# Assignment1Report

February 9, 2020

## 0.1 Ashika Prakash Acharya (axa190084)

### 0.1.1 Linear Regression (Melbourne Housing Dataset)

### 0.1.2 Read the dataset

[1]:
```
options(warn=-1)
options(repr.plot.width=10, repr.plot.height=8)
```

[2]:
```
install.packages("tidyverse")
install.packages("corrplot")
install.packages("Metrics")
library(tidyverse)
library(Metrics)
require("corrplot")
```

```
The downloaded binary packages are in
/var/folders/kn/jtmjzb3938740xmg3c8dtmkc0000gn/T//RtmpOHDVNJ/downloaded_packages

The downloaded binary packages are in
/var/folders/kn/jtmjzb3938740xmg3c8dtmkc0000gn/T//RtmpOHDVNJ/downloaded_packages

The downloaded binary packages are in
/var/folders/kn/jtmjzb3938740xmg3c8dtmkc0000gn/T//RtmpOHDVNJ/downloaded_packages

  Attaching packages                                       tidyverse
1.3.0

  ggplot2 3.2.1       purrr   0.3.3
  tibble  2.1.3       dplyr   0.8.4
  tidyr   1.0.2       stringr 1.4.0
  readr   1.3.1       forcats 0.4.0

  Conflicts
tidyverse_conflicts()
  dplyr::filter() masks stats::filter()
  dplyr::lag()    masks stats::lag()

Loading required package: corrplot
```

corrplot 0.84 loaded

[3]:
```
housing = read.csv("https://personal.utdallas.edu/~axa190084/
↪Melbourne_housing_FULL.csv", stringsAsFactors = FALSE, quote = "")
```

[4]:
```
head(housing)
```

A data.frame: 6 × 21

| | Suburb <chr> | Address <chr> | Rooms <chr> | Type <chr> | Price <int> | Method <chr> | SellerG <chr> | Da <c |
|---|---|---|---|---|---|---|---|---|
| 1 | Abbotsford | 68 Studley St | 2 | h | NA | SS | Jellis | 3/( |
| 2 | Abbotsford | 85 Turner St | 2 | h | 1480000 | S | Biggin | 3/1 |
| 3 | Abbotsford | 25 Bloomburg St | 2 | h | 1035000 | S | Biggin | 4/( |
| 4 | Abbotsford | 18/659 Victoria St | 3 | u | NA | VB | Rounds | 4/( |
| 5 | Abbotsford | 5 Charles St | 3 | h | 1465000 | SP | Biggin | 4/( |
| 6 | Abbotsford | 40 Federation La | 3 | h | 850000 | PI | Biggin | 4/( |

[5]:
```
dim(housing)
```

1. 34790 2. 21

### 0.1.3 Eliminate records that do not have price value.

[6]:
```
clean_housing = housing
clean_housing <- clean_housing %>% filter(Price != "")
dim(clean_housing)
```

1. 27194 2. 21

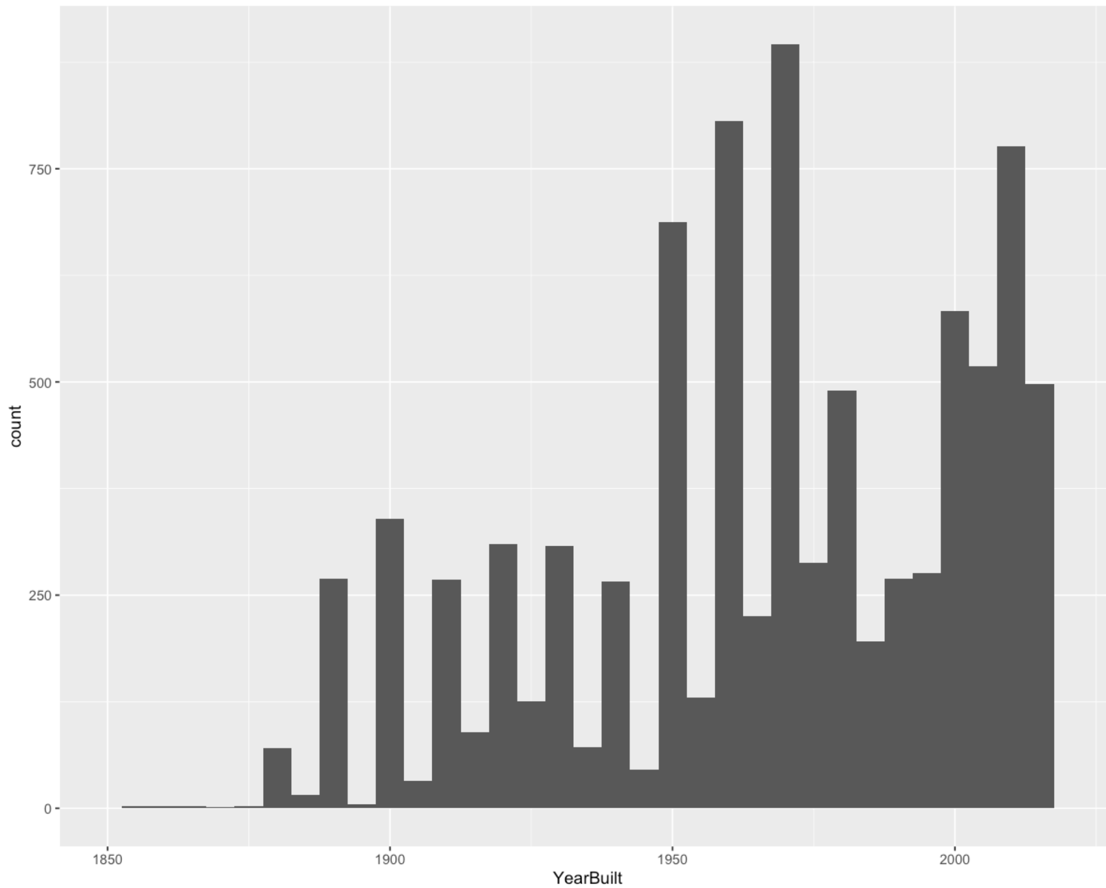
[7]:
```
# code mutation to be able to see corelation between these features.

clean_housing$Rooms =  as.numeric(clean_housing$Rooms)
clean_housing$Price =  as.numeric(clean_housing$Price)
clean_housing$Distance = as.numeric(clean_housing$Distance)
clean_housing$Bathroom = as.numeric(clean_housing$Bathroom)
clean_housing$Car = as.numeric(clean_housing$Car)
clean_housing$Landsize = as.numeric(clean_housing$Landsize)
clean_housing$Longtitude = as.numeric(clean_housing$Longtitude)
clean_housing$BuildingArea = as.numeric(clean_housing$BuildingArea)
clean_housing$Propertycount = as.numeric(clean_housing$Propertycount)
clean_housing$Postcode <- as.numeric(clean_housing$Postcode)
```

### 0.1.4 Some interesting finds and plots to support them

### 0.1.5 Let us see when the houses were built.

[20]:
```
ggplot(data = clean_housing, aes(x= YearBuilt)) + geom_histogram(binwidth = 5)
 ↪+ xlim(1850,2020)
```

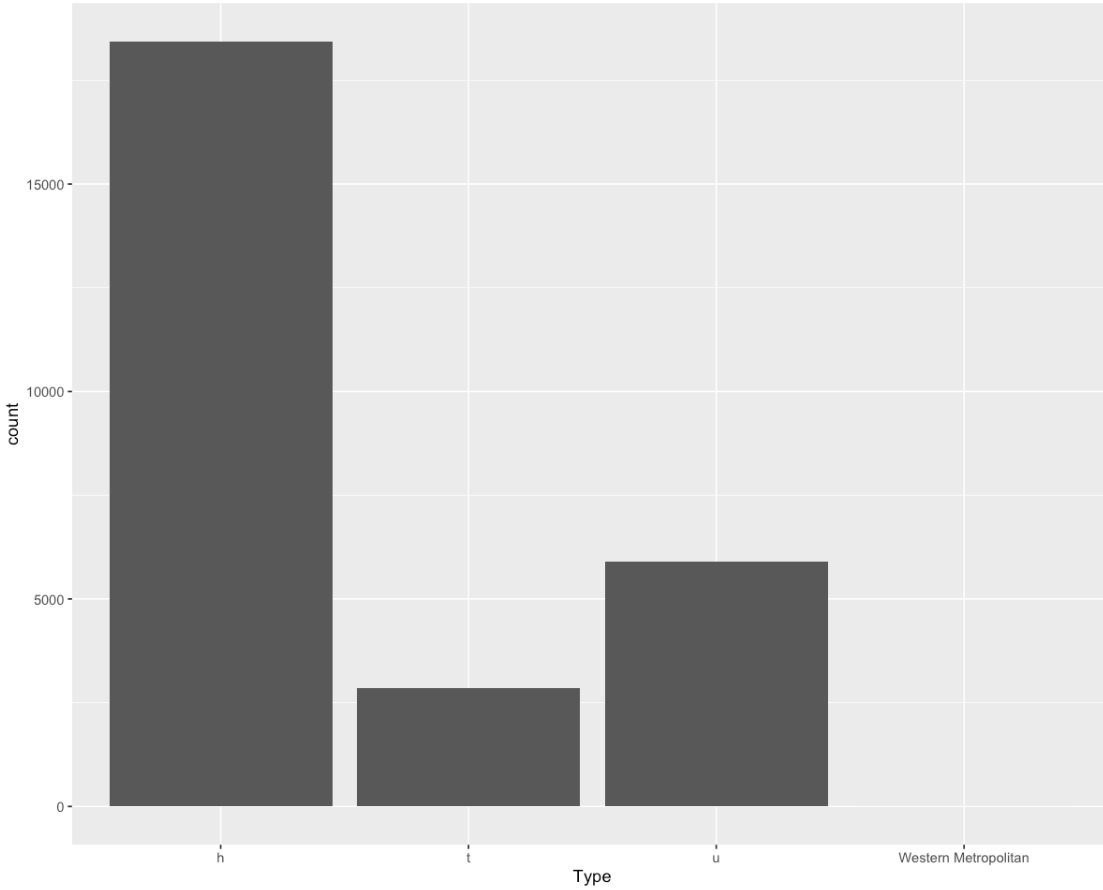



**As seen from the graph, most of the houses were built between 1950-2010**

### 0.1.6 What type of the houses are they?

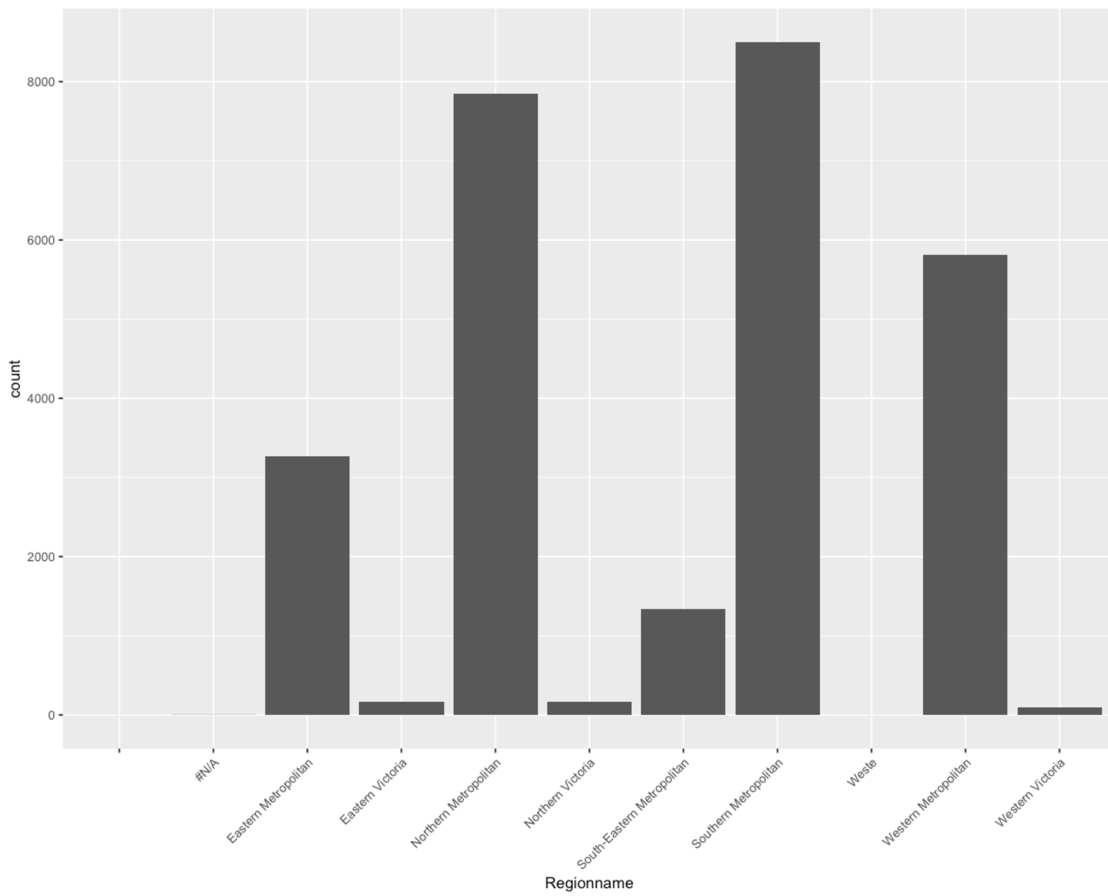
**h-house or t-townhouse or u-apartment unit**

[9]:
```
ggplot(data = clean_housing, aes(x=Type)) + geom_bar()
```



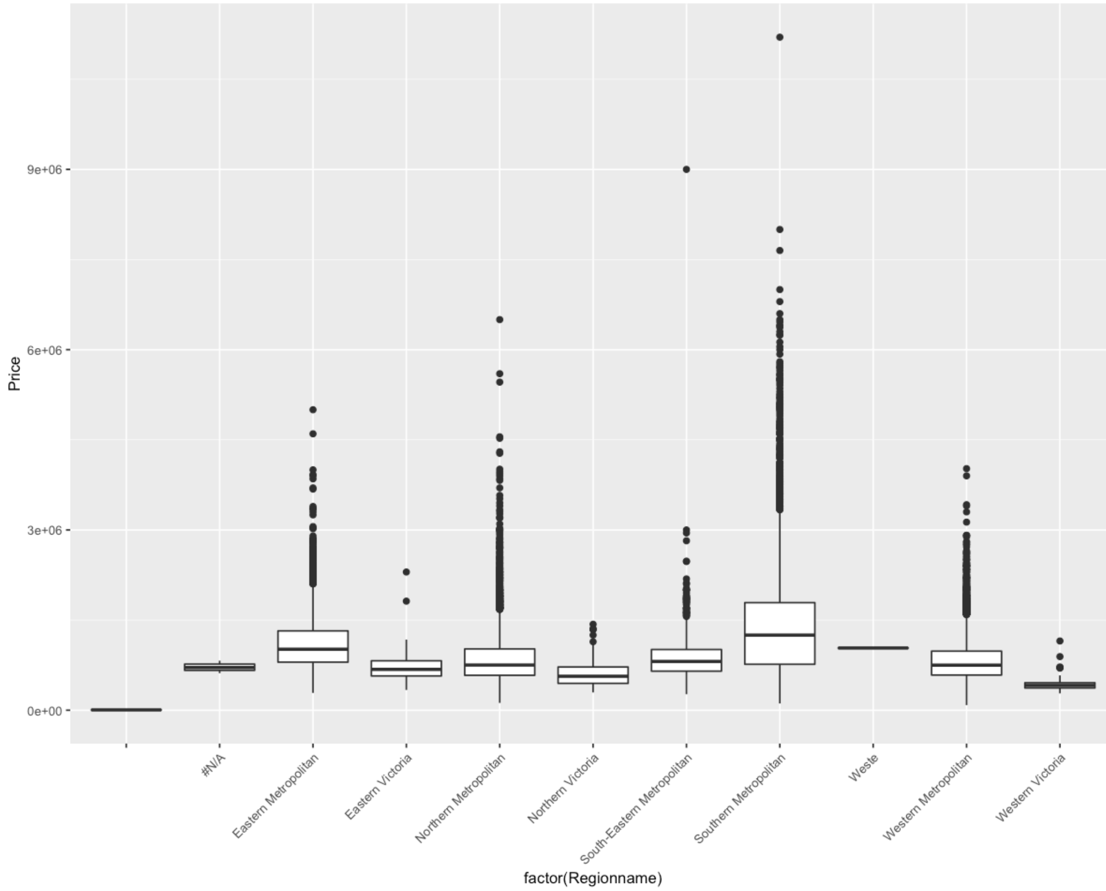

### 0.1.7 Let us see where theses houses are located

[10]:
```
ggplot(data = clean_housing, aes(x = Regionname)) + geom_bar() + theme(text =
    element_text(size=10), axis.text.x = element_text(angle=45, hjust=1))
```




### 0.1.8 Is there any relationship between price and the region

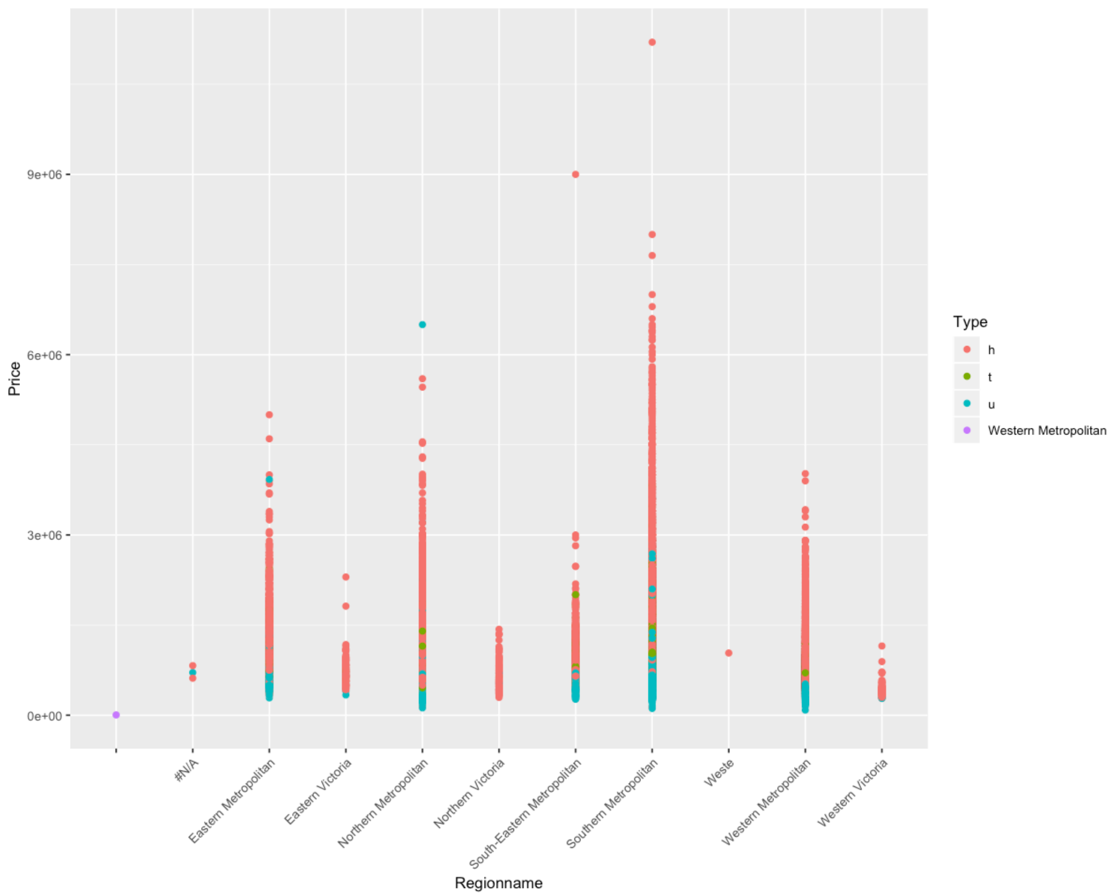
[11]:
```
ggplot(data = clean_housing, aes(x = factor(Regionname), y = Price)) +
  geom_boxplot() + theme(text = element_text(size=10), axis.text.x =
  element_text(angle=45, hjust=1))
```




**From the above box plot, it is evident that houses at Southern Metropolitan is most expensive**

### 0.1.9 Lets see the price of various type of houses in these region

[12]:
```
ggplot(data = clean_housing, aes(x = Regionname, y = Price, color = Type)) +
  geom_point() + theme(text = element_text(size=10), axis.text.x =
  element_text(angle=45, hjust=1))
```




### 0.1.10 Data Cleaning and its statistics

[13]:
```
# omit NA values
clean_housing = na.omit(clean_housing)

dim(clean_housing)
summary(clean_housing)
```

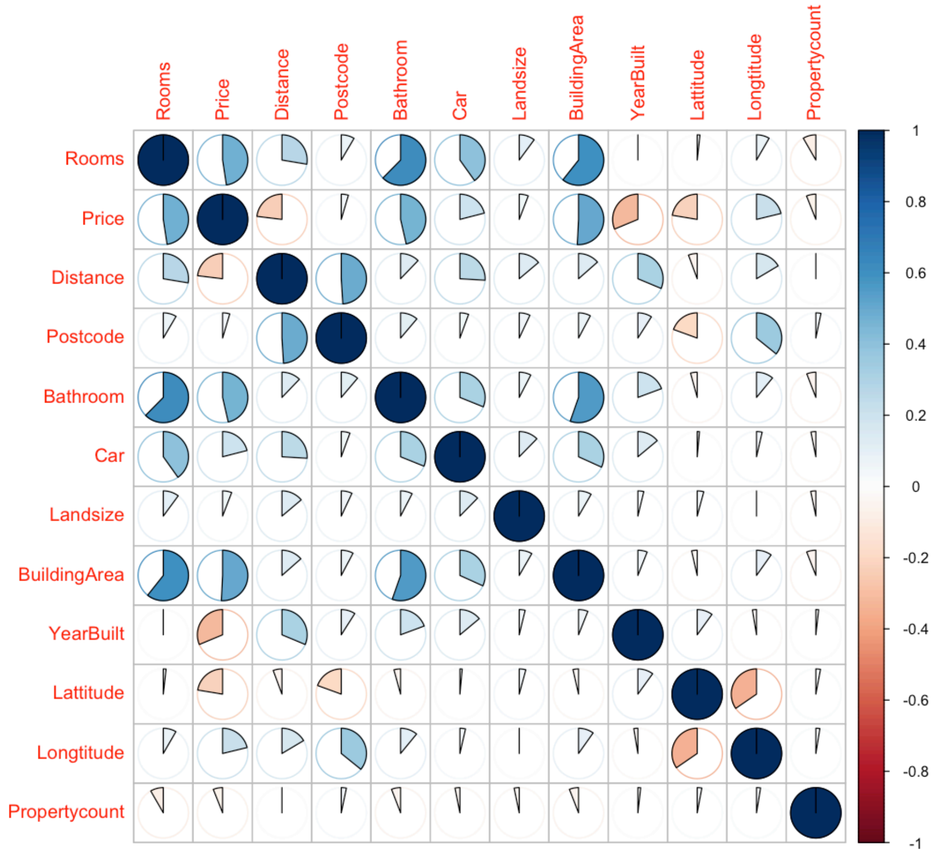
1. 8875 2. 21

```
    Suburb              Address               Rooms              Type
 Length:8875         Length:8875         Min.   : 1.000   Length:8875
 Class :character    Class :character    1st Qu.: 2.000   Class :character
 Mode  :character    Mode  :character    Median : 3.000   Mode  :character
                                         Mean   : 3.099
                                         3rd Qu.: 4.000
                                         Max.   :12.000
     Price             Method              SellerG              Date
 Min.   : 131000   Length:8875         Length:8875         Length:8875
 1st Qu.: 640500   Class :character    Class :character    Class :character
```

```
 Median : 900000   Mode  :character   Mode  :character   Mode  :character
 Mean   :1093091
 3rd Qu.:1346000
 Max.   :9000000
    Distance        Postcode        Bedroom2         Bathroom
 Min.   : 0.0   Min.   :3000   Min.   : 0.000   Min.   :1.000
 1st Qu.: 6.4   1st Qu.:3044   1st Qu.: 2.000   1st Qu.:1.000
 Median :10.2   Median :3084   Median : 3.000   Median :2.000
 Mean   :11.2   Mean   :3112   Mean   : 3.078   Mean   :1.646
 3rd Qu.:13.9   3rd Qu.:3150   3rd Qu.: 4.000   3rd Qu.:2.000
 Max.   :47.4   Max.   :3977   Max.   :12.000   Max.   :9.000
      Car             Landsize        BuildingArea      YearBuilt
 Min.   : 0.000   Min.   :    0.0   Min.   :   0.0   Min.   :1196
 1st Qu.: 1.000   1st Qu.:  212.0   1st Qu.: 100.0   1st Qu.:1945
 Median : 2.000   Median :  478.0   Median : 132.0   Median :1970
 Mean   : 1.692   Mean   :  523.5   Mean   : 149.3   Mean   :1966
 3rd Qu.: 2.000   3rd Qu.:  652.0   3rd Qu.: 180.0   3rd Qu.:2000
 Max.   :10.000   Max.   :42800.0   Max.   :3112.0   Max.   :2019
 CouncilArea          Lattitude        Longtitude     Regionname
 Length:8875        Min.   :-38.17   Min.   :144.4   Length:8875
 Class :character   1st Qu.:-37.86   1st Qu.:144.9   Class :character
 Mode  :character   Median :-37.80   Median :145.0   Mode  :character
                    Mean   :-37.80   Mean   :145.0
                    3rd Qu.:-37.75   3rd Qu.:145.1
                    Max.   :-37.41   Max.   :145.5
 Propertycount
 Min.   :  249
 1st Qu.: 4380
 Median : 6567
 Mean   : 7476
 3rd Qu.:10331
 Max.   :21650
```

### 0.1.11 Let us see how the features are co related to each other

[14]:
```r
numeric_housing = clean_housing[c(3,5,9:10,12:16,18,19,21)]
numeric_housing <- na.omit(numeric_housing)
housing_corelation = cor(numeric_housing)
corrplot(housing_corelation, method = "pie")
```




### 0.1.12 Split data into training and test subset

[15]:
```
#split the data into training and testing data.

set.seed(123)
sample_size = ceiling(nrow(clean_housing) * 0.8)
train_index = sample(nrow(clean_housing), sample_size)

training_data = clean_housing[train_index, ]
test_data = clean_housing[-train_index, ]
```

### 0.1.13 Build the model using training data

[16]:
```
model = lm(Price ~ Rooms + Distance  + Bathroom + BuildingArea + YearBuilt +␣
 ↪Lattitude + Longtitude , data = training_data)
summary(model)
```

```
Call:
lm(formula = Price ~ Rooms + Distance + Bathroom + BuildingArea +
```

```
    YearBuilt + Lattitude + Longtitude, data = training_data)

Residuals:
     Min       1Q   Median       3Q      Max
-3925554  -227546   -48867   146372  8065274

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.428e+08  6.627e+06  -21.55   <2e-16 ***
Rooms         1.665e+05  8.256e+03   20.16   <2e-16 ***
Distance     -3.202e+04  8.782e+02  -36.46   <2e-16 ***
Bathroom      1.894e+05  1.022e+04   18.54   <2e-16 ***
BuildingArea  2.612e+03  8.836e+01   29.56   <2e-16 ***
YearBuilt    -4.713e+03  1.578e+02  -29.86   <2e-16 ***
Lattitude    -1.194e+06  6.280e+04  -19.00   <2e-16 ***
Longtitude    7.395e+05  4.811e+04   15.37   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 444000 on 7092 degrees of freedom
Multiple R-squared:  0.5856,Adjusted R-squared:  0.5852
F-statistic:  1432 on 7 and 7092 DF,  p-value: < 2.2e-16
```

### 0.1.14 The above model seemed to have given the best result in terms of R-squared and F-statistic.

**The behavior of model with few other feature selection.**

<img src="https://personal.utdallas.edu/~axa190084/stats_figure.png" width=700 height=500/>

### 0.1.15 Now that the model is ready, let us validate it with test data.

[17]:
```
predicted_price = data.frame(predict(model, test_data))
```

### 0.1.16 The mse of the model

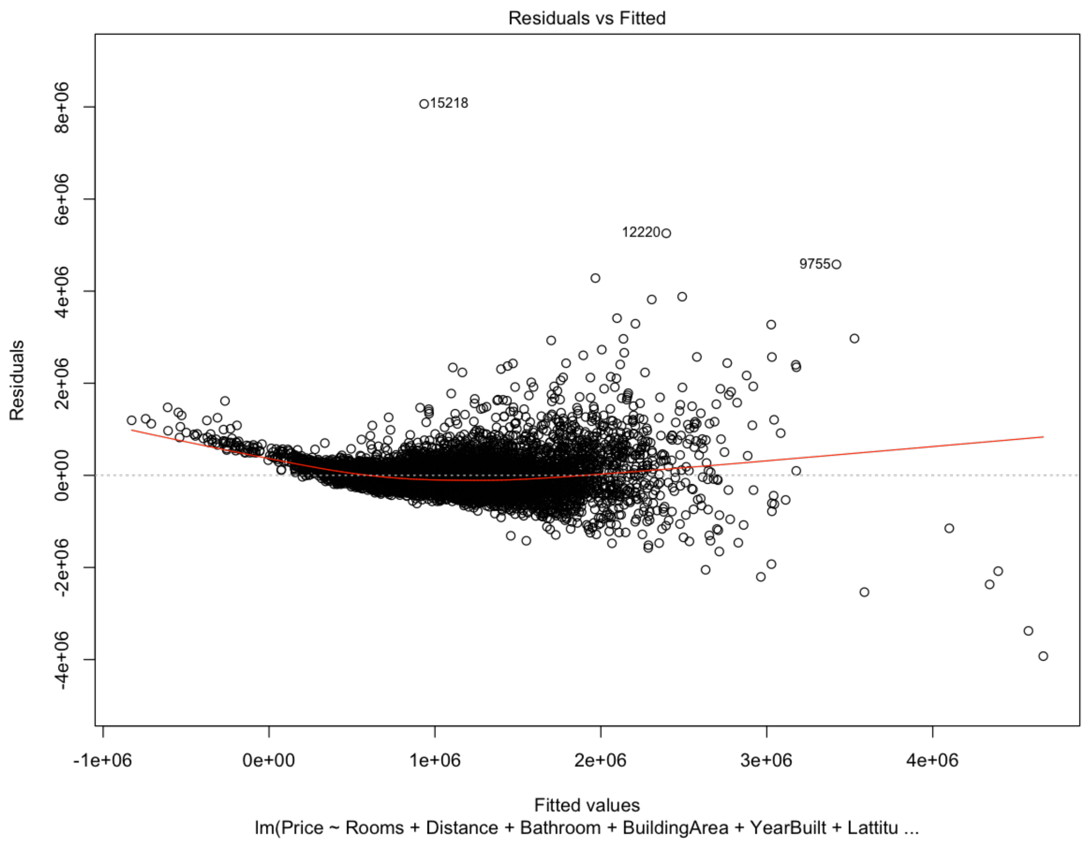
[18]:
```
head(predicted_price)
mse(test_data$Price, predicted_price$predict.model..test_data.)
```

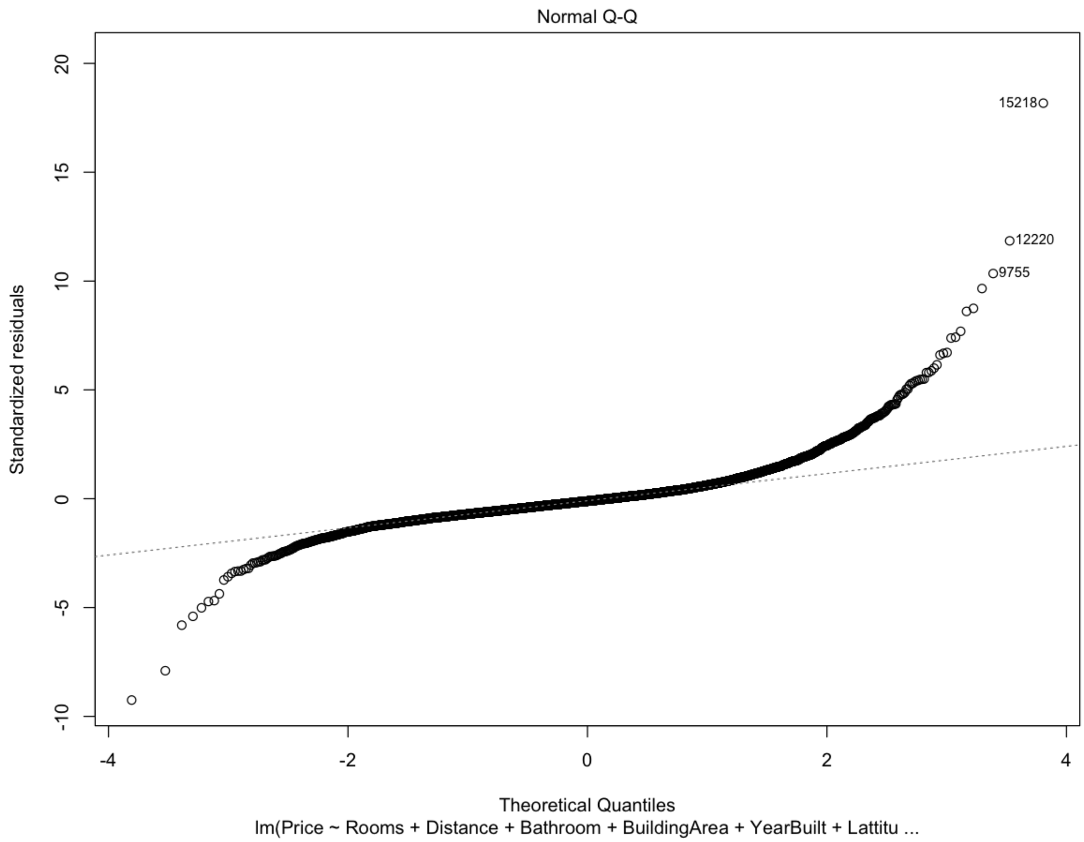
A data.frame: 6 × 1

| | predict.model..test_data.<br><dbl> |
|---|---|
| 45 | 1243353.0 |
| 51 | 1067350.3 |
| 54 | 879789.3 |
| 57 | 651537.0 |
| 68 | 142790.9 |
| 69 | 130393.0 |

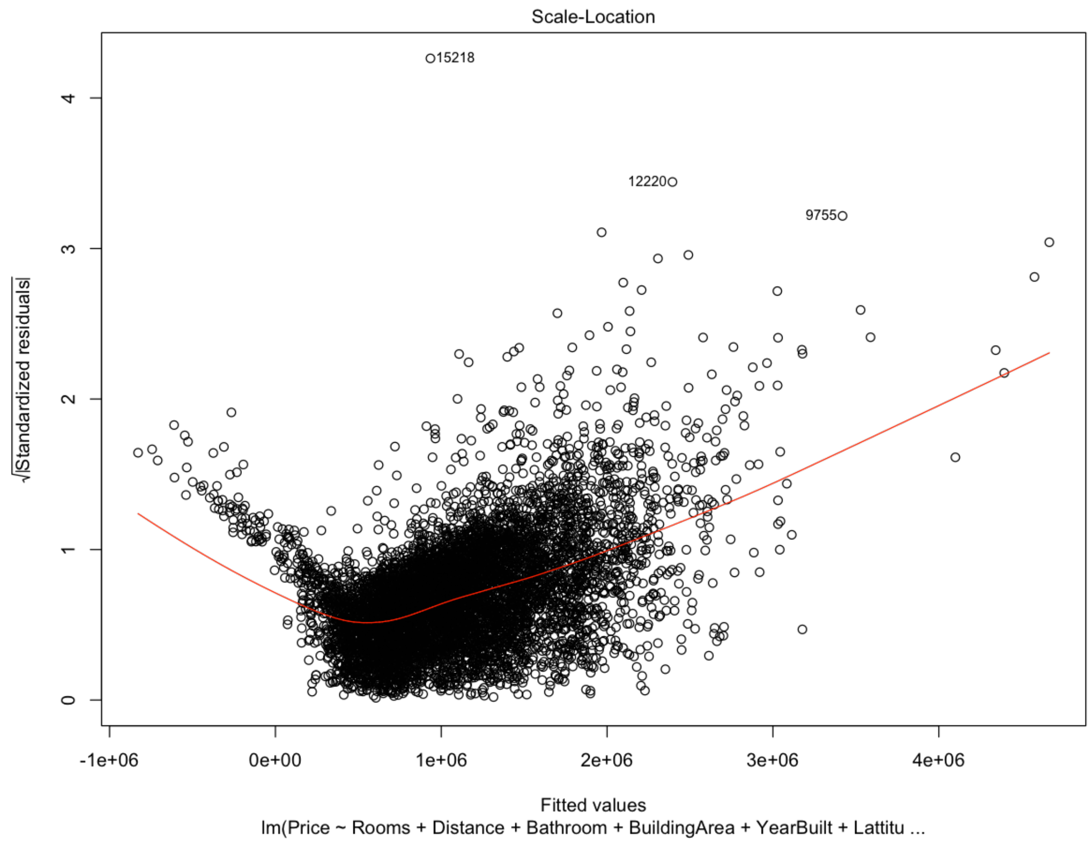
195801339104.108

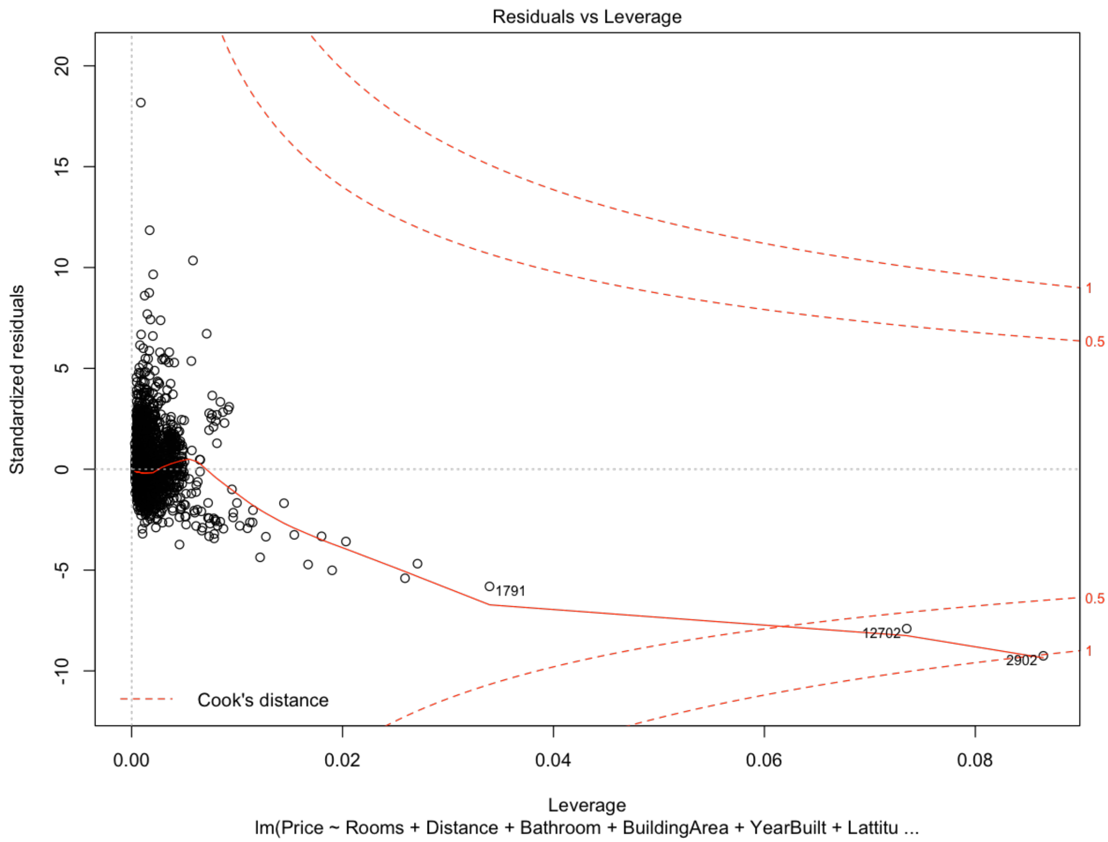
[19]:
```
plot(model)
```





Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Price ~ Rooms + Distance + Bathroom + BuildingArea + YearBuilt + Lattitu ...
15218
12220
9755
20
15
10
5
0
-5
-10
-4
-2
0
2
4


Scale-Location
√|Standardized residuals|
Fitted values
lm(Price ~ Rooms + Distance + Bathroom + BuildingArea + YearBuilt + Lattitu ...
15218
12220
9755
0
1
2
3
4
-1e+06
0e+00
1e+06
2e+06
3e+06
4e+06


Residuals vs Leverage
Standardized residuals
Leverage
lm(Price ~ Rooms + Distance + Bathroom + BuildingArea + YearBuilt + Lattitu ...
Cook's distance
1791
12702
2902
1
0.5
0.5
1
20
15
10
5
0
-5
-10
0.00
0.02
0.04
0.06
0.08