Melbourne_house

Importing the Libraries

```r
require(tidyverse)
```

```
## Loading required package: tidyverse

## Warning: package 'tidyverse' was built under R version 3.6.3

## -- Attaching packages --------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.2
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.6.3

## Warning: package 'tibble' was built under R version 3.6.3

## Warning: package 'dplyr' was built under R version 3.6.3

## -- Conflicts ------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
require(ISLR)
```

```
## Loading required package: ISLR

## Warning: package 'ISLR' was built under R version 3.6.3
```

```r
require(MASS)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
#Loading the Dataset
mel_data <- read.csv('melbourne_data.csv', header = T, stringsAsFactors = F)

mel_house_data <- data.frame(mel_data, stringsAsFactors = F)

class(mel_house_data)
```

```
## [1] "data.frame"
```

visualizing the dataset

```r
head(mel_house_data)
```

```
##        Date Type   Price Landsize BuildingArea Rooms Bathroom Car
## 1  3/9/2016    h      NA      126           NA     2        1   1
## 2 3/12/2016    h 1480000      202           NA     2        1   1
## 3  4/2/2016    h 1035000      156           79     2        1   0
## 4  4/2/2016    u      NA        0           NA     3        2   1
## 5  4/3/2017    h 1465000      134          150     3        2   0
## 6  4/3/2017    h  850000       94           NA     3        2   1
##   YearBuilt Distance            Regionname Propertycount
## 1        NA      2.5 Northern Metropolitan          4019
## 2        NA      2.5 Northern Metropolitan          4019
## 3      1900      2.5 Northern Metropolitan          4019
## 4        NA      2.5 Northern Metropolitan          4019
## 5      1900      2.5 Northern Metropolitan          4019
## 6        NA      2.5 Northern Metropolitan          4019
```

```r
#View(mel_house_data)
```

##Seeing all the column names

```r
colnames(mel_house_data)
```

```
##  [1] "Date"          "Type"          "Price"         "Landsize"
##  [5] "BuildingArea"  "Rooms"         "Bathroom"      "Car"
##  [9] "YearBuilt"     "Distance"      "Regionname"    "Propertycount"
```

#Seeing the Structure and Descriptive Summary of the dataset

#finding out the structure of the melbourne dataset

```r
str(mel_house_data)
```

```
## 'data.frame':    34857 obs. of  12 variables:
##  $ Date         : chr  "3/9/2016" "3/12/2016" "4/2/2016" "4/2/2016" ...
##  $ Type         : chr  "h" "h" "h" "u" ...
##  $ Price        : int  NA 1480000 1035000 NA 1465000 850000 1600000 NA NA NA ...
##  $ Landsize     : int  126 202 156 0 134 94 120 400 201 202 ...
##  $ BuildingArea : num  NA NA 79 NA 150 NA 142 220 NA NA ...
##  $ Rooms        : int  2 2 2 3 3 3 4 4 2 2 ...
```

```
##  $ Bathroom     : int  1 1 1 2 2 2 1 2 1 2 ...
##  $ Car          : int  1 1 0 1 0 1 2 2 2 1 ...
##  $ YearBuilt    : int  NA NA 1900 NA 1900 NA 2014 2006 1900 1900 ...
##  $ Distance     : chr  "2.5" "2.5" "2.5" "2.5" ...
##  $ Regionname   : chr  "Northern Metropolitan" "Northern Metropolitan" "Northern Metropolitan" "Nort
##  $ Propertycount: chr  "4019" "4019" "4019" "4019" ...
```

#As we can see here, we are having lots of NA values in most of the attributes. like yearbuilt is not available for lots of the houses. So, first we need to clean our data for doing the EDA of dataset.

#Descriptive Analysis

```r
summary(mel_house_data)
```

```
##      Date               Type               Price
##  Length:34857       Length:34857       Min.   :   85000
##  Class :character   Class :character   1st Qu.:  635000
##  Mode  :character   Mode  :character   Median :  870000
##                                        Mean   : 1050173
##                                        3rd Qu.: 1295000
##                                        Max.   :11200000
##                                        NA's   :7610
##     Landsize         BuildingArea         Rooms           Bathroom
##  Min.   :     0.0   Min.   :    0.0   Min.   : 1.000   Min.   : 0.000
##  1st Qu.:   224.0   1st Qu.:  102.0   1st Qu.: 2.000   1st Qu.: 1.000
##  Median :   521.0   Median :  136.0   Median : 3.000   Median : 2.000
##  Mean   :   593.6   Mean   :  160.3   Mean   : 3.031   Mean   : 1.625
##  3rd Qu.:   670.0   3rd Qu.:  188.0   3rd Qu.: 4.000   3rd Qu.: 2.000
##  Max.   :433014.0   Max.   :44515.0   Max.   :16.000   Max.   :12.000
##  NA's   :11810      NA's   :21115                      NA's   :8226
##       Car           YearBuilt       Distance          Regionname
##  Min.   : 0.000   Min.   :1196    Length:34857       Length:34857
##  1st Qu.: 1.000   1st Qu.:1940    Class :character   Class :character
##  Median : 2.000   Median :1970    Mode  :character   Mode  :character
##  Mean   : 1.729   Mean   :1965
##  3rd Qu.: 2.000   3rd Qu.:2000
##  Max.   :26.000   Max.   :2106
##  NA's   :8728     NA's   :19306
##  Propertycount
##  Length:34857
##  Class :character
##  Mode  :character
##
##
##
##
```

with the help of descriptive analysis of the dataset we can get the statistical perspective. For example like, we can see that Building Areas can be of maximum 44515 but if you will the mean of all the house, it is 160 which means we have some outliers as well in the dataset. So, we need to preprocessing to clean the dataset.

Data Preprocessing Step : Removing NA Values from the dataset :

#importing library

```
library(ggplot2)
mel_house_data <- data.frame(lapply(mel_house_data,function(x) { gsub("#N/A", NA, x) }))

class(mel_house_data)
```

```
## [1] "data.frame"
```

#Finding Count of Na values in each column

```
NA_count_of_each_col<-sapply(mel_house_data,function(x) sum(is.na(x)==TRUE))
NA_count_of_each_col
```

```
##          Date          Type         Price      Landsize  BuildingArea
##             0             0          7610         11810         21115
##         Rooms      Bathroom           Car     YearBuilt      Distance
##             0          8226          8728         19306             1
##    Regionname Propertycount
##             3             3
```

Find percent of missing in each column

```
for(i in 1:ncol(mel_house_data)) {
  colName <- colnames(mel_house_data[i])
  pctNull <- sum(is.na(mel_house_data[,i]))/length(mel_house_data[,i])
  if (pctNull > 0.50) {
    print(paste("Column ", colName, " has ", round(pctNull*100, 3), "% of nulls"))
  }
}
```

```
## [1] "Column  BuildingArea  has  60.576 % of nulls"
## [1] "Column  YearBuilt  has  55.386 % of nulls"
```

#Droping all the columns which are having more than 50 percent NA values

```
mel_house_data[,c("BuildingArea","YearBuilt")]<-NULL
```

Changing the type of the variables as per the need

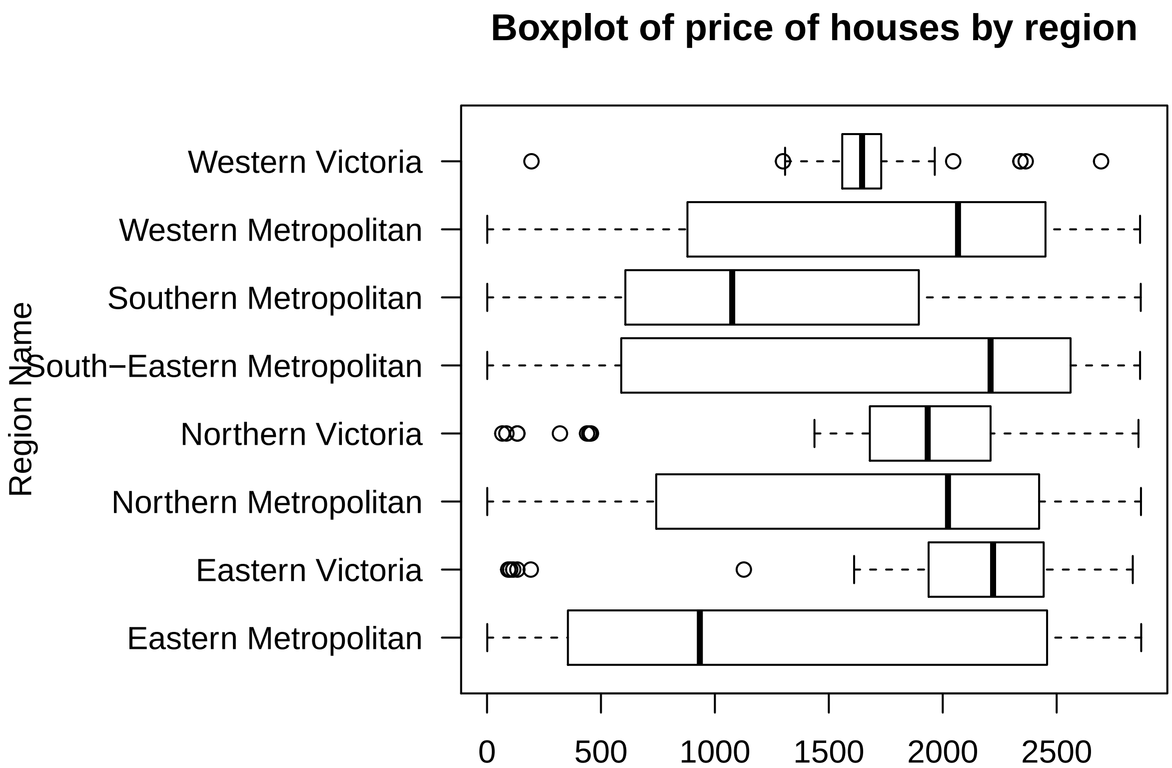
```r
mel_house_data_clean<-na.exclude(mel_house_data)

mel_house_data_clean$Type<-as.factor(mel_house_data_clean$Type)
mel_house_data_clean$Propertycount<-as.numeric(mel_house_data_clean$Propertycount)
mel_house_data_clean$Regionname<-as.factor(mel_house_data_clean$Regionname)
mel_house_data_clean$Distance<-as.numeric(mel_house_data_clean$Distance)
mel_house_data_clean$Price<-as.numeric(mel_house_data_clean$Price)
mel_house_data_clean$Landsize <- as.numeric(mel_house_data_clean$Landsize)
mel_house_data_clean$Car <- as.numeric(mel_house_data_clean$Car)
mel_house_data_clean$Bathroom <- as.numeric(mel_house_data_clean$Bathroom)
mel_house_data_clean$Rooms <- as.numeric(mel_house_data_clean$Rooms)


head(mel_house_data_clean)
```

```
##          Date Type Price Landsize Rooms Bathroom Car Distance
## 2  3/12/2016    h   589      554     5        2   2       81
## 3   4/2/2016    h    51      399     5        2   1       81
## 5   4/3/2017    h   574      287     6        4   1       81
## 6   4/3/2017    h  2612     1615     6        4   2       81
## 7   4/6/2016    h   692      189     7        2   7       81
## 11  7/5/2016    h  2782      495     5        2   1       81
##                Regionname Propertycount
## 2  Northern Metropolitan           190
## 3  Northern Metropolitan           190
## 5  Northern Metropolitan           190
## 6  Northern Metropolitan           190
## 7  Northern Metropolitan           190
## 11 Northern Metropolitan           190
```

#Boxplot to check the data and outliers

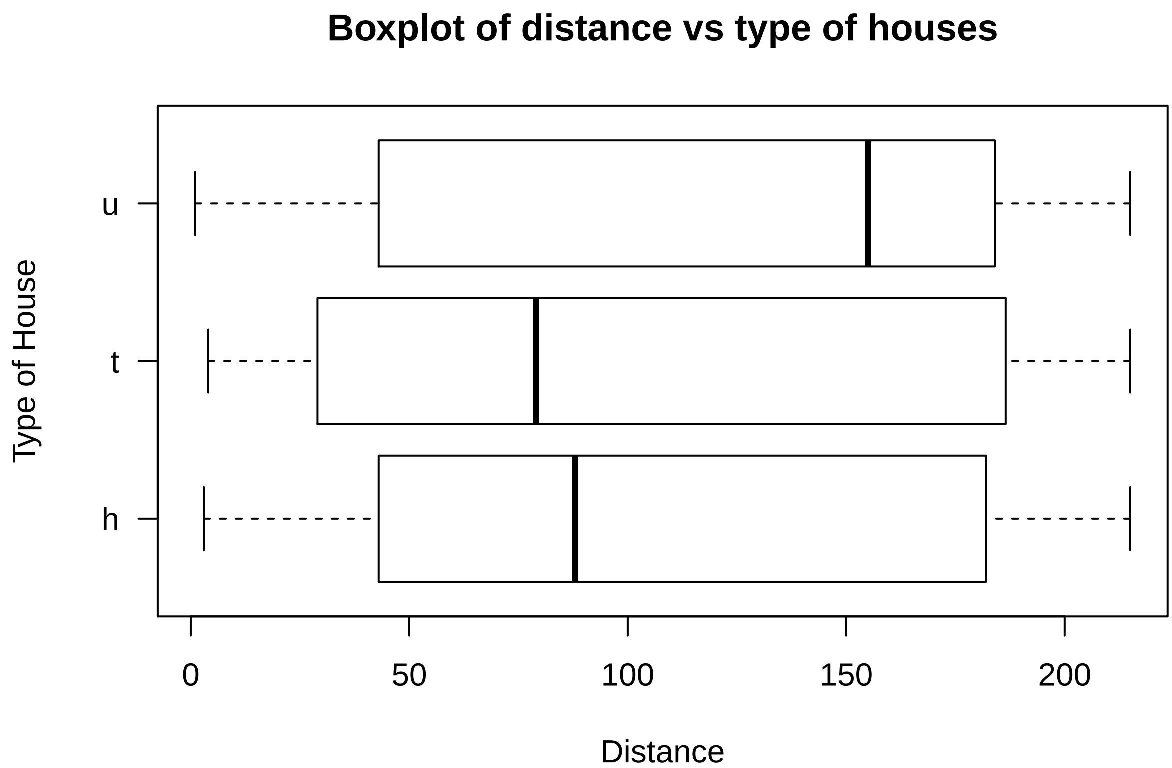
#making boxplot of price range in different regions

```r
par(mar=c(3.1,12,4.1,2.1), mgp = c(11, 1, 0))
boxplot(mel_house_data_clean$Price ~ mel_house_data_clean$Regionname,horizontal = TRUE,  ylab = "Region
```



#Boxplot of distance vs type of houses

```
boxplot(mel_house_data_clean$Distance ~ mel_house_data_clean$Type, horizontal = TRUE, ylab = "Type of Ho
```



#As we can see, there are few outliers in the price range by different region boxplot graph. but Currently we are not removing any because it will not affect our EDA part but yes we can see the impact of it when we do some modelling on the dataset.

#Step -2 ##Dataset has preprocessed So lets see the Statistics and summary of the Clean dataset

#columns in the clean dataset

```
colnames(mel_house_data_clean)
```

```
##  [1] "Date"          "Type"          "Price"         "Landsize"
##  [5] "Rooms"         "Bathroom"      "Car"           "Distance"
##  [9] "Regionname"    "Propertycount"
```

#structure of the clean dataset

```
str(mel_house_data_clean)
```

```
## 'data.frame':    17701 obs. of  10 variables:
##  $ Date         : Factor w/ 78 levels "1/7/2017","10/12/2016",..: 56 65 66 66 67 73 73 74 74 74 ...
##  $ Type         : Factor w/ 3 levels "h","t","u": 1 1 1 1 1 1 1 1 3 1 ...
##  $ Price        : num  589 51 574 2612 692 ...
##  $ Landsize     : num  554 399 287 1615 189 ...
##  $ Rooms        : num  5 5 6 6 7 5 6 5 1 5 ...
##  $ Bathroom     : num  2 2 4 4 2 2 4 2 2 2 ...
##  $ Car          : num  2 1 1 2 7 1 1 7 2 7 ...
```

```
##  $ Distance     : num  81 81 81 81 81 81 81 81 81 81 ...
##  $ Regionname   : Factor w/ 8 levels "Eastern Metropolitan",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Propertycount: num  190 190 190 190 190 190 190 190 190 190 ...
##  - attr(*, "na.action")= 'exclude' Named int  1 4 8 9 10 13 14 16 17 20 ...
##   ..- attr(*, "names")= chr  "1" "4" "8" "9" ...
```

#descriptive Summary of the clean dataset

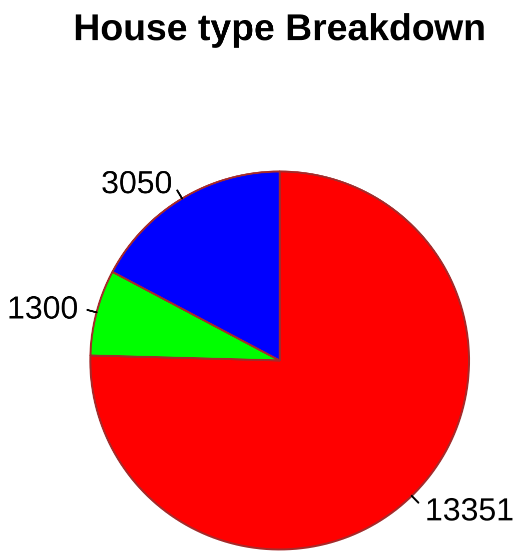
```
summary(mel_house_data_clean)
```

```
##          Date        Type          Price          Landsize
##  17/03/2018:  531   h:13351   Min.   :   1   Min.   :   1.0
##  24/02/2018:  489   t: 1300   1st Qu.: 609   1st Qu.: 573.0
##  27/05/2017:  473   u: 3050   Median :1726   Median :1050.0
##  3/3/2018  :  424             Mean   :1526   Mean   : 891.3
##  3/6/2017  :  397             3rd Qu.:2364   3rd Qu.:1268.0
##  12/8/2017 :  388             Max.   :2871   Max.   :1684.0
##  (Other)   :14999
##      Rooms           Bathroom          Car           Distance
##  Min.   : 1.000   Min.   : 1.00   Min.   : 1.00   Min.   :  1.0
##  1st Qu.: 5.000   1st Qu.: 2.00   1st Qu.: 2.00   1st Qu.: 41.0
##  Median : 6.000   Median : 2.00   Median : 7.00   Median : 91.0
##  Mean   : 5.935   Mean   : 3.07   Mean   : 5.08   Mean   :108.2
##  3rd Qu.: 7.000   3rd Qu.: 4.00   3rd Qu.: 7.00   3rd Qu.:183.0
##  Max.   :11.000   Max.   :11.00   Max.   :15.00   Max.   :215.0
##
##                      Regionname    Propertycount
##  Southern Metropolitan     :5530   Min.   :  1.0
##  Northern Metropolitan     :5063   1st Qu.: 63.0
##  Western Metropolitan      :3936   Median :202.0
##  Eastern Metropolitan      :2111   Mean   :177.2
##  South-Eastern Metropolitan: 789   3rd Qu.:272.0
##  Eastern Victoria          : 105   Max.   :342.0
##  (Other)                   : 167
```

#Lets make some plots

#pie chart

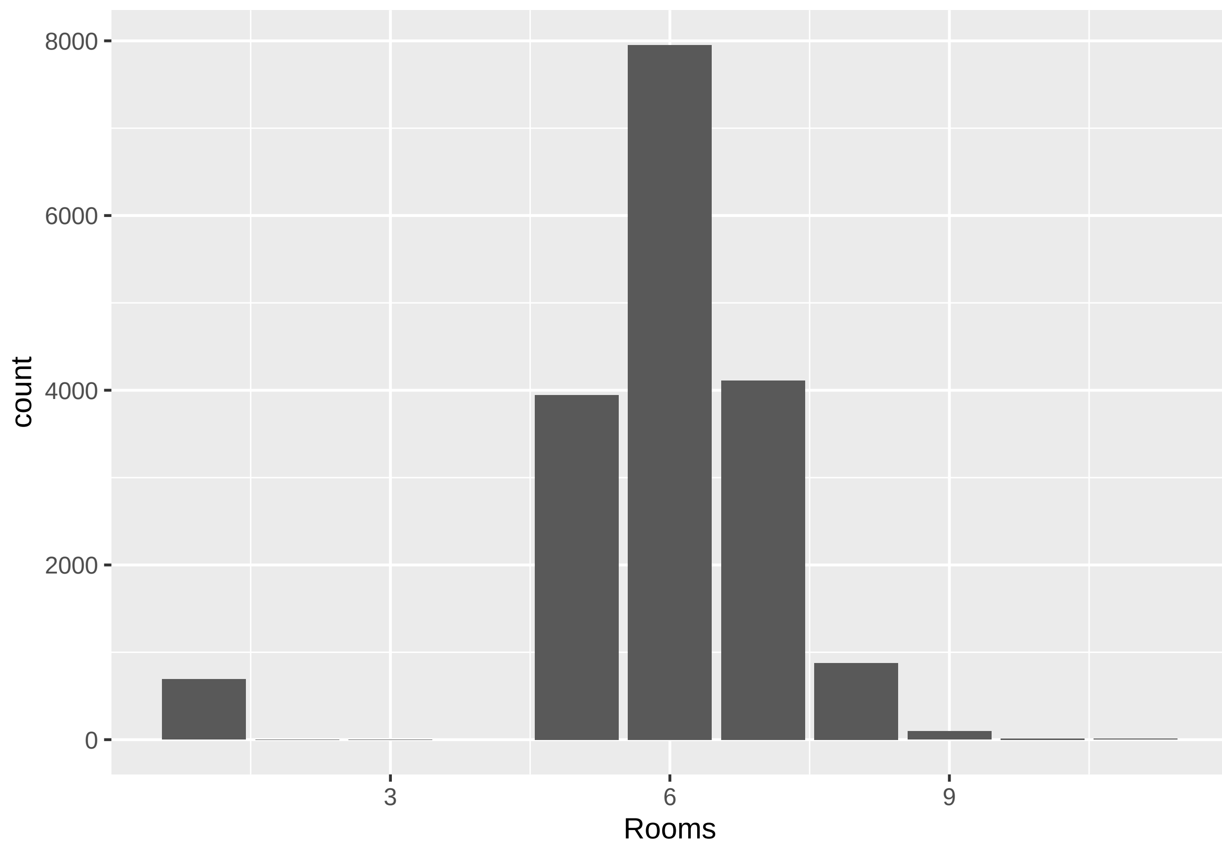
```
pie(table(mel_house_data_clean$Type),
    labels=table(mel_house_data_clean$Type),
    main="House type Breakdown",
    col=c("red","green","blue"),
    border="brown",
    clockwise=TRUE
)
```



#bar chart

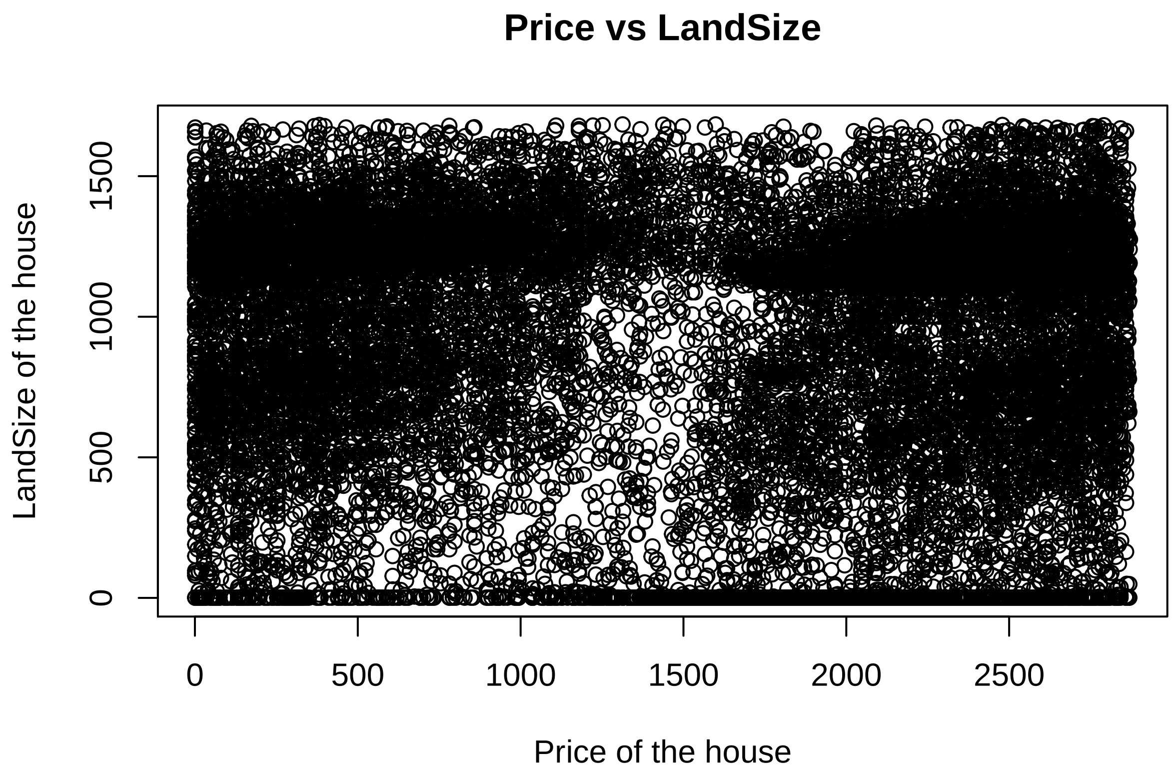
Distribution of Rooms in the houses:

```
ggplot(data =mel_house_data_clean) +
  geom_bar(mapping = aes(x = Rooms),position = "dodge")
```



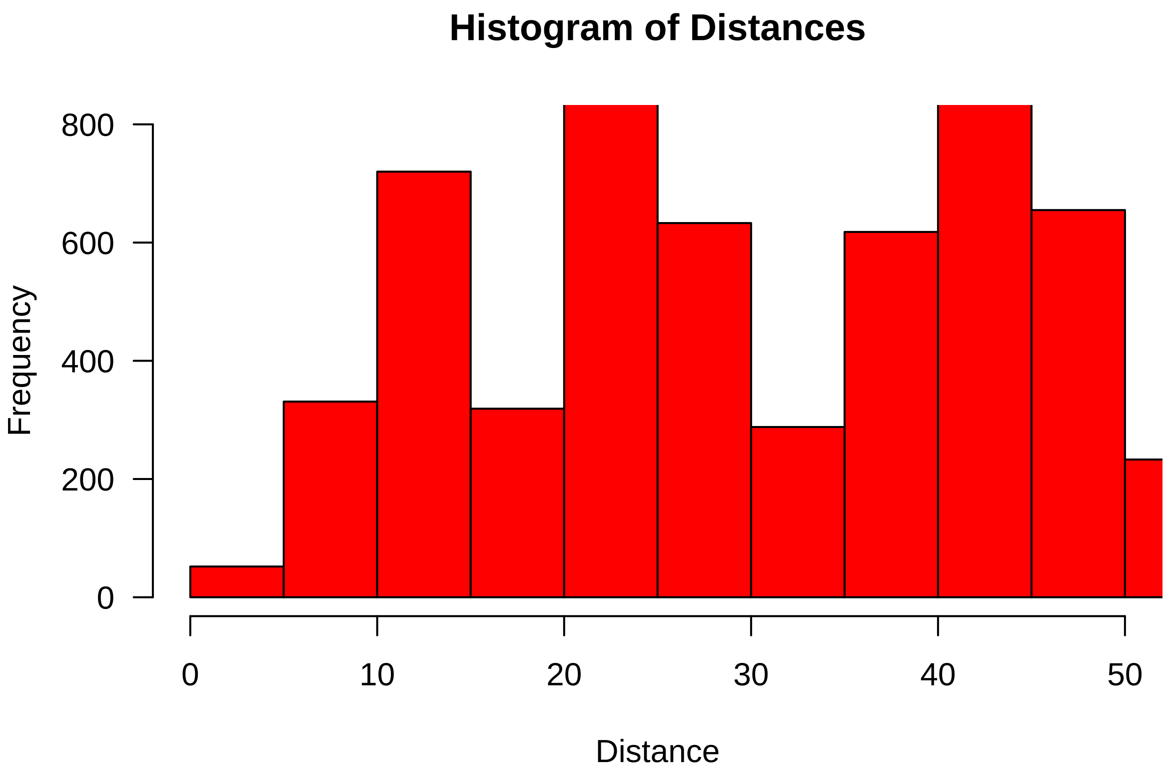
#Scatterplot

```
plot(x = mel_house_data_clean$Price,y = mel_house_data_clean$Landsize,
     xlab = "Price of the house",
     ylab = "LandSize of the house",
     main = "Price vs LandSize"
)
```

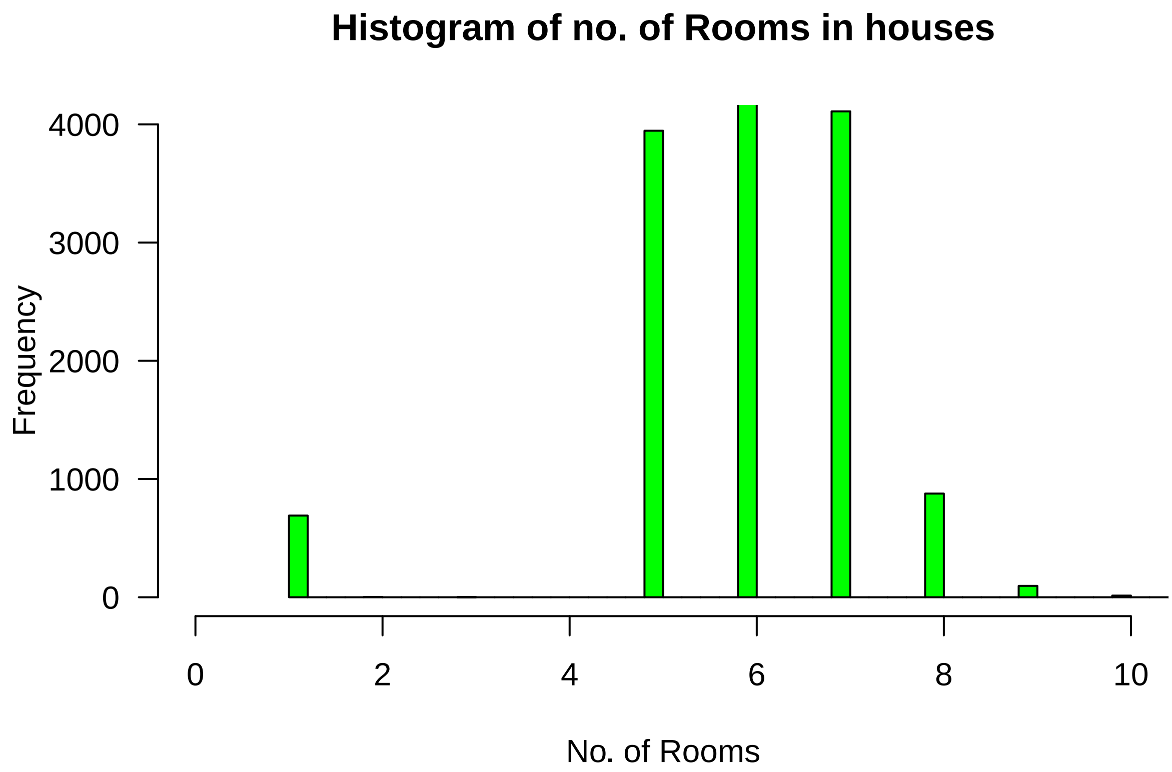


#Histograms #seeing the distribution of all the variables

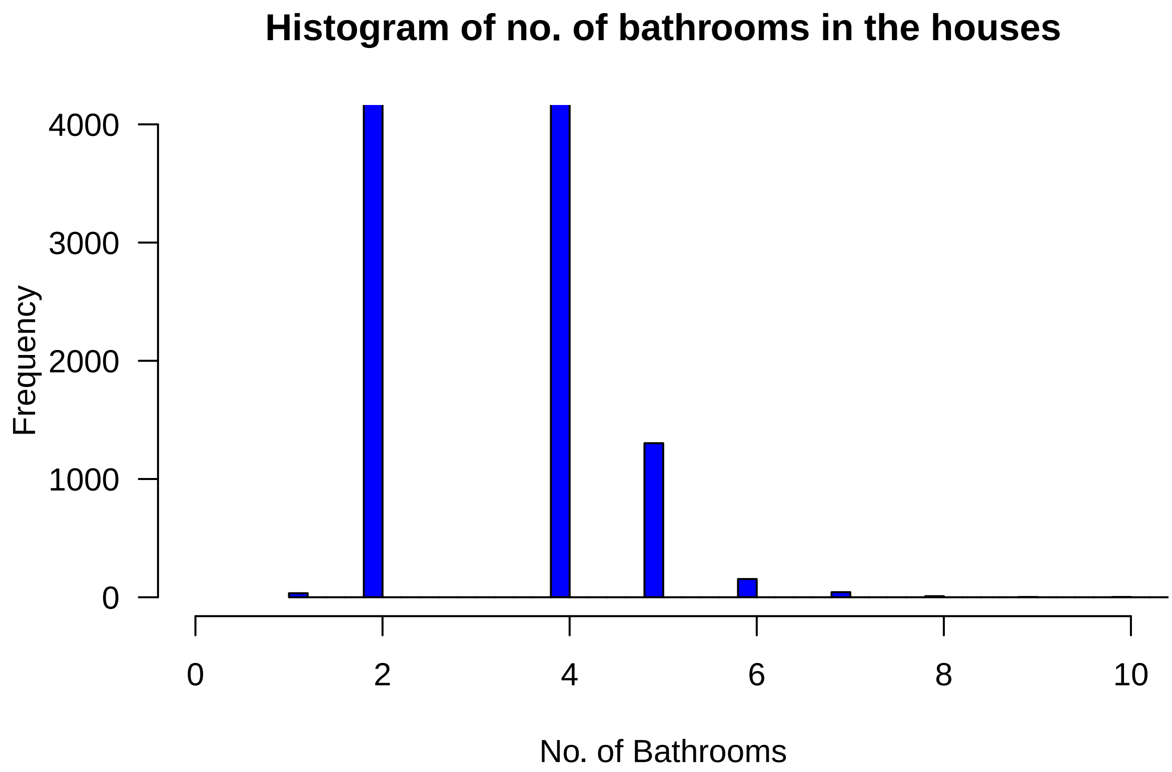
```
hist(mel_house_data_clean$Distance, breaks = 40, xlim = c(0,50), ylim = c(0,800),xlab = "Distance", col
```



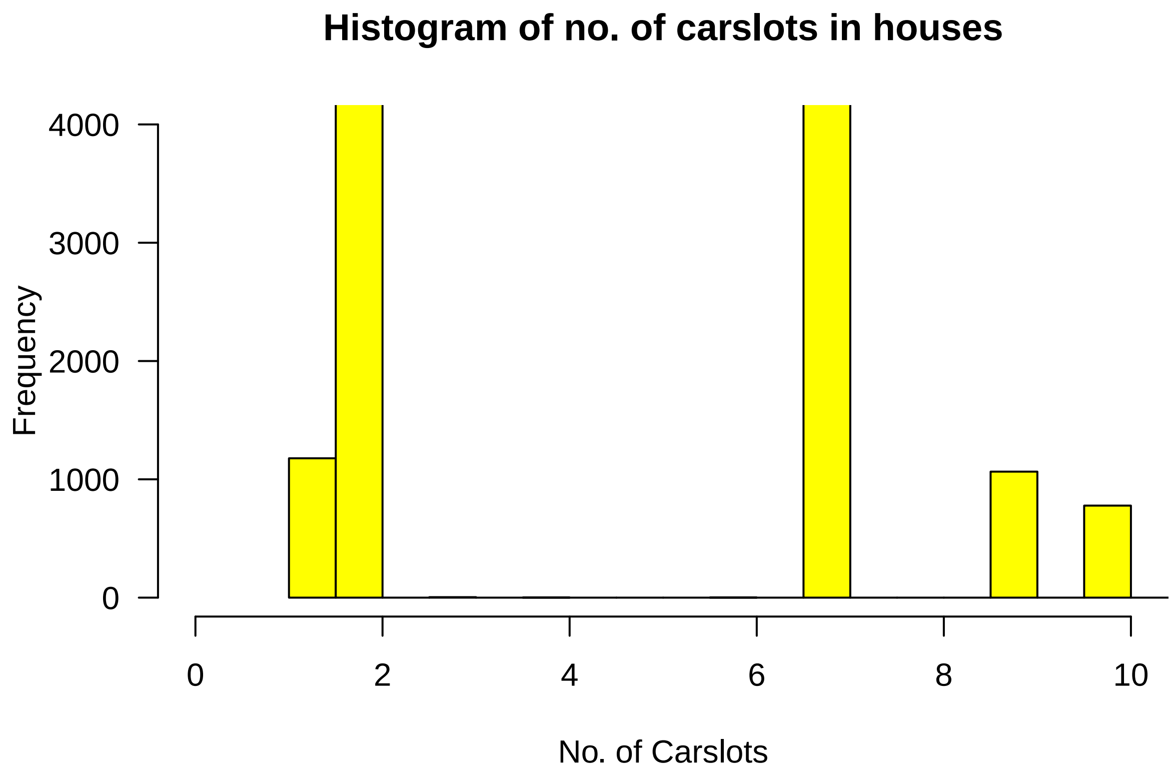

```
hist(mel_house_data_clean$Rooms, breaks = 40, xlim = c(0,10), ylim = c(0,4000),xlab = "No. of Rooms", c
```




```
hist(mel_house_data_clean$Bathroom, breaks = 40, xlim = c(0,10), ylim = c(0,4000),xlab = "No. of Bathro
```



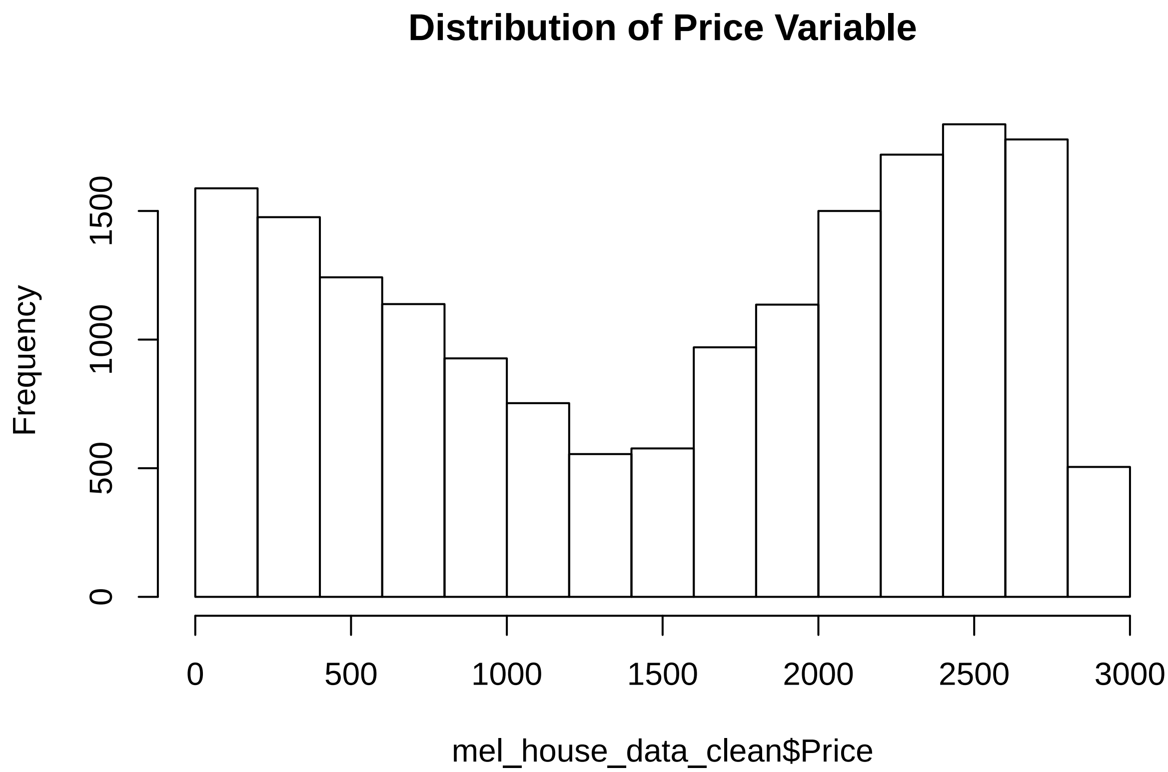

```
hist(mel_house_data_clean$Car, breaks = 40, xlim = c(0,10), ylim = c(0,4000),xlab = "No. of Carslots",
```



#Step-3

#Show the histogram of the price variable. Describe it briefy. Include summary statistics like mean,median, and variance.

```
hist(mel_house_data_clean$Price, main = "Distribution of Price Variable")
```



As we can see in the distribution of price variable, there are So many house at very low cost and very few house at very high cost.

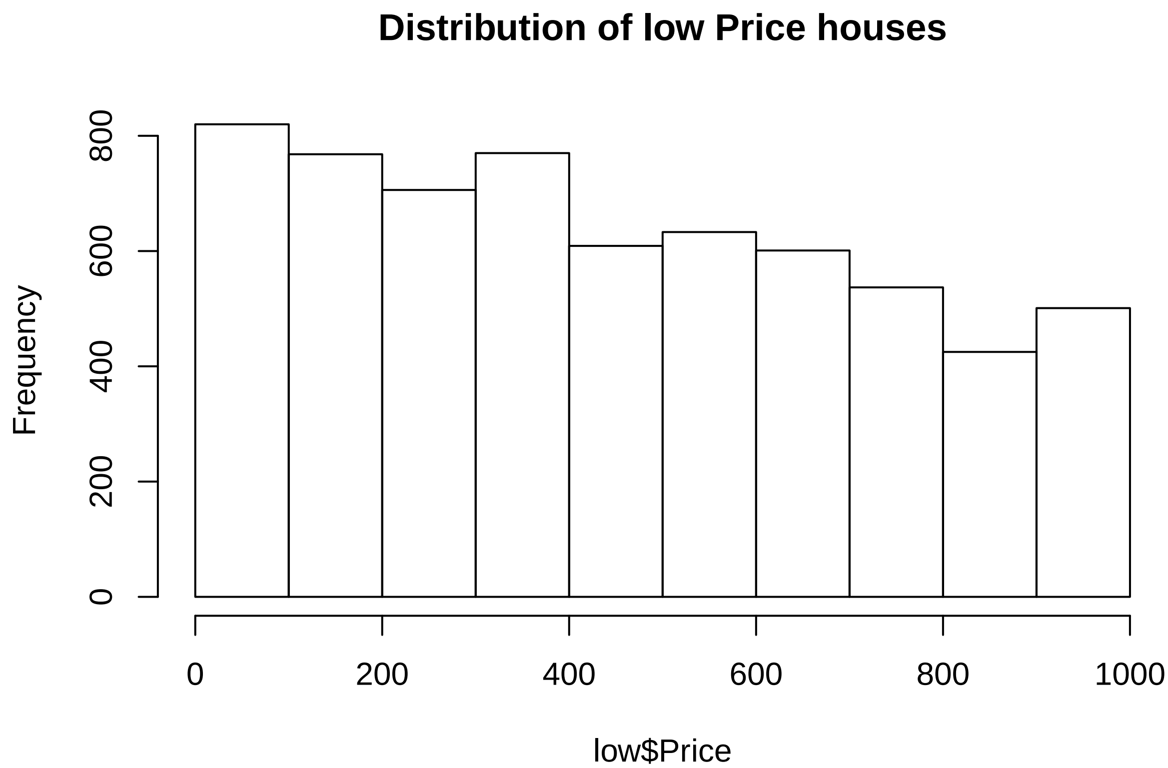
average of all the house price is 1526 and for first 25% house average price is 609.

#Min. :1.0
#1st Qu.:609
#Median :1726
#Mean :1526
#3rd Qu.:2364
#Max. :2871

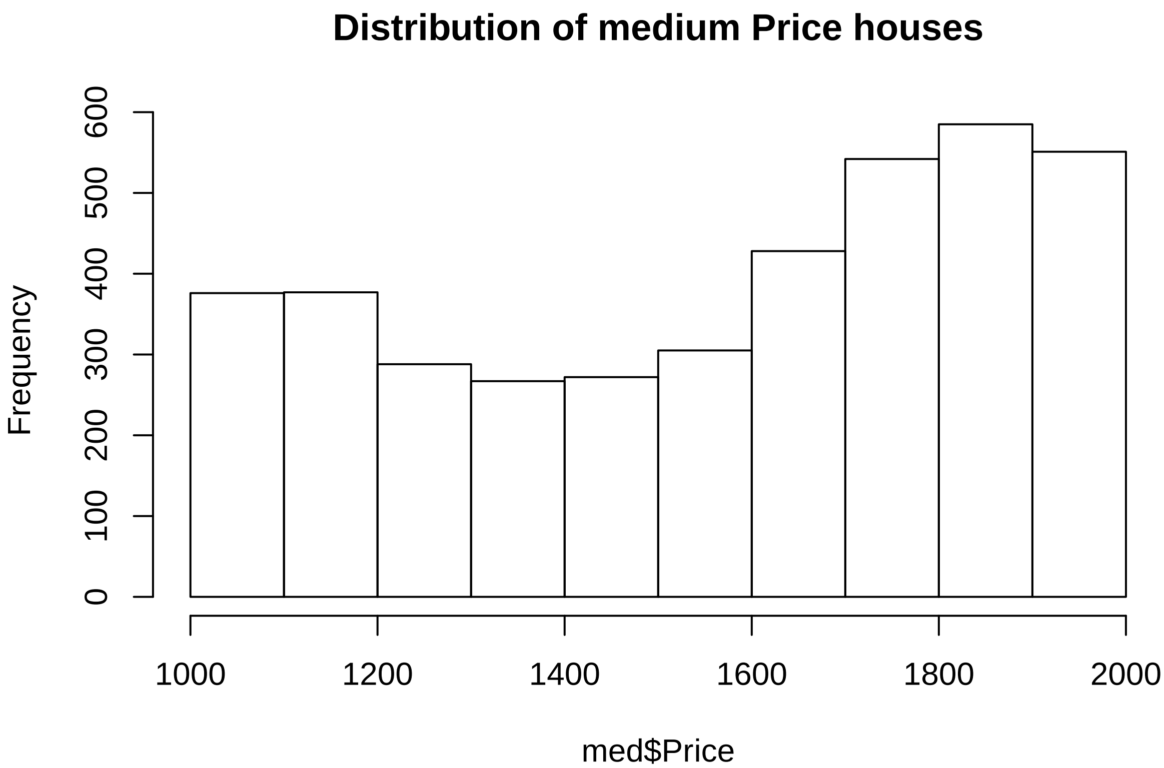
#We can divide house in different range like below 1000 - low cost, between 1000 to 2000 - medium cost, above 2000 - high cost house

#Group houses by some price ranges (like low, medium, high,etc.) and summarise those groups separately

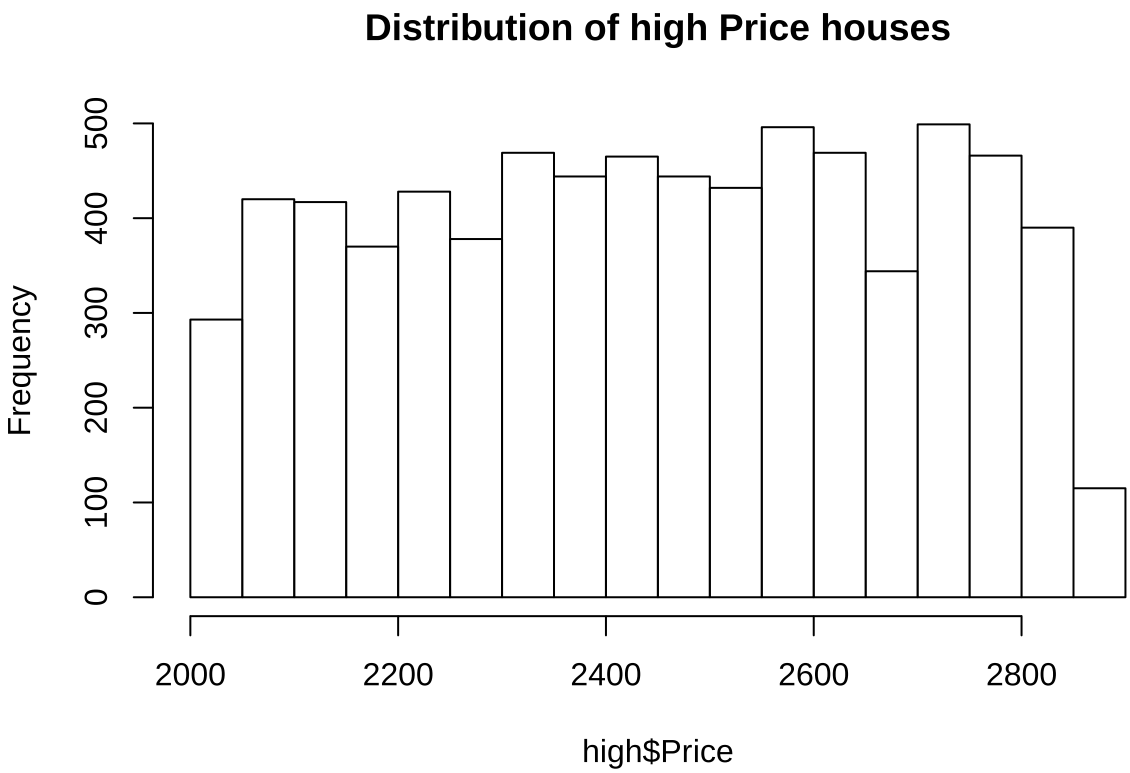
```
low <- mel_house_data_clean%>%filter(Price < 1000)
hist(low$Price, main = "Distribution of low Price houses")
```




```
med <- mel_house_data_clean%>%filter(Price > 1000 & Price < 2000)
hist(med$Price, main = "Distribution of medium Price houses")
```

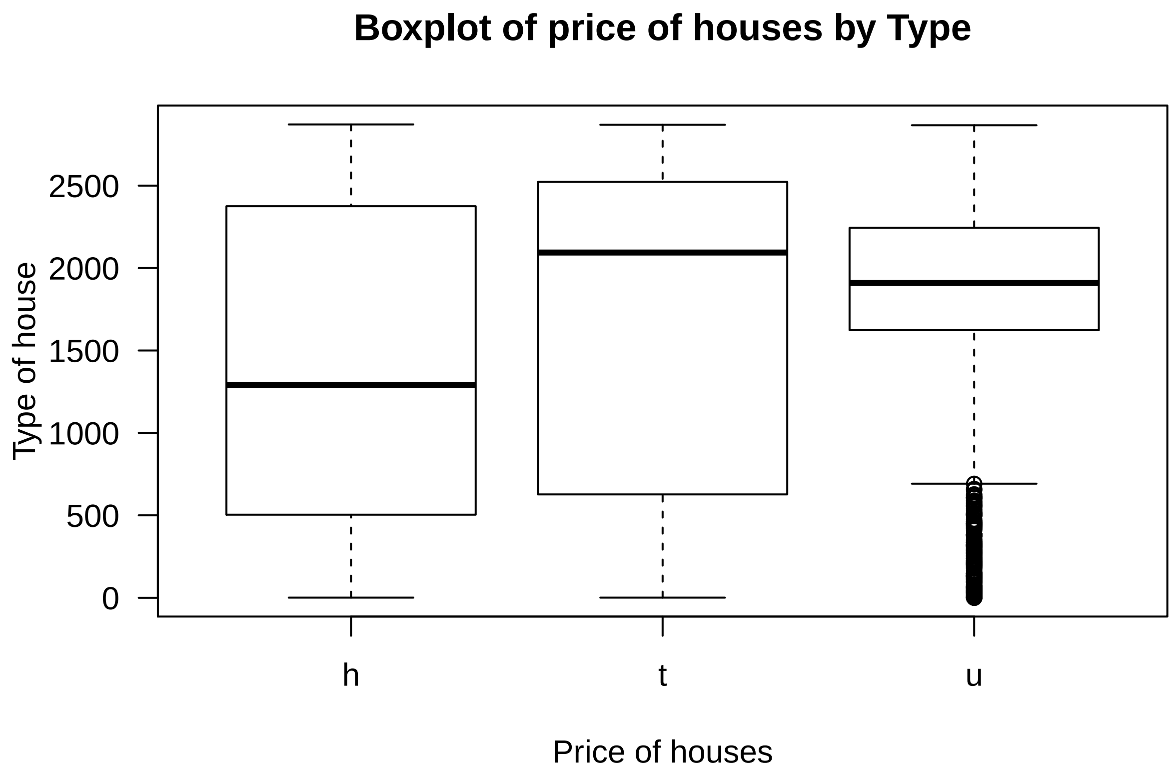



```
high <- mel_house_data_clean%>%filter(Price > 2000)
hist(high$Price, main = "Distribution of high Price houses")
```



#Explore prices for different house types. You might want to use the boxplot.

```
boxplot(mel_house_data_clean$Price ~ mel_house_data_clean$Type, horizontal = FALSE,  ylab = "Type of hou
```

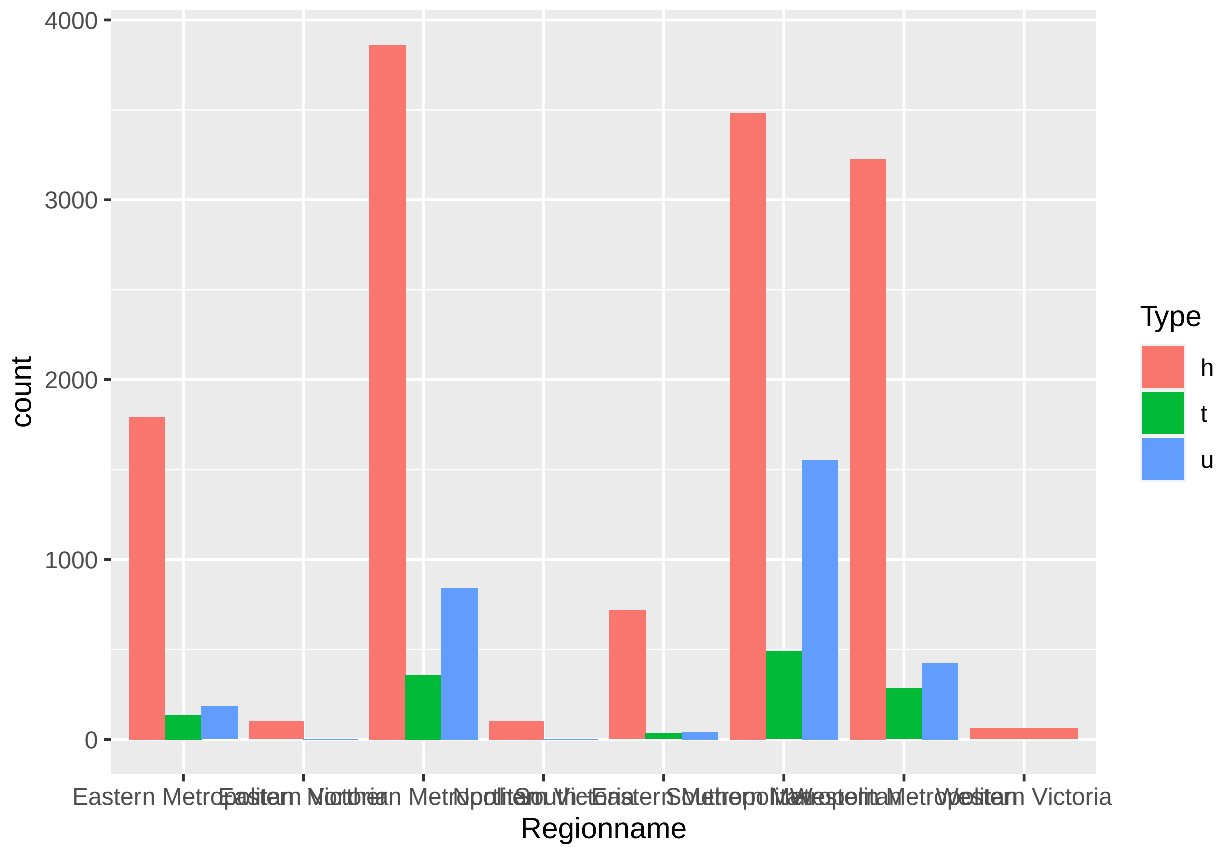


we can see that h type house are having around 1.4k price, t type houses are having more than average 2k price and u type having around 1.9k price.

#But there are so many outliers in the u type houses

Distribution of different Type of houses over the regions:

```
ggplot(data =mel_house_data_clean) +
  geom_bar(mapping = aes(x = Regionname, fill = Type),position = "dodge")
```



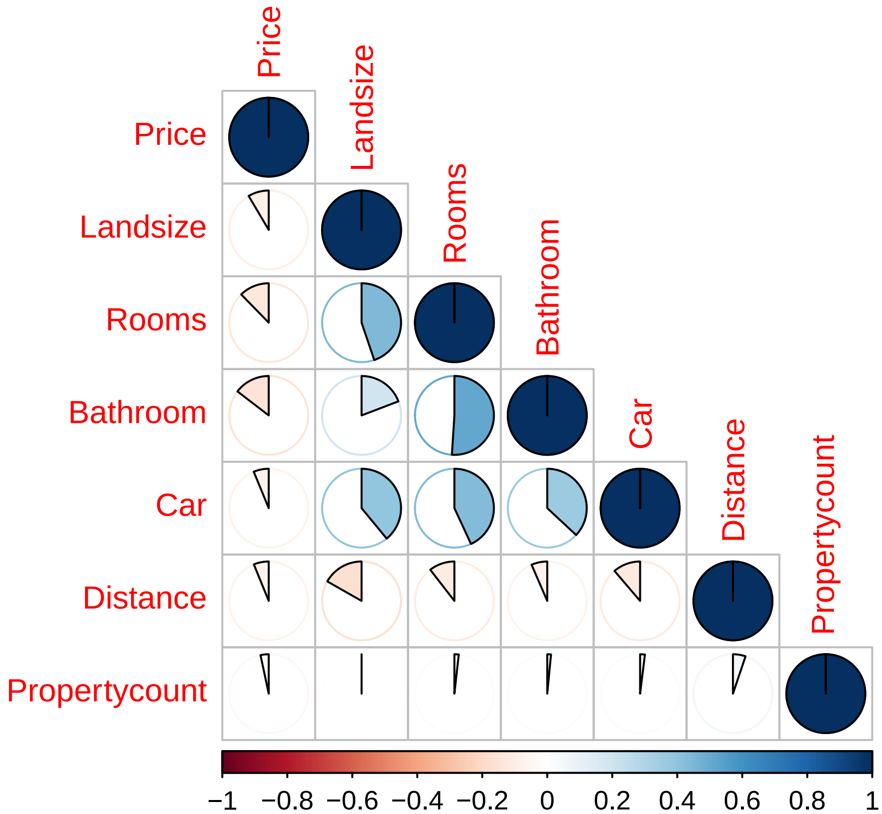
#How different attributes are correlated with the price? Which of the variables are correlated the most with price?

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.6.3
```

```
## corrplot 0.84 loaded
```

```
cor_data<- as.data.frame(mel_house_data_clean[,c(3:8, 10)])
corrplot(cor(as.matrix(cor_data)), method = "pie", type="lower")
```



price is correlated with the number of the bathroom in the house, LandSize of the House and number of the rooms in the house

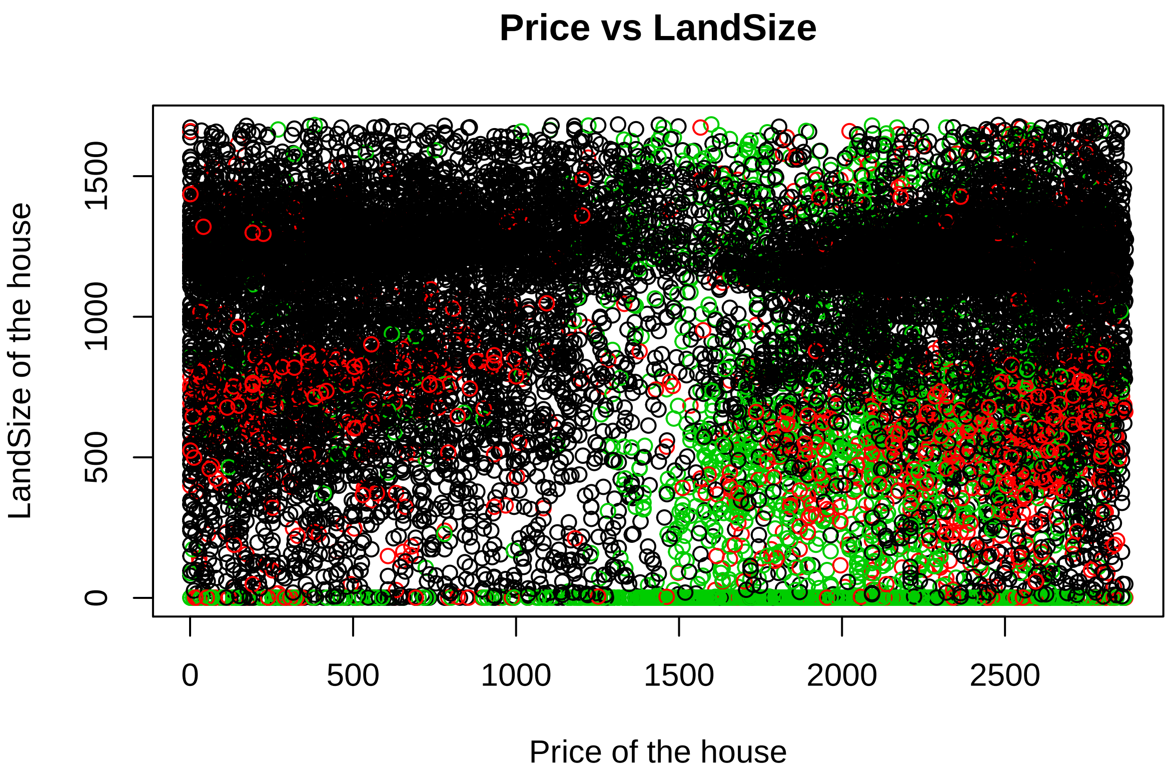
#step-4 #List the frequencies of houses for various types. Create 2 scatterplots and colour the house price by landsize and type.

```
table(mel_house_data_clean$Type)
```

```
##
##     h     t     u
## 13351  1300  3050
```

we can clearly see that we are having 13351 houses of h type, 1300 houses of t type and 3050 houses of u type.

```
plot(x = mel_house_data_clean$Price,y = mel_house_data_clean$Landsize,
     xlab = "Price of the house",
     ylab = "LandSize of the house",
     main = "Price vs LandSize",
     col = mel_house_data_clean$Type
)
```



#Scatter plot between price and LandSize and by type we can see the color as the type.