Project Report

Importing Libraries

There are a library to use called faraway we can loading using this following code

```
library('faraway')
library('corrplot')

## corrplot 0.92 loaded

library('mctest')
```

Loading Data

Next we would like to load the data using this following command

```
data <- read.csv('M:/MA335/Test/Lab Test/Real estate.csv')
```

Data Exploration

After we make some data loading we will understand deeper about our dataset

```
summary(data)

##      price            date           age          MRT.station     
##  Min.   : 7.60   Min.   :2013   Min.   : 0.00   Min.   :  23.38  
##  1st Qu.:27.60   1st Qu.:2013   1st Qu.: 9.85   1st Qu.: 289.32  
##  Median :39.40   Median :2013   Median :15.75   Median : 492.23  
##  Mean   :38.40   Mean    2013   Mean   :17.91   Mean   :1118.21  
##  3rd Qu.:47.75   3rd Qu.:2013   3rd Qu.:29.32   3rd Qu.:1403.12  
##  Max.   :73.60   Max.    2014   Max.   :43.80   Max.   :6396.28  
##      stores          latitude       longitude       properties  
##  Min.   : 0.000   Min.   :24.93   Min.   :121.5   Min.   :1.00  
##  1st Qu.: 1.000   1st Qu.:24.96   1st Qu.:121.5   1st Qu.:1.00  
##  Median : 5.000   Median :24.97   Median :121.5   Median :2.00  
##  Mean   : 4.135   Mean   :24.97   Mean   :121.5   Mean   :1.88  
##  3rd Qu.: 6.000   3rd Qu.:24.98   3rd Qu.:121.5   3rd Qu.:2.00  
##  Max.   :10.000   Max.   :25.01   Max.   :121.5   Max.   :3.00  
```

From here we can see that from the price, we get the minimum value of 7.60 and the median of 39.4 aand also the mean of 38.4, for the specific column we can also using slicing method which shown below

```
summary(data[,1])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    7.60   27.60   39.40   38.40   47.75   73.60
```

for the checking of dimension of our data we can also using this methods

```
dim(data)

## [1] 200   8
```

so from here we can now that our dataset have 200 rows and 8 columns

Simple linear regression

We asked to make a model from there and use price as our target variable and MRT station as our features so from here we can use this following code

```
attach(data)



x <- MRT.station
y <- data$price
```

after we define the variables we can make a simple linear models using this following codes

```
model1 <- lm(y~x)
```

After we fitted our model we need to see the summary of our models, we can use this code

```
summary(model1)

## 
## Call:
## lm(formula = y ~ x)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -36.077  -5.859  -0.902   5.853  29.193 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 46.5375666  0.8712541   53.41   <2e-16 ***
## x           -0.0072733  0.0005014  -14.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 9.431 on 198 degrees of freedom
## Multiple R-squared:  0.5152, Adjusted R-squared:  0.5128 
## F-statistic: 210.4 on 1 and 198 DF,  p-value: < 2.2e-16
```

From here, we can see that there are several things that we can see, if we use 0.01 as our CI (convidence interval) /alpha, we can see that our predictors are affecting, because of the pvalue there, but the model r squared is only 51% this means that our model can explain only 51% of the data, we can do further analysis, but from this models we can have an equation of

y = -0.007x + 46.53

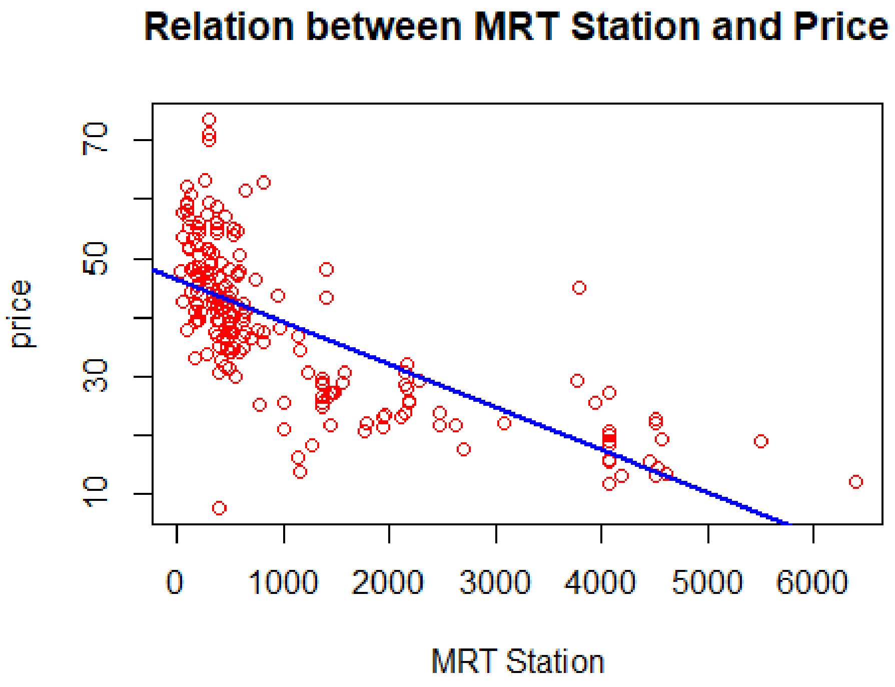
This shows that our model are not have any positive correlation because the gradient are below 0, this means that this have negative correlation, we can also check it using this code

```
cor(x,y)

## [1] -0.7177779
```

in here we can interpret that our model have negative relation, its mean that if the price is going up then the mrt station is going down. for the correlation it self the changing of mrt station are pretty affected the price also.

We can also plot the data using this code below

```
plot(x,y,col='red',xlab='MRT Station',ylab='price',main='Relation between MRT
Station and Price')
abline(model1,lwd=2,col='blue')
```



Multiple linear regression

From here we know that maybe there is more predictors that we can include, we can also using multi linear regression using this code

```
model_multi <-
lm(y~date+age+MRT.station+stores+latitude+longitude+properties)
```

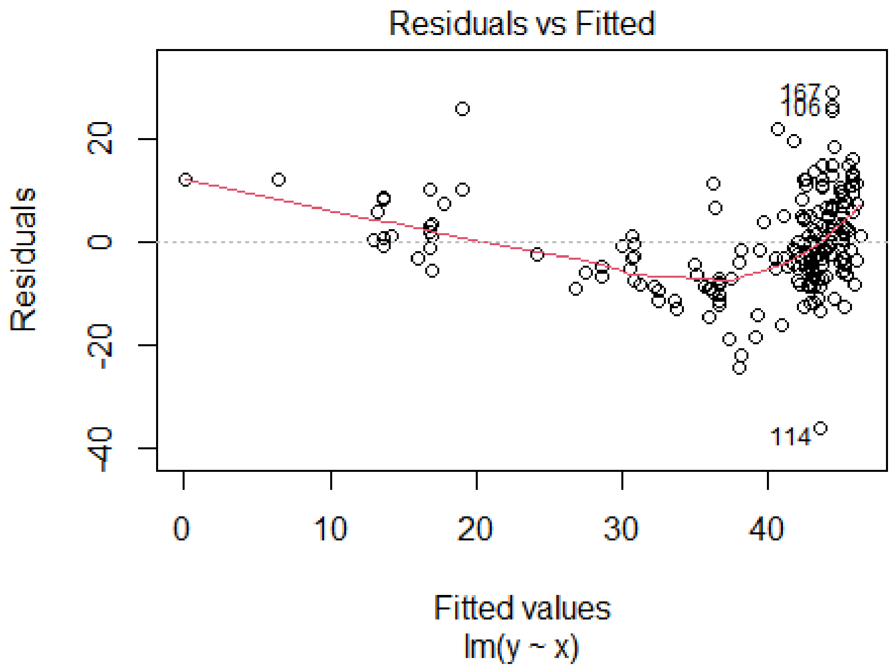
and we can also see the summary using this code

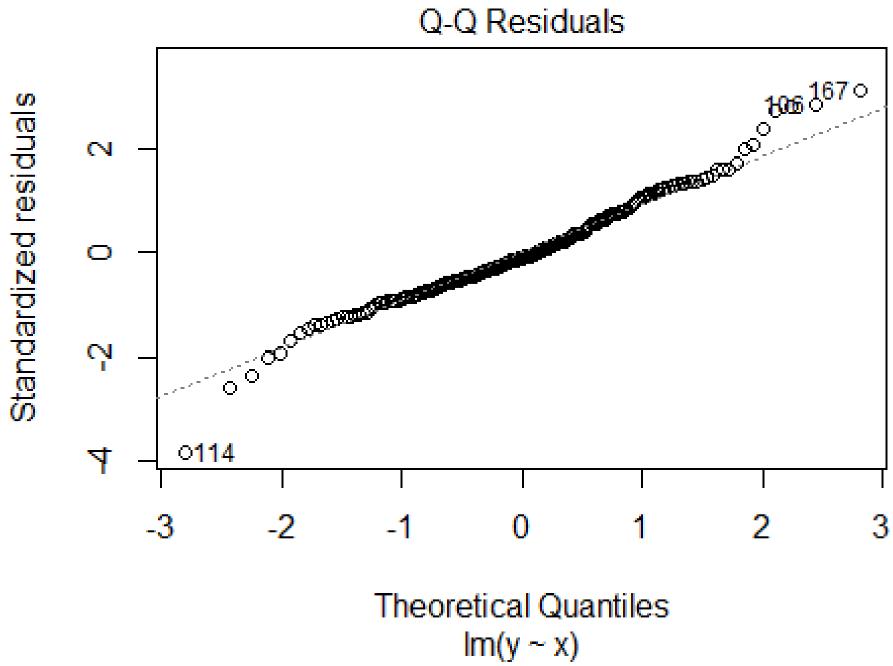
```
summary(model_multi)

##
## Call:
## lm(formula = y ~ date + age + MRT.station + stores + latitude +
##     longitude + properties)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.570  -3.156  -0.146   3.295  25.192
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.244e+03  1.066e+04  -0.773 0.440290
## date         2.959e+00  1.730e+00   1.710 0.088812 .
## age         -2.100e-01  4.568e-02  -4.597 7.76e-06 ***
## MRT.station -3.950e-03  9.940e-04  -3.973 0.000100 ***
## stores       8.048e-01  2.364e-01   3.404 0.000809 ***
## latitude     2.012e+02  5.023e+01   4.006 8.82e-05 ***
## longitude   -2.227e+01  8.438e+01  -0.264 0.792142
## properties   6.023e+00  7.079e-01   8.508 4.98e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.945 on 192 degrees of freedom
## Multiple R-squared:  0.7451, Adjusted R-squared:  0.7358
## F-statistic: 80.17 on 7 and 192 DF,  p-value: < 2.2e-16
```

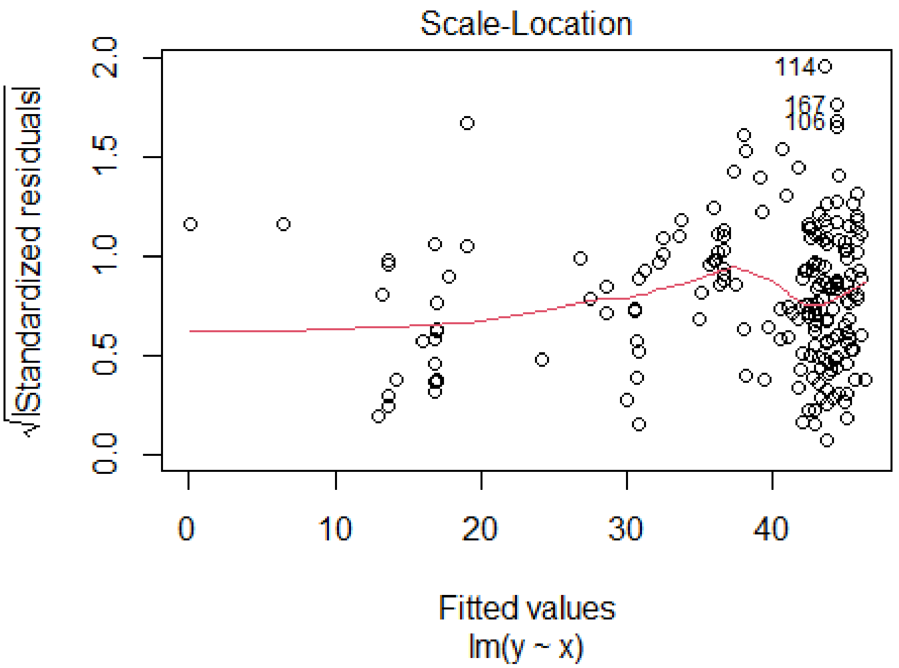
From here we can see that the R squared we can see its better then our models, so it can fit the data betters compared to our first models, but there are some of our models that not statistically significant into our dataset, we can see from there if we take CI of 1% we can see that there are some features that not statistically important.

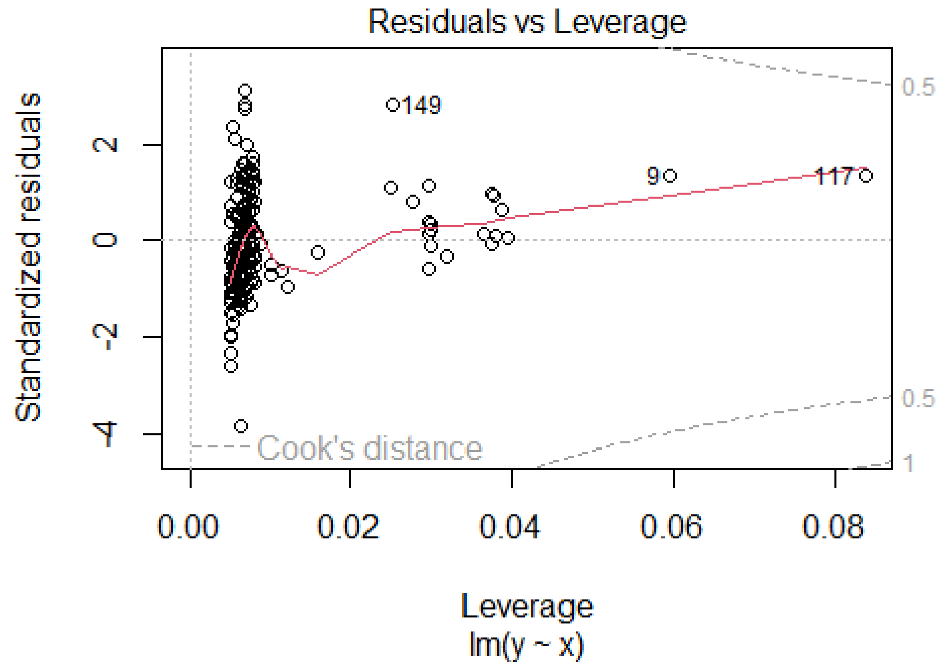
But for future understanding we can plotting our models

```
plot(model1)
```


Residuals vs Fitted
Residuals
20
0
-20
-40
0
10
20
30
40
Fitted values
lm(y ~ x)
167
106
114

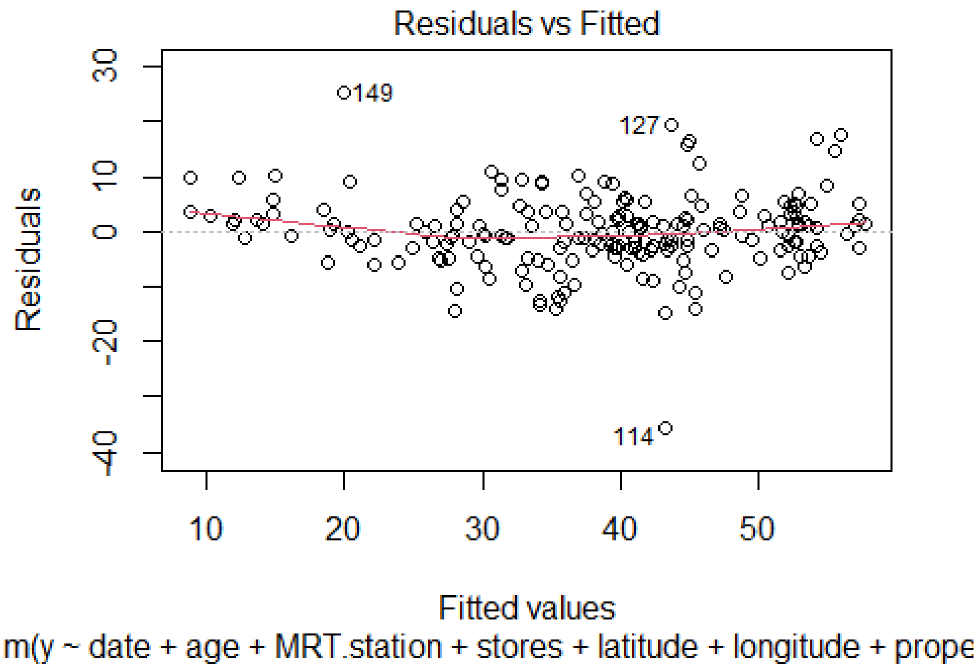

Q-Q Residuals
Standardized residuals
2
0
-2
-4
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(y ~ x)
106
167
114

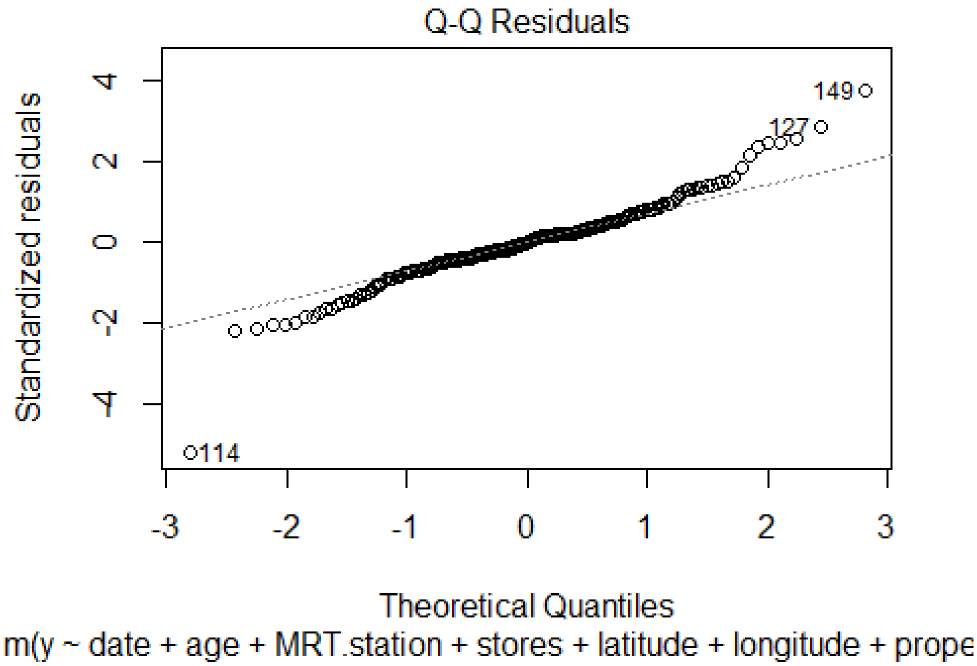


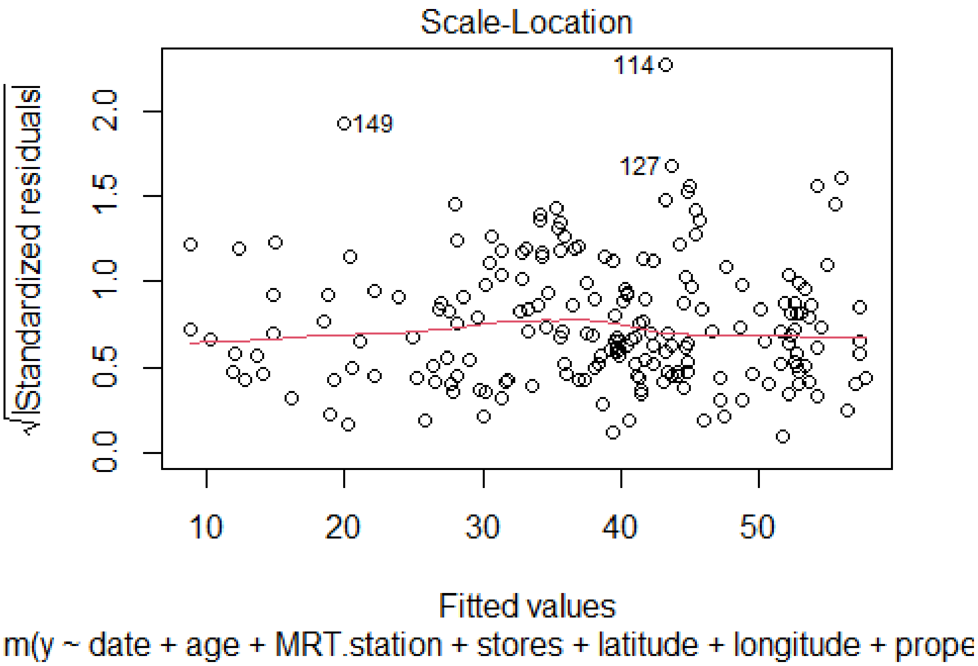


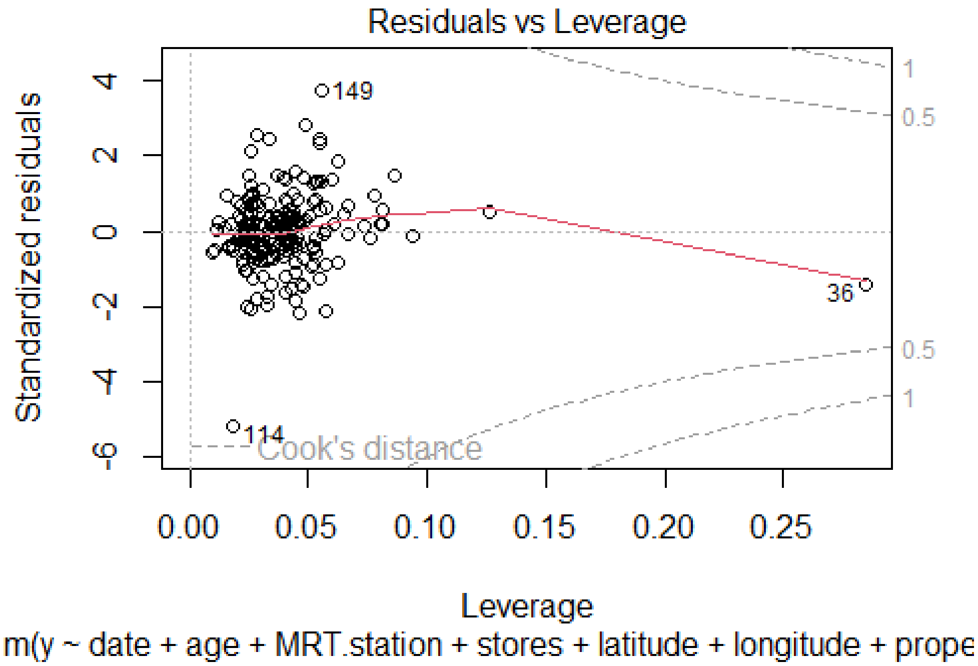
Compared to our multi models

```
plot(model_multi)
```


Residuals vs Fitted
Residuals
30
10
0
-20
-40
149
127
114
10
20
30
40
50
Fitted values
m(y ~ date + age + MRT.station + stores + latitude + longitude + prope


Q-Q Residuals
Standardized residuals
4
2
0
-2
-4
149
127
114
-3
-2
-1
0
1
2
3
Theoretical Quantiles
m(y ~ date + age + MRT.station + stores + latitude + longitude + prope





From here we can see that our models are more stable, because from the residuals vs fited plot we can see that the movement of our residuals are more stables instead of our simple regression models, this can be interpreted as our multi also have more balanced residuals, this also confirmed by our Rsquared values.

Collinearity Check

From the collinearity check first, we can check it using our summary models

```
summary(model_multi)

##
## Call:
## lm(formula = y ~ date + age + MRT.station + stores + latitude +
##     longitude + properties)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.570  -3.156  -0.146   3.295  25.192
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.244e+03  1.066e+04  -0.773 0.440290
## date         2.959e+00  1.730e+00   1.710 0.088812 .
## age         -2.100e-01  4.568e-02  -4.597 7.76e-06 ***
## MRT.station -3.950e-03  9.940e-04  -3.973 0.000100 ***
## stores       8.048e-01  2.364e-01   3.404 0.000809 ***
## latitude     2.012e+02  5.023e+01   4.006 8.82e-05 ***
## longitude   -2.227e+01  8.438e+01  -0.264 0.792142
## properties   6.023e+00  7.079e-01   8.508 4.98e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.945 on 192 degrees of freedom
## Multiple R-squared:  0.7451, Adjusted R-squared:  0.7358
## F-statistic: 80.17 on 7 and 192 DF,  p-value: < 2.2e-16
```

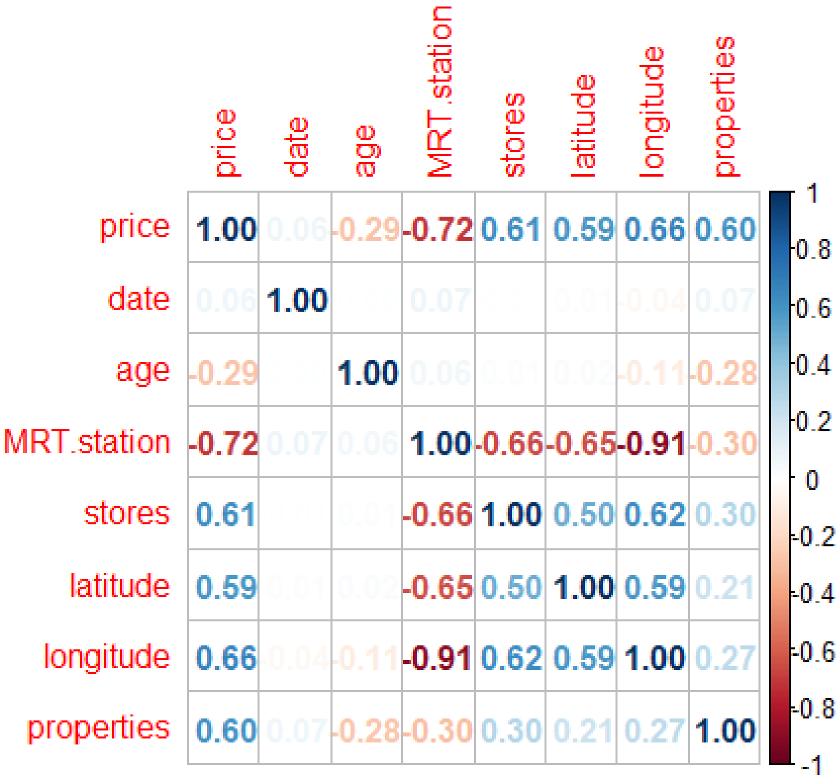
so in here, there might be a collinearity between them, so we can use correlation analysis for our features using corelation function below

```
cor(data[,-1])

##                      date           age MRT.station       stores
latitude
## date         1.0000000000  0.0002444424  0.07024104 -0.006477008
0.01224001
## age          0.0002444424  1.0000000000  0.05886413  0.014621421
0.02348799
## MRT.station  0.0702410444  0.0588641325  1.00000000 -0.664151558 -
0.64892424
## stores      -0.0064770082  0.0146214213 -0.66415156  1.000000000
0.49915044
## latitude     0.0122400117  0.0234879937 -0.64892424  0.499150439
1.00000000
## longitude   -0.0364854554 -0.1091813395 -0.90993452  0.623025488
0.58587149
```

```
## properties   0.0657462407 -0.2774521738 -0.30237579  0.296172884
0.21380001
##               longitude  properties
## date        -0.03648546  0.06574624
## age         -0.10918134 -0.27745217
## MRT.station -0.90993452 -0.30237579
## stores       0.62302549  0.29617288
## latitude     0.58587149  0.21380001
## longitude    1.00000000  0.26757864
## properties   0.26757864  1.00000000
```

or we can also plot it using this function

```
corrplot(cor(data),method='number')
```



	price	date	age	MRT.station	stores	latitude	longitude	properties
price	1.00	0.06	-0.29	-0.72	0.61	0.59	0.66	0.60
date	0.06	1.00	0.00	0.07	-0.01	0.01	-0.04	0.07
age	-0.29	0.00	1.00	0.06	0.01	0.02	-0.11	-0.28
MRT.station	-0.72	0.07	0.06	1.00	-0.66	-0.65	-0.91	-0.30
stores	0.61	-0.01	0.01	-0.66	1.00	0.50	0.62	0.30
latitude	0.59	0.01	0.02	-0.65	0.50	1.00	0.59	0.21
longitude	0.66	-0.04	-0.11	-0.91	0.62	0.59	1.00	0.27
properties	0.60	0.07	-0.28	-0.30	0.30	0.21	0.27	1.00

and also we can confirm it using vif value using this function

```
vif(data[,-1])
```

```
##        date         age MRT.station      stores    latitude   longitude 
##    1.025425    1.132183    7.247759    1.881644    1.772472    6.010217 
##  properties 
##    1.237575
```

But for the testing that more validate to check is there any multicorrelation between those value, we can use a function called mctest below

```
mctest(model_multi)
```

```
##
## Call:
## omcdiag(mod = mod, Inter = TRUE, detr = detr, red = red, conf = conf,
##     theil = theil, cn = cn)
##
##
## Overall Multicollinearity Diagnostics
##
##                          MC Results detection
## Determinant |X'X|:          0.0420         0
## Farrar Chi-Square:        620.9224         1
## Red Indicator:              0.3824         0
## Sum of Lambda Inverse:     20.3073         0
## Theil's Method:            -1.5370         0
## Condition Number:       76961.0823         1
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

Yes the multicollinearity are detected, for suggestion, we can make a smaller models using this function, we will keep date age and mrt to here

```
model2 <- lm(y~age+MRT.station+stores)
model3 <- lm(y~age+MRT.station+latitude)
model4 <- lm(y~age+MRT.station+longitude)
model5 <- lm(y~age+MRT.station+properties)
model6 <- lm(y~MRT.station+properties)
model7 <- lm(y~age+MRT.station+stores+latitude+properties)
```

and after that we can perform anova test to check if preferable or not to use the smaller features models

```
anova(model7,model_multi)

## Analysis of Variance Table
##
## Model 1: y ~ age + MRT.station + stores + latitude + properties
## Model 2: y ~ date + age + MRT.station + stores + latitude + longitude +
##     properties
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    194 9402.4
## 2    192 9260.3  2     142.1 1.4731 0.2318

summary(model7)

##
## Call:
## lm(formula = y ~ age + MRT.station + stores + latitude + properties)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.921  -3.698  -0.365   3.036  26.012 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -5.144e+03  1.254e+03  -4.102 6.03e-05 ***
## age         -2.072e-01  4.525e-02  -4.580 8.30e-06 ***
## MRT.station -3.619e-03  5.744e-04  -6.300 1.96e-09 ***
## stores       8.128e-01  2.361e-01   3.442 0.000707 ***
## latitude     2.073e+02  5.022e+01   4.127 5.46e-05 ***
## properties   6.138e+00  7.050e-01   8.707 1.36e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 6.962 on 194 degrees of freedom
## Multiple R-squared: 0.7412, Adjusted R-squared:  0.7345 
## F-statistic: 111.1 on 5 and 194 DF,  p-value: < 2.2e-16
```

From here we can see that we cannot reject the null hyphotesis so we can use the smaller models based on the pvalue.

One way anova

we can check is there any differences between average price of each properties type, first we can see the group means by applying this function

```
group_mean <- tapply(price,properties,mean)
group_mean

##        1        2        3 
## 32.08630 34.17692 54.54694
```

So from here we can also using the aov test to do that

```
cek <- as.factor(data$properties)
aov(price~cek)

## Call:
##    aov(formula = price ~ cek)
## 
## Terms:
##                      cek Residuals
## Sum of Squares  17076.52  19250.67
## Deg. of Freedom        2       197
## 
## Residual standard error: 9.885298
## Estimated effects may be unbalanced
```

we can use null hyphotesis as “there are no differences between group means”

```
summary(aov(price~cek))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)    
## cek           2  17077    8538   87.38 <2e-16 ***
## Residuals   197  19251      98                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

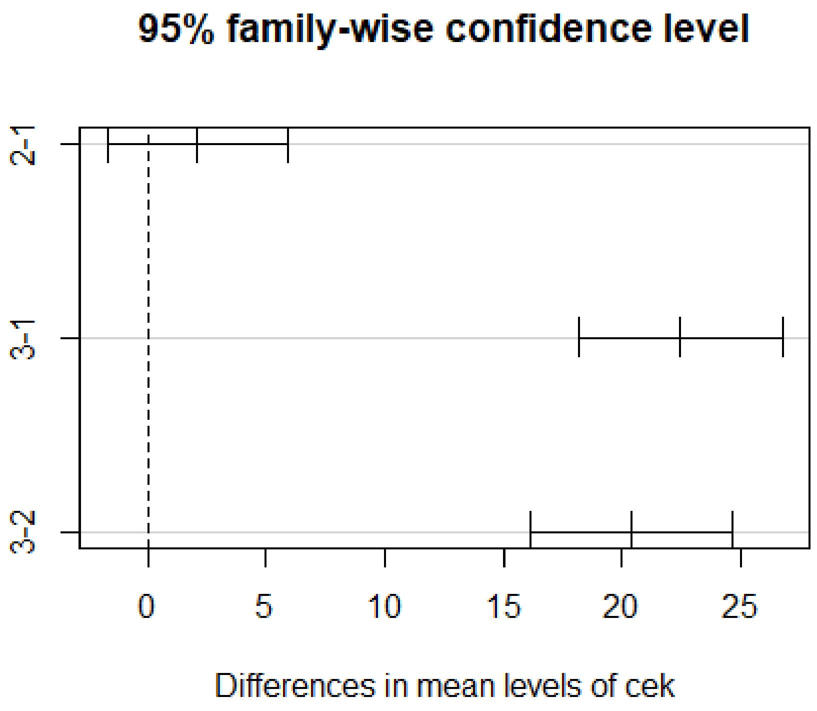
So from here we can reject our null hyphotesis that there are no differences means between those, we can also use the post hoc test to do the testing

```
TukeyHSD(aov(price~cek))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
## 
## Fit: aov(formula = price ~ cek)
## 
## $cek
##          diff       lwr       upr     p adj
## 2-1  2.090622 -1.711005  5.892248 0.3976061
## 3-1 22.460637 18.149319 26.771956 0.0000000
## 3-2 20.370016 16.114559 24.625473 0.0000000
```

we can plot it using

```
plot(TukeyHSD(aov(price~cek)))
```



so from here we can see that there are major differences between 1-3 and 3-2, so we can also using saphiro test to do if the data comes from normality value using this code

```
shapiro.test(price)

##
##  Shapiro-Wilk normality test
##
## data:  price
## W = 0.98856, p-value = 0.1093
```

so from here we can see that we reject null hyphotesis which the data comes from normal distribution