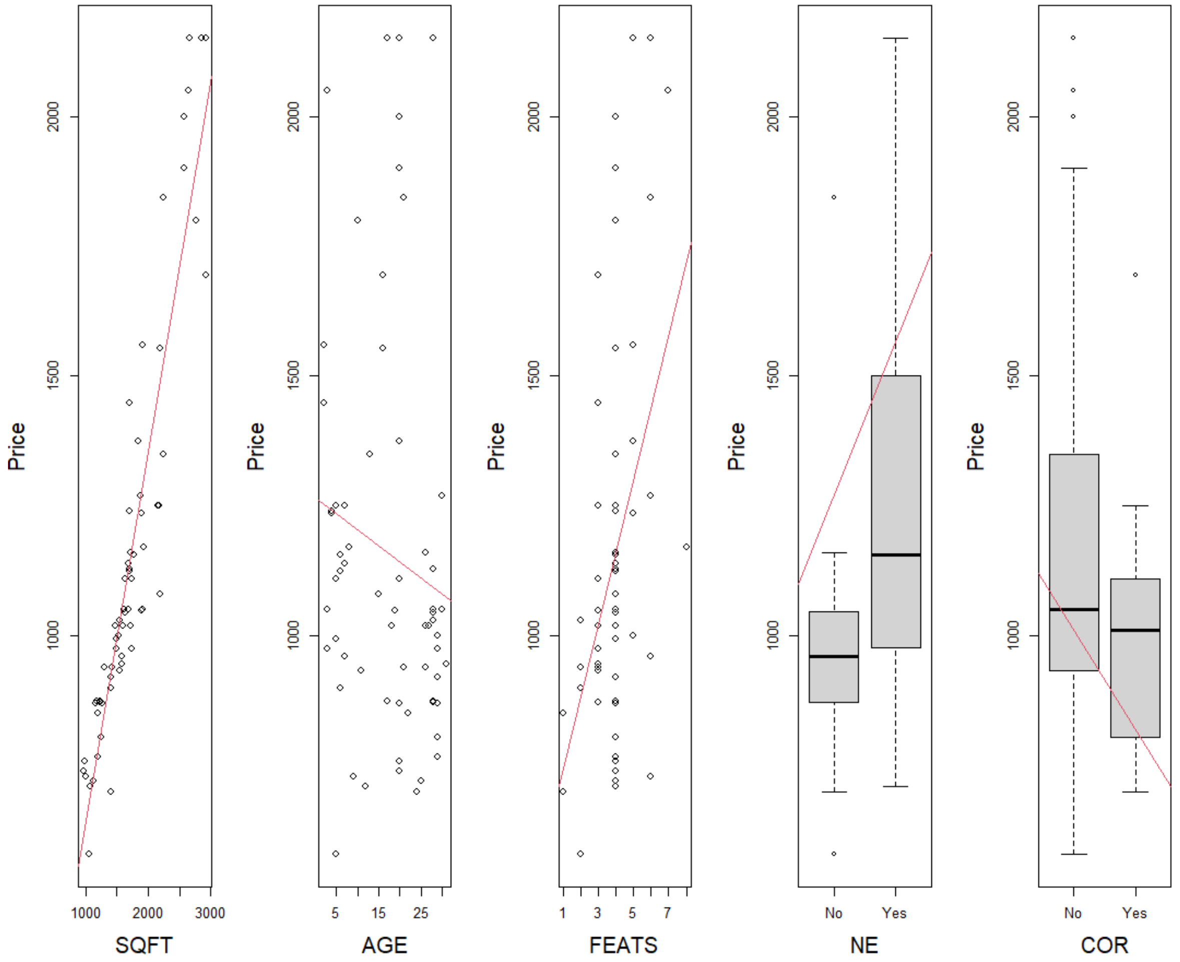

🟥 No
🟩 Yes
COR
NE
0.0
0.2
0.4
0.6
0.8
1.0

So, the first view helps us understand we are dealing with mostly "middle class" houses that are not too big, have mostly 3 or 4 main features covered and are relatively older. That would explain possibly the price being mostly around 1k. Also looks like most houses are in-between others rather than having a more "open" view and are located NE.

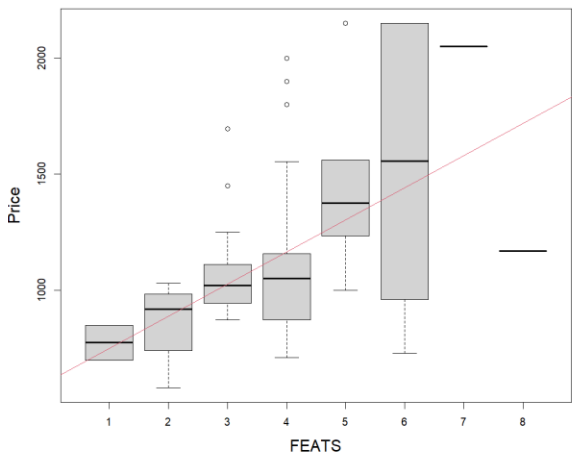
Question 4: Conduct pairwise comparisons between the variables in the dataset to investigate if there are any associations implied by the dataset. Comment on your findings. Is there a linear relationship between PRICE and any of the variables in the dataset?



Correlation matrix gives us a very helpful insight on some probable association between PRICE and SQFT. It could be something expected as bigger house usually means more expensive. There also seems to be some correlation between PRICE and house FEATS, and AGE playing small to no part in PRICE (as expected AGE has nothing to do with FEATS or SQFT of a house). We can probably assume that PRICE and SQFT would be significant variables we should take in consideration when applying a model further down.

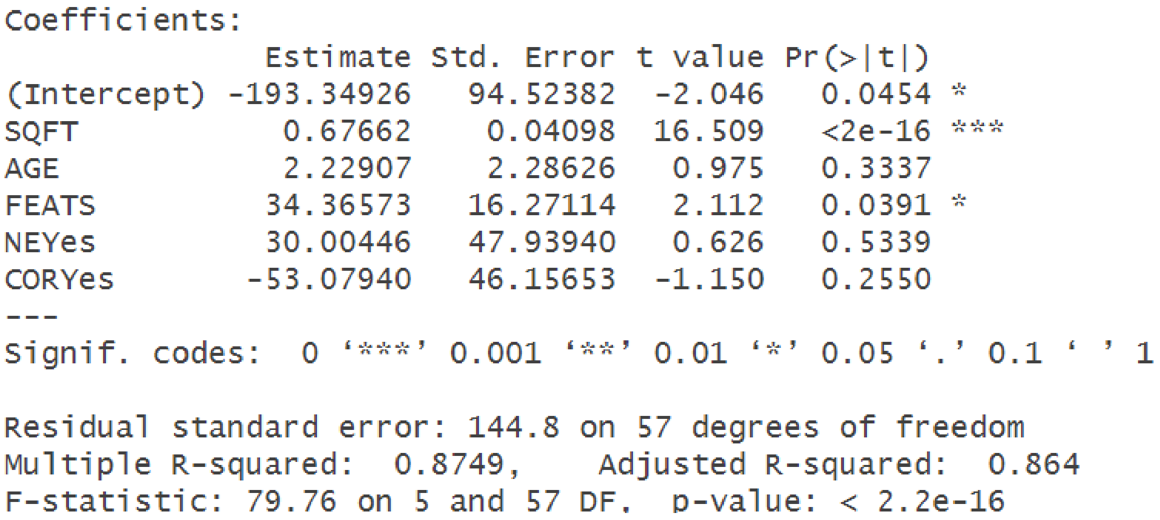


Additionally, we can see here that the median of Yes on NE seems to be higher than the 3rd quantile of No, so it could be a relation between price and location of the house.



From this plot, it's easier to see there could be a trend in PRICE and FEATS as well.

Question 5: Construct a model for the expected selling prices (PRICE) according to the remaining features. Does this linear model fit well to the data?



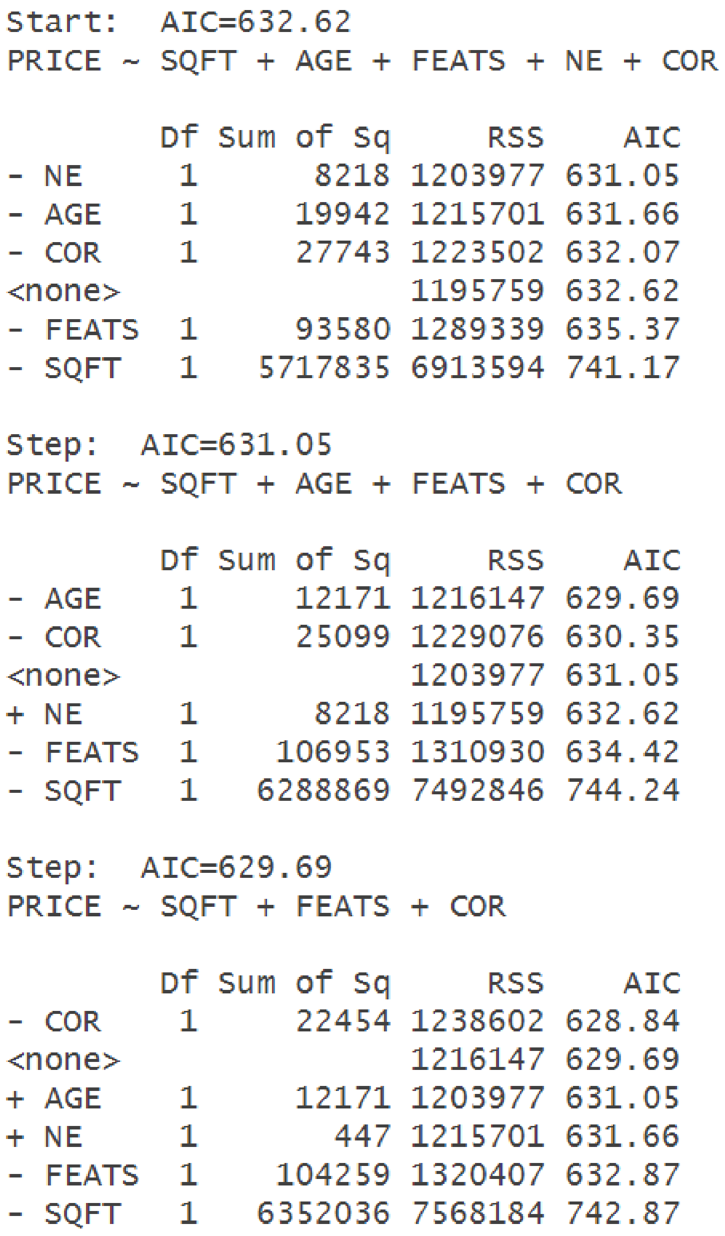
```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -193.34926   94.52382  -2.046   0.0454 *  
SQFT           0.67662    0.04098  16.509   <2e-16 ***
AGE            2.22907    2.28626   0.975   0.3337    
FEATS         34.36573   16.27114   2.112   0.0391 *  
NEYes         30.00446   47.93940   0.626   0.5339    
CORYes       -53.07940   46.15653  -1.150   0.2550    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 144.8 on 57 degrees of freedom
Multiple R-squared:  0.8749,	Adjusted R-squared:  0.864 
F-statistic: 79.76 on 5 and 57 DF,  p-value: < 2.2e-16
```

Looks like the most important variants are SQFT, which lines with our original expectations from the graphical interpretation as a significant variable. We also notice an R^2 adjusted of 86,4% as a goodness of fit measure that the variance of price is explained well from these selected attributes. Our Intercept is our base value of getting that house, so the expected amount to be paid and our price would be modeled as:

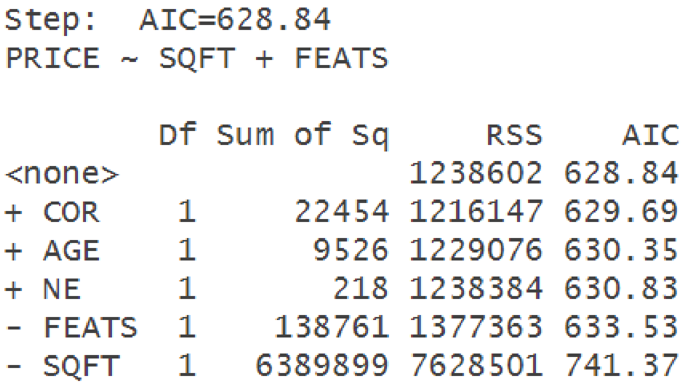
Price = -193,35 + SQFT * 0,68 + AGE * 2,2 + FEAT * 34,37 + 30 if in NE – 53,08 if in location COR + e

where e = N(0, 144,8). e comes from the residual standard error (144,8) output.

Question 6 and 7: Find the best model for predicting the selling prices (PRICE). Select the appropriate features using stepwise methods. Get the summary of your final model and comment on the output. Interpret the coefficients. Comment on the significance of each coefficient and write down the mathematical formulation of the model. Should the intercept be excluded from our model?

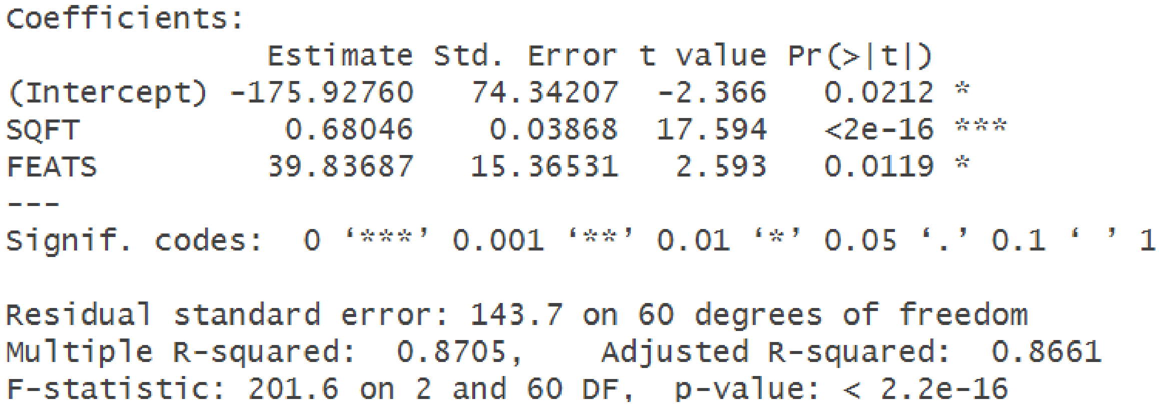


```
Start:  AIC=632.62
PRICE ~ SQFT + AGE + FEATS + NE + COR

        Df Sum of Sq     RSS    AIC
- NE     1      8218 1203977 631.05
- AGE    1     19942 1215701 631.66
- COR    1     27743 1223502 632.07
<none>               1195759 632.62
- FEATS  1     93580 1289339 635.37
- SQFT   1   5717835 6913594 741.17

Step:  AIC=631.05
PRICE ~ SQFT + AGE + FEATS + COR

        Df Sum of Sq     RSS    AIC
- AGE    1     12171 1216147 629.69
- COR    1     25099 1229076 630.35
<none>               1203977 631.05
+ NE     1      8218 1195759 632.62
- FEATS  1    106953 1310930 634.42
- SQFT   1   6288869 7492846 744.24

Step:  AIC=629.69
PRICE ~ SQFT + FEATS + COR

        Df Sum of Sq     RSS    AIC
- COR    1     22454 1238602 628.84
<none>               1216147 629.69
+ AGE    1     12171 1203977 631.05
+ NE     1       447 1215701 631.66
- FEATS  1    104259 1320407 632.87
- SQFT   1   6352036 7568184 742.87
```



```
Step:  AIC=628.84
PRICE ~ SQFT + FEATS

        Df Sum of Sq     RSS    AIC
<none>                1238602 628.84
+ COR    1     22454 1216147 629.69
+ AGE    1      9526 1229076 630.35
+ NE     1       218 1238384 630.83
- FEATS  1    138761 1377363 633.53
- SQFT   1   6389899 7628501 741.37
```

After performing the stepwise method using 'both' direction, meaning that in each step, it can add or subtract an attribute based on AIC. Each time, it calculates the AIC and if it is smaller, then it chooses to proceed to a new model, with or without the attribute that has the smallest AIC.



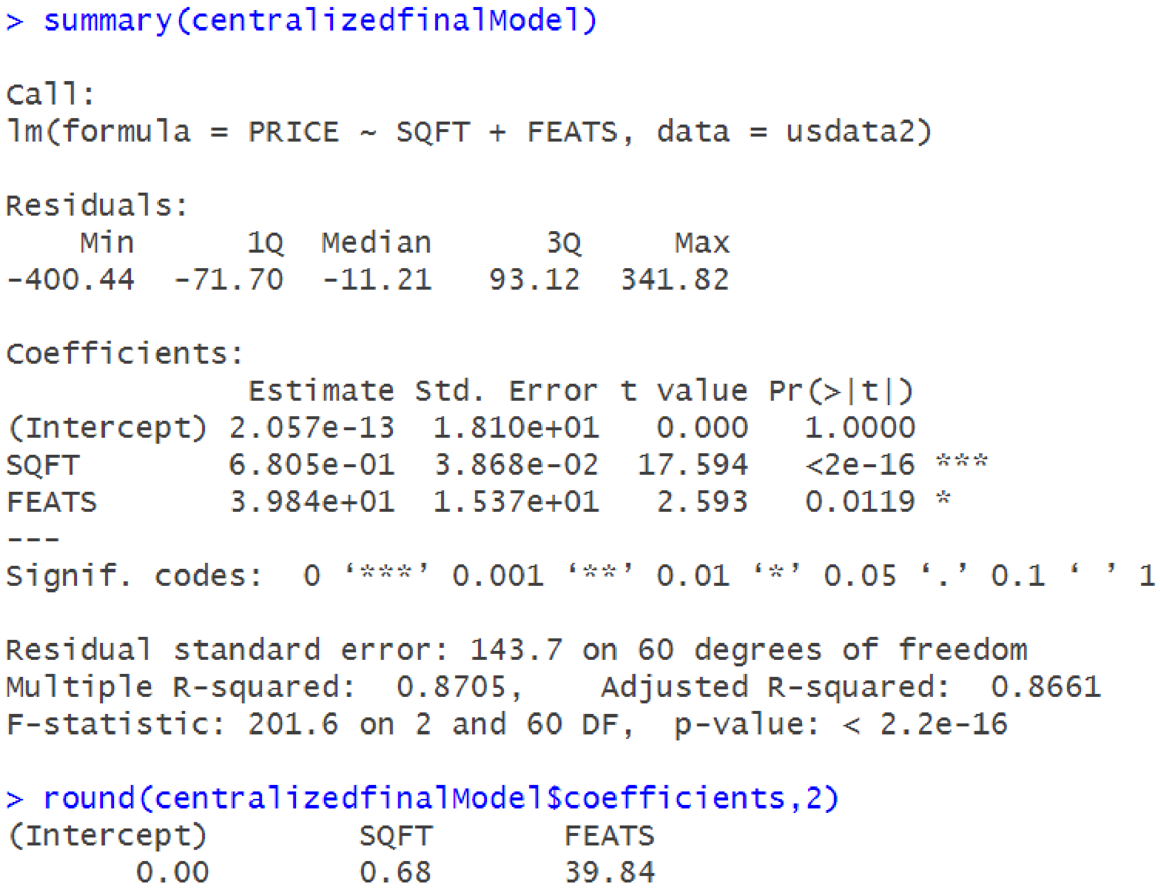
```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -175.92760   74.34207  -2.366   0.0212 *  
SQFT           0.68046    0.03868  17.594   <2e-16 ***
FEATS         39.83687   15.36531   2.593   0.0119 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 143.7 on 60 degrees of freedom
Multiple R-squared:  0.8705,    Adjusted R-squared:  0.8661 
F-statistic: 201.6 on 2 and 60 DF,  p-value: < 2.2e-16
```

Stepwise AIC gives us our final model (interpretation similar to above) and gives us the estimated pricing to be:

Price = -175,93 + SQFT * 0,68 + FEAT * 39,84 + e

Our estimated PRICE is a measure of how big the house is and how many features it has (which makes total sense), where e = N(0, 143,7). e comes from the residual standard error (143,7) output. Our R^2 adjusted is 86,6% which, even if it's not to be relying solely on to compare two models, we get an idea we are in the right direction.

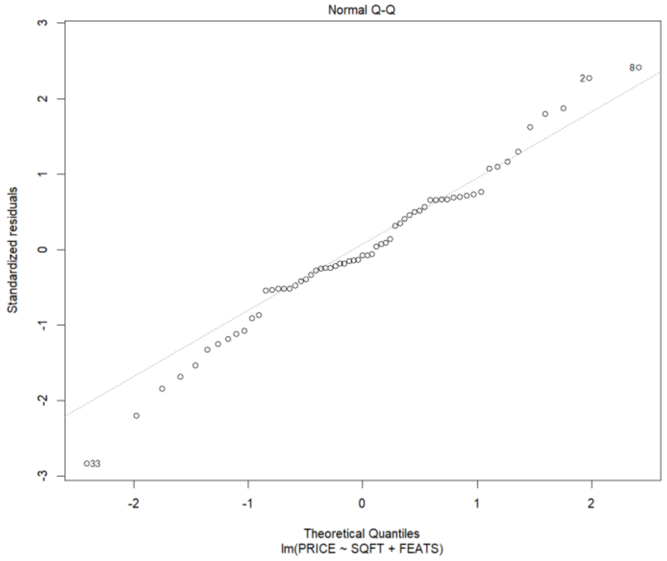
The intercept doesn't really make any sense because if you buy a house with 0 SQFT and 0 FEATS, you cannot pay -17.593 PRICE currency, but we cannot exclude it, because otherwise, the model will not fit in our data. We can center the covariance in order to have a model which the intercept makes sense as we do below.

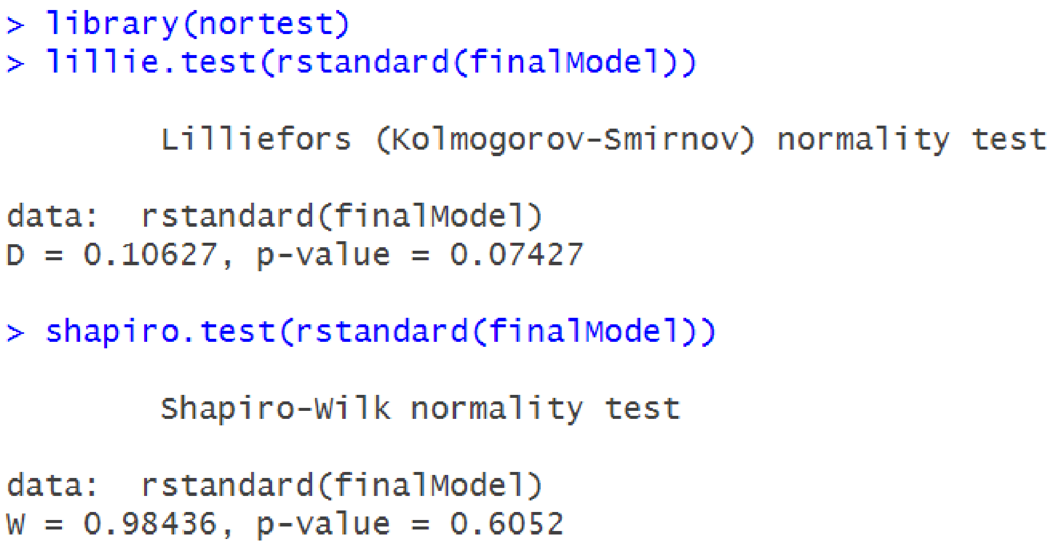


```
> summary(centralizedfinalModel)

Call:
lm(formula = PRICE ~ SQFT + FEATS, data = usdata2)

Residuals:
    Min      1Q  Median      3Q     Max 
-400.44  -71.70  -11.21   93.12  341.82 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.057e-13  1.810e+01   0.000   1.0000    
SQFT        6.805e-01  3.868e-02  17.594   <2e-16 ***
FEATS       3.984e+01  1.537e+01   2.593   0.0119 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 143.7 on 60 degrees of freedom
Multiple R-squared:  0.8705,	Adjusted R-squared:  0.8661 
F-statistic: 201.6 on 2 and 60 DF,  p-value: < 2.2e-16

> round(centralizedfinalModel$coefficients,2)
(Intercept)        SQFT       FEATS 
       0.00        0.68       39.84 
```

As you can see, now the intercept is almost 0 and that makes sense. If you are going to buy a house with 0 SQFT and 0 FEATS, you will need 0 money to buy it.

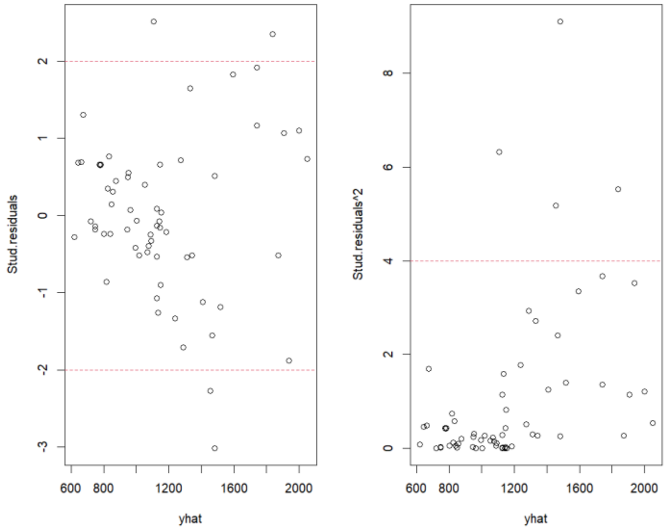
Question 8: Check the assumptions of your final model. Are the assumptions satisfied? If not, what is the impact of the violation of the assumption not satisfied in terms of inference? What could someone do about it?

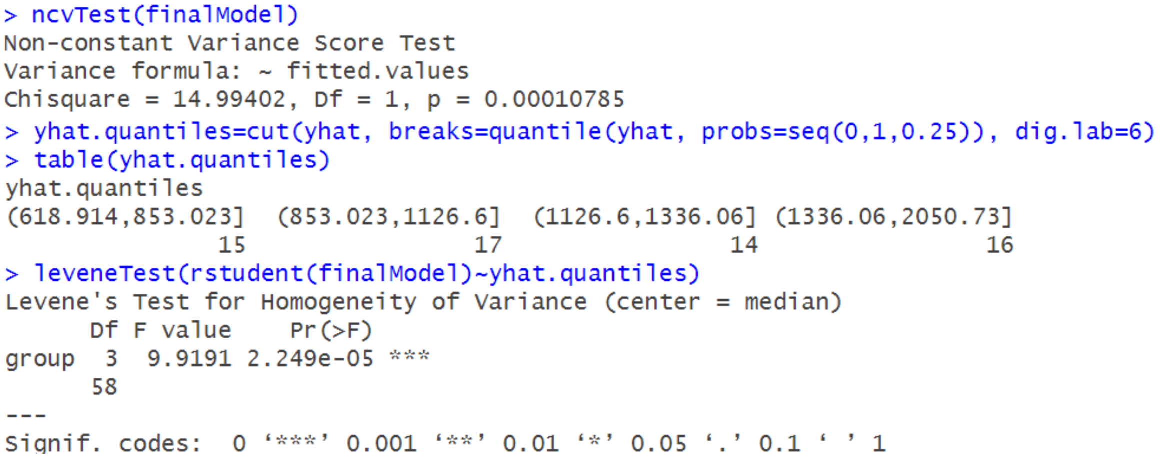




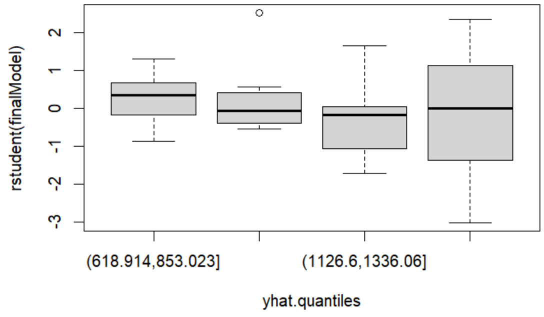
```
> library(nortest)
> lillie.test(rstandard(finalModel))

        Lilliefors (Kolmogorov-Smirnov) normality test

data:  rstandard(finalModel)
D = 0.10627, p-value = 0.07427

> shapiro.test(rstandard(finalModel))

        Shapiro-Wilk normality test

data:  rstandard(finalModel)
W = 0.98436, p-value = 0.6052
```

Running Kolmogorov-Smirnov and Shapiro-Wilk normality test, we check the normality of our final model, where we get a value >5%, not rejecting the Null hypothesis of linearity.

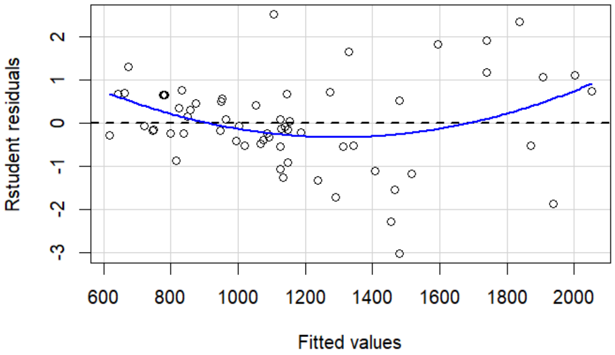




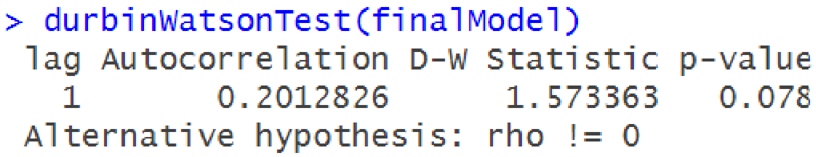
```
> ncvTest(finalModel)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 14.99402, Df = 1, p = 0.00010785
> yhat.quantiles=cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)
> table(yhat.quantiles)
yhat.quantiles
(618.914,853.023]   (853.023,1126.6]   (1126.6,1336.06] (1336.06,2050.73] 
               15                 17                 14                16 
> leveneTest(rstudent(finalModel)~yhat.quantiles)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value    Pr(>F)    
group  3  9.9191 2.249e-05 ***
      58                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



As we can see from the plots, we don't have homoscedasticity because we have errors outside the barriers. Furthermore, both tests reject the H_0 hypothesis, which makes our belief even stronger. In the end, there is a boxplot in order to see that they aren't in the same size and in the same position in order to have homoscedasticity.

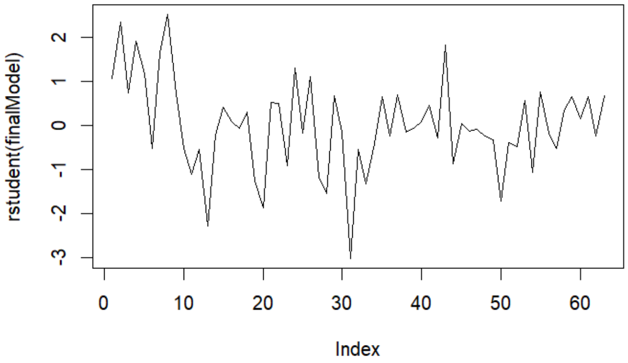


As we can see, there is no linearity. It seems like x^2 polynomial.



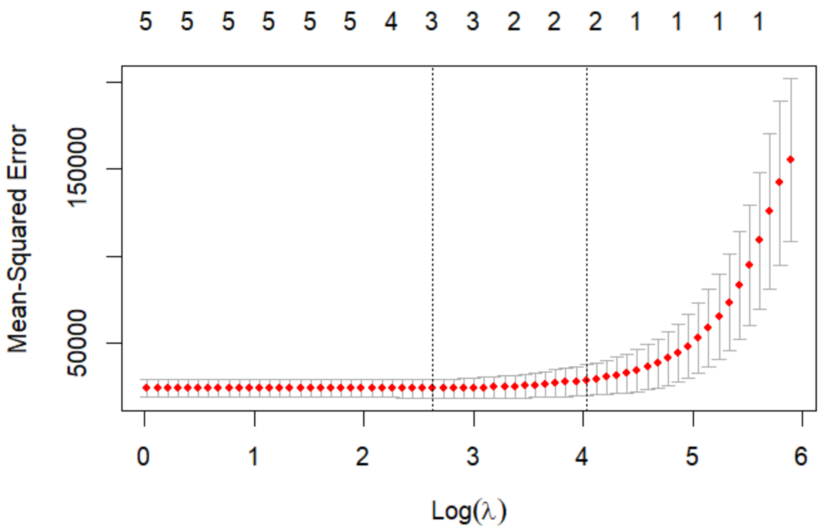
```
> durbinWatsonTest(finalModel)
 lag Autocorrelation D-W Statistic p-value
   1       0.2012826      1.573363   0.078
 Alternative hypothesis: rho != 0
```

The test above concludes that the residuals in this regression model are not autocorrelated because we don't reject H_0 hypothesis.

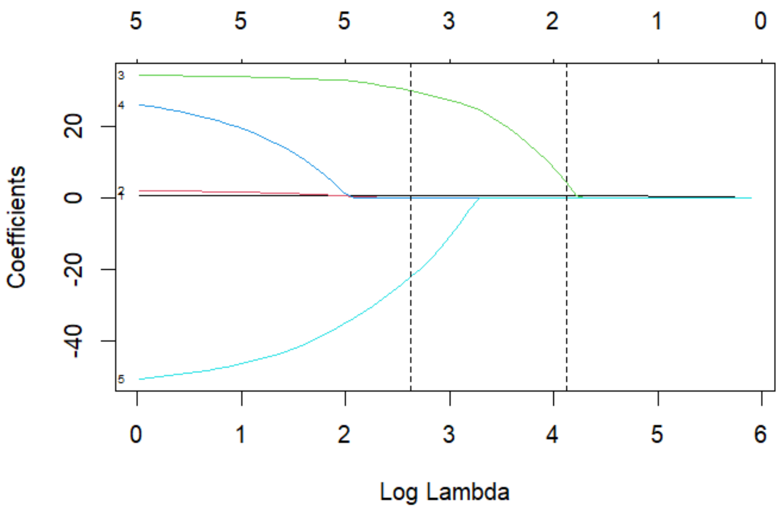


As for the independency of errors, we can see that in our case, there is an independency.

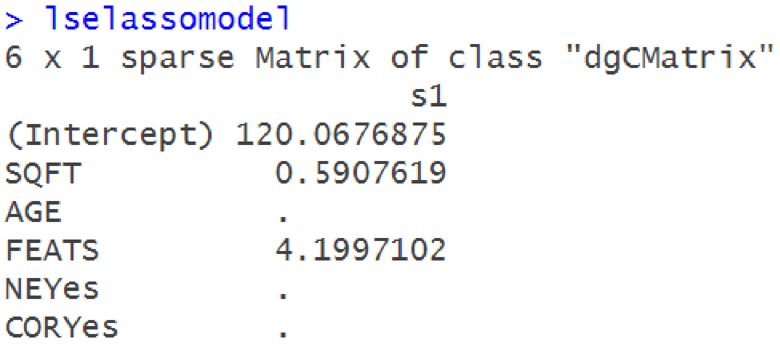
Question 9: Conduct LASSO as a variable selection technique and compare the variables that you end up having using LASSO to the variables that you ended up having using stepwise methods in (VI). Are you getting the same results? Comment.



As we can see from the plot, we need to get a simple model with 2 variables that is won't hurt accuracy compared to minimum. In other words, error is within 1 standard error of the minimum.



Moreover, the above plot suggest us to keep only the 1st column and the 3rd column for our model which is SQFT and FEATS. Looks like we are getting the same variables after conducting LASSO if we choose to go with the simple LSE model of 2 variables. Looking at the graph we do not notice a large deviation that would prompt us to avoid using LSE as our model. The final model from LASSO is:

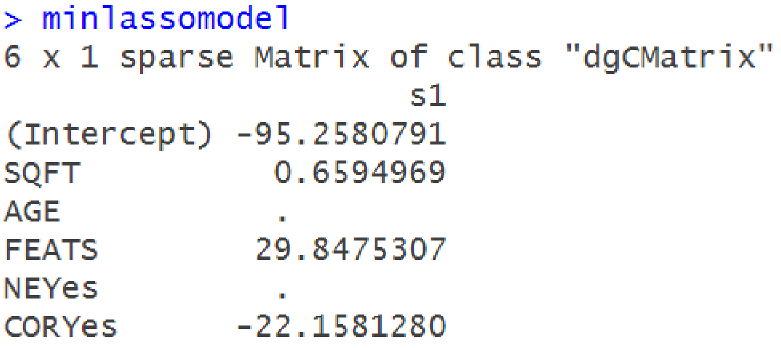


```
> lselassomodel
6 x 1 sparse Matrix of class "dgCMatrix"
                     s1
(Intercept) 120.0676875
SQFT          0.5907619
AGE           .
FEATS         4.1997102
NEYes         .
CORYes        .
```

And our model is:

Price = 120,07 + SQFT * 0,59 + FEAT * 4,2 + e

While the model with the minimum error is:



```
> minlassomodel
6 x 1 sparse Matrix of class "dgCMatrix"
                     s1
(Intercept) -95.2580791
SQFT          0.6594969
AGE           .
FEATS        29.8475307
NEYes         .
CORYes      -22.1581280
```

Price = -95,26 + SQFT * 0,66 + FEAT * 29,85 -22,16 if in location COR + e with 3 attributes.

In conclusion, the min error lasso model seems to be closer in comparison to our linear model approach.

Code:

```
# 1

usdata <- read.table("usdata", sep = " ")

str(usdata)

# 2

usdata$PRICE <- as.numeric(usdata$PRICE)
```

```
usdata$SQFT <- as.numeric(usdata$SQFT)
usdata$AGE <- as.numeric(usdata$AGE)
usdata$FEATS <- as.numeric(usdata$FEATS)
# legend='NE = Located in northeast section of city, COR = Corner location')
usdata$NE <- as.factor(usdata$NE)
levels(usdata$NE) <- list("No" = "0", "Yes" = "1")
usdata$COR <- as.factor(usdata$COR)
levels(usdata$COR) <- list("No" = "0", "Yes" = "1")
# 3
summary(usdata)
# Visual analysis
par(mfrow=c(2,2));n=nrow(usdata)
hist(usdata$PRICE, main=names(usdata)[1])
hist(usdata$SQFT, main=names(usdata)[2])
plot(table(usdata$AGE)/n, type='h',xlim=range(usdata$AGE)+c(-1,1), main="Age", ylab='Relative frequency')
plot(table(usdata$FEATS)/n, type='h',xlim=range(usdata$FEATS)+c(-1,1), main="Feats", ylab='Relative frequency')
par(mfrow=c(1,1))
barplot(sapply(usdata[, c(5,6)],table)/n, horiz=T, las=1, col=2:3, ylim=c(0,8), cex.names=1.3)
legend('top', fil=2:3, legend=c('No','Yes'), ncol=2, bty='n',cex=1.5)
# 4
pairs(usdata)
require(corrplot)
corrplot(cor(usdata[c(1:4)]),method = 'number')
par(mfrow=c(1,5))
for(j in 2:6){
  plot(usdata[,j], usdata[,1], xlab=names(usdata)[j], ylab='Price',cex.lab=1.5)
  abline(lm(usdata[,1]~usdata[,j]),col=2)
}
par(mfrow=c(1,1))
boxplot(usdata[,1]~usdata$FEATS, xlab="FEATS", ylab='Price',cex.lab=1.5)
abline(lm(usdata[,1]~usdata$FEATS),col=2)
par(mfrow = c(1,1))
corrplot(cor(usdata[,c(1:4)]), method = "number")
# 5
model = lm(PRICE ~ ., data = usdata)
```

```
summary(model)

# 6 & 7

finalModel = step(model, direction='both')

summary(finalModel)

require(psych)

index <- sapply(usdata, class) == "numeric"

usdatanum <- usdata[,index]

usdata2 <- as.data.frame(scale(usdatanum, center = TRUE, scale = F))

round(sapply(usdata2,mean),5)

round(sapply(usdata2,sd),2)

model2 = lm(PRICE ~ ., data = usdata2)

summary(model2)

centralizedfinalModel = step(model2, direction='both')

summary(centralizedfinalModel)

round(centralizedfinalModel$coefficients,2)

# 8

plot(finalModel, which = 2) #Normality of the residuals

library(nortest)

lillie.test(rstandard(finalModel))

shapiro.test(rstandard(finalModel))

Stud.residuals = rstudent(finalModel)

yhat = fitted(finalModel)

par(mfrow=c(1,2))

plot(yhat, Stud.residuals)

abline(h=c(-2,2), col=2, lty=2)

plot(yhat, Stud.residuals^2)

abline(h=4, col=2, lty=2)

library(car)

par(mfrow=c(1,1))

ncvTest(finalModel)

yhat.quantiles=cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)), dig.lab=6)

table(yhat.quantiles)

leveneTest(rstudent(finalModel)~yhat.quantiles)

boxplot(rstudent(finalModel)~yhat.quantiles)
```

```
residualPlot(finalModel, type='rstudent')

residualPlots(finalModel, plot=F, type = "rstudent")

durbinWatsonTest(finalModel)

plot(rstudent(finalModel), type='l')

# 9

require(glmnet)

X = model.matrix(model)[,-1]

lasso = cv.glmnet(X, usdata$PRICE,alpha = 1)

plot(lasso)

lasso$lambda

lasso$lambda.min

lasso$lambda.1se

minlassomodel <- coef(lasso, s = "lambda.min")

minlassomodel

svlasso <- minlassomodel[-1] * apply(X,2,sd)

svols <- coef(finalModel)[-1] * apply(X,2,sd)

s <- sum( abs( svlasso))/sum( abs(svols))

s

lselassomodel <- coef(lasso, s = "lambda.1se")

lselassomodel

svlasso1 <- lselassomodel[-1] * apply(X,2,sd)

svols1 <- coef(finalModel)[-1] * apply(X,2,sd)

s1 <- sum( abs( svlasso1))/sum( abs(svols1))

s1

plot(lasso$glmnet.fit, xvar = "lambda", label = T)

abline(v=log(c(lasso$lambda.min, lasso$lambda.1se)), lty =2)
```