Big Data Analysis Assignment

Rohin Mehra Student Number: 3082862

2022-10-27

Contents

MISSION STATEMENT

In this assignment we will perform exploratory data analysis and calculate the strength of relationships between the variables of the data set. The housing data set contains the prices and other attributes of almost 35,000 houses in the city of Melbourne.

Reading the data set and storing it into a data set so that we can view structure of our data frame and perform exploratory data analysis.

So we now have read our CSV file and

```
housing.dataset<-read.csv(file = "/Users/rohinmehra/Downloads/melbourne_data.csv",header = TRUE)
```

Libraries that are being involved for both analysis

1. GGPlot (Ggplot is a package in R by tidyverse. It is based on Leland Wilkinson's Grammar of Graphics . ggplot creates complex and intricate plots using the principles listed in the grammar of graphics. Users can use all types of data, such as uni variate, multivariate, or categorical, to create data)
2. Tidy Verse (The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures. The core tidyverse includes the packages that you're likely to use in everyday data analyses.)
3. Dplyr (dplyr is an R package for working with structured data both in and outside of R. dplyr makes data manipulation for R users easy, consistent, and performance. With dplyr as an interface to manipulating Spark Data Frames, you can: Select, filter, and aggregate data)

```
library(ggplot2)
library(tidyverse)
require("RColorBrewer")
library(dplyr)
```

Data Set Structure and it's Associations

We will see now how does our original data is structured this will include detail's such as what all column's are present and what information they hold in association to our data set.

```
str(housing.dataset)
```

```
## 'data.frame':    34857 obs. of  13 variables:
##  $ X            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Date         : chr  "3/09/2016" "3/12/2016" "4/02/2016" "4/02/2016" ...
##  $ 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" "North
##  $ Propertycount: chr  "4019" "4019" "4019" "4019" ...
```

```
head(housing.dataset)
```

```
##   X      Date Type   Price Landsize BuildingArea Rooms Bathroom Car YearBuilt
## 1 1 3/09/2016    h      NA      126           NA     2        1   1        NA
## 2 2 3/12/2016    h 1480000      202           NA     2        1   1        NA
## 3 3 4/02/2016    h 1035000      156           79     2        1   0      1900
## 4 4 4/02/2016    u      NA        0           NA     3        2   1        NA
## 5 5 4/03/2017    h 1465000      134          150     3        2   0      1900
## 6 6 4/03/2017    h  850000       94           NA     3        2   1        NA
##   Distance            Regionname Propertycount
## 1      2.5 Northern Metropolitan          4019
## 2      2.5 Northern Metropolitan          4019
## 3      2.5 Northern Metropolitan          4019
## 4      2.5 Northern Metropolitan          4019
## 5      2.5 Northern Metropolitan          4019
## 6      2.5 Northern Metropolitan          4019
```

Now we know what all column's are present in our CSV file and also know that this data is not clean so we need cleaning .

Data Set Cleaning

Next step is we clean data by replacing "NA" values with the figure of "0" .we make a data frame of our data set and store it in object " t " , then we detect all NA values in our data frame and replace them with 0 because we don't want to omit and row while cleaning .

Final result is stored in object " f " this is our clean data frame free of NA values.

```
t<-data.frame(housing.dataset)
```

1. **Unclean data frame :**

```
head(t)
```

```
##   X      Date Type   Price Landsize BuildingArea Rooms Bathroom Car YearBuilt
## 1 1 3/09/2016    h      NA      126           NA     2        1   1        NA
## 2 2 3/12/2016    h 1480000      202           NA     2        1   1        NA
## 3 3 4/02/2016    h 1035000      156           79     2        1   0      1900
## 4 4 4/02/2016    u      NA        0           NA     3        2   1        NA
## 5 5 4/03/2017    h 1465000      134          150     3        2   0      1900
## 6 6 4/03/2017    h  850000       94           NA     3        2   1        NA
##   Distance            Regionname Propertycount
## 1      2.5 Northern Metropolitan          4019
## 2      2.5 Northern Metropolitan          4019
## 3      2.5 Northern Metropolitan          4019
## 4      2.5 Northern Metropolitan          4019
## 5      2.5 Northern Metropolitan          4019
## 6      2.5 Northern Metropolitan          4019
```

2. **Partial clean data frame, Here NA values are replaced by figure of "0" :**

```
f<-replace(t ,is.na(t), 0)
head(f)
```

```
##   X      Date Type   Price Landsize BuildingArea Rooms Bathroom Car YearBuilt
## 1 1 3/09/2016    h       0      126            0     2        1   1         0
## 2 2 3/12/2016    h 1480000      202            0     2        1   1         0
## 3 3 4/02/2016    h 1035000      156           79     2        1   0      1900
## 4 4 4/02/2016    u       0        0            0     3        2   1         0
## 5 5 4/03/2017    h 1465000      134          150     3        2   0      1900
## 6 6 4/03/2017    h  850000       94            0     3        2   1         0
##   Distance            Regionname Propertycount
## 1      2.5 Northern Metropolitan          4019
## 2      2.5 Northern Metropolitan          4019
## 3      2.5 Northern Metropolitan          4019
## 4      2.5 Northern Metropolitan          4019
## 5      2.5 Northern Metropolitan          4019
## 6      2.5 Northern Metropolitan          4019
```

3. **Final cleaning step now we will eliminate row's which have Building Area equal to 0 , Price equal to 0 ,land size not equal to 0 as it will affect our plotting and exploratory analysis. Finally our data frame is cleaned of missing values , NA values and is ready for analysis.**

```
f<-subset(f,f$Price != 0 & f$BuildingArea != 0 & f$Landsize != 0)
head(f)
```

```
##     X      Date Type   Price Landsize BuildingArea Rooms Bathroom Car YearBuilt
```

```
## 3   3 4/02/2016    h 1035000      156           79     2        1   0      1900
## 5   5 4/03/2017    h 1465000      134          150     3        2   0      1900
## 7   7 4/06/2016    h 1600000      120          142     4        1   2      2014
## 12 12 7/05/2016    h 1876000      245          210     3        2   0      1910
## 15 15 8/10/2016    h 1636000      256          107     2        1   2      1890
## 19 19 8/10/2016    h 1097000      220           75     2        1   2      1900
##    Distance            Regionname Propertycount
## 3       2.5 Northern Metropolitan          4019
## 5       2.5 Northern Metropolitan          4019
## 7       2.5 Northern Metropolitan          4019
## 12      2.5 Northern Metropolitan          4019
## 15      2.5 Northern Metropolitan          4019
## 19      2.5 Northern Metropolitan          4019
```

```
str(f)
```

```
## 'data.frame':    8272 obs. of  13 variables:
##  $ X            : int  3 5 7 12 15 19 25 31 33 36 ...
##  $ Date         : chr  "4/02/2016" "4/03/2017" "4/06/2016" "7/05/2016" ...
##  $ Type         : chr  "h" "h" "h" "h" ...
##  $ Price        : num  1035000 1465000 1600000 1876000 1636000 ...
##  $ Landsize     : num  156 134 120 245 256 220 214 238 113 138 ...
##  $ BuildingArea : num  79 150 142 210 107 75 190 97 110 105 ...
##  $ Rooms        : int  2 3 4 3 2 2 3 2 3 3 ...
##  $ Bathroom     : num  1 2 1 2 1 1 2 1 2 1 ...
##  $ Car          : num  0 0 2 0 2 2 2 2 1 1 ...
##  $ YearBuilt    : num  1900 1900 2014 1910 1890 ...
##  $ Distance     : chr  "2.5" "2.5" "2.5" "2.5" ...
##  $ Regionname   : chr  "Northern Metropolitan" "Northern Metropolitan" "Northern Metropolitan" "Nortl
##  $ Propertycount: chr  "4019" "4019" "4019" "4019" ...
```

Start Of Analysis

1. **Scatter Plot**

   Definition : A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.

   Use in Data Frame : Now we will plot a scatter plot to show a relation between the price of the houses and land area so that we can observe how price of a house changes with land size also we can also observe that the color's of the point's is based on the type of the house in our data frame.

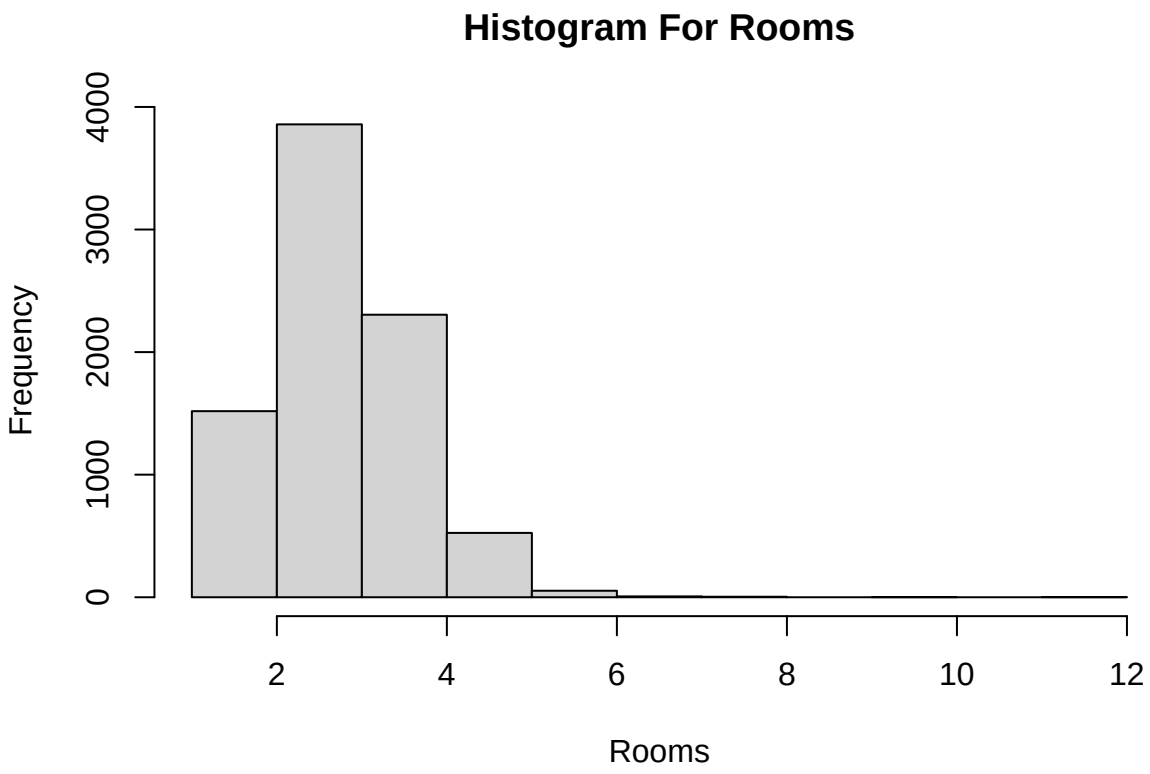
```
ggplot(f,aes(x = Price , y = Landsize,size = BuildingArea  , colour = Type))+ geom_point() + ggtitl
```

2. **Histogram Plot**

   Definition : A histogram is a graphical representation of data points organised into user-specified ranges. Similar in appearance to a bar graph, the histogram condenses a data series into an easily interpreted visual by taking many data points and grouping them into logical ranges or bins.

   Use in Data Frame : Now we will plot a Histogram to show a relation between the number of rooms in a houses and Number of houses with similar rooms, so that we can observe how frequency of houses changes with quantity of rooms.

```
hist(f$Rooms, main = "Histogram For Rooms", xlab = "Rooms")
```

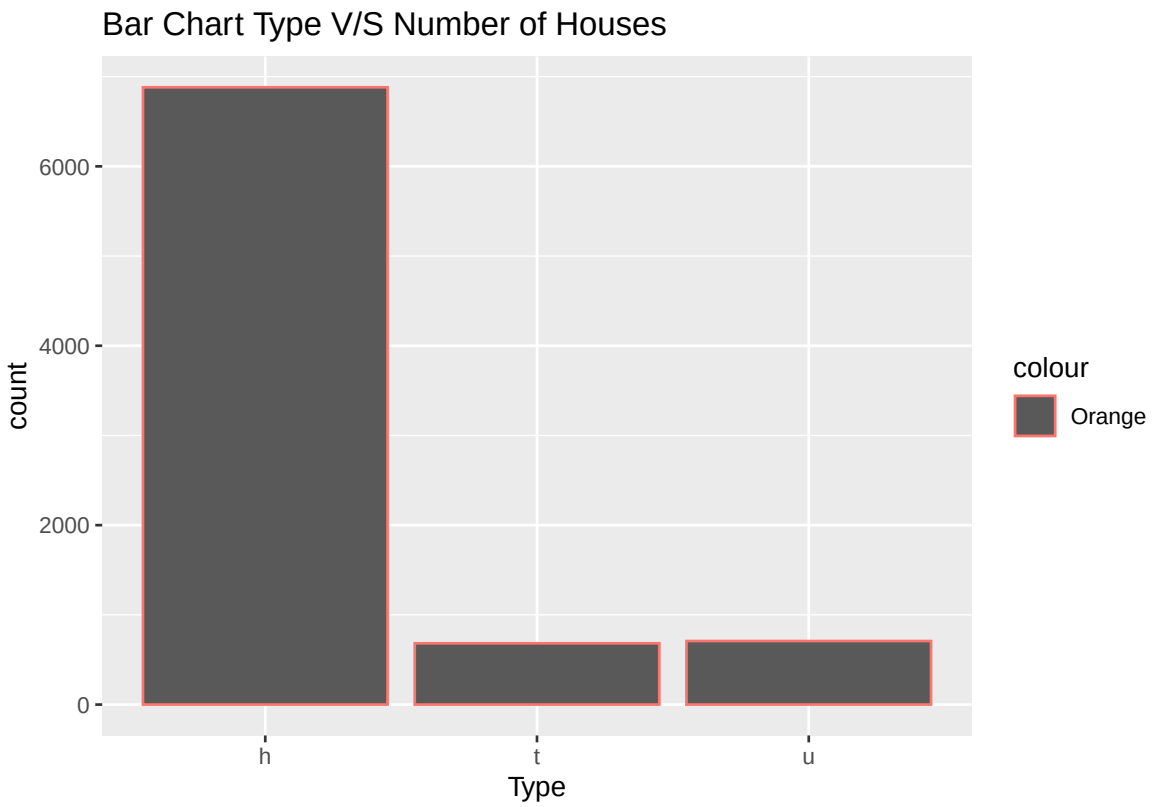


3. **Bar Chart**

   Definition : A bar chart (or graph) organises information into a graphic using bars of different lengths. The length of these bars is proportional to the size of the information they represent. To read a bar chart, consider the length of the bar connected to each category to find its value.Bar charts organise categorical data, whereas histograms organise numerical data.

   Use in Data Frame : Now we will plot a Bar Chart to show a relation between the number of the houses and house type so that we can observe how number of the houses change with house type.

```
ggplot(f,aes(x = Type , color = "Orange")) + geom_bar()+ labs(title = "Bar Chart Type V/S Number of
```

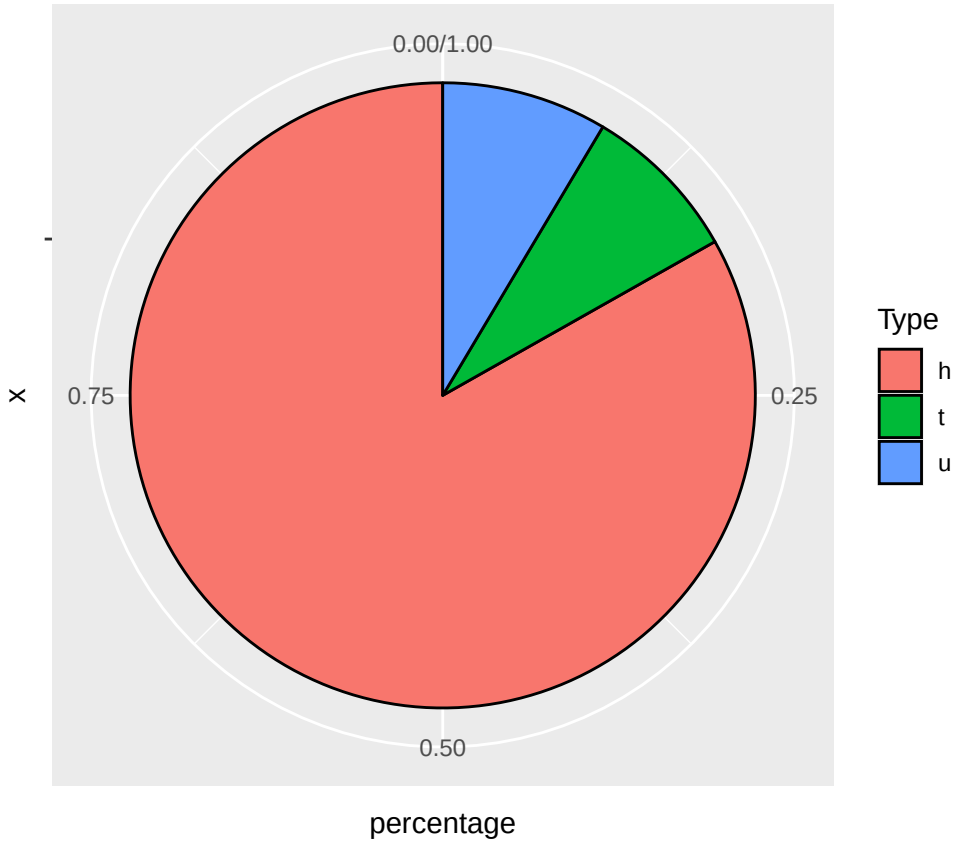


4. **Pie Chart**

   Definition : A pie chart (or a circle chart) is a circular statistical graphic, which is divided into slices to illustrate numerical proportions. In a pie chart, the arc length of each slice (and consequently its central angle and area) is proportional to the quantity it represents. While it is named for its resemblance to a pie which has been sliced, there are variations in the way it can be presented. The earliest known pie chart is generally credited to William Playfair's Statistical Breviary of 1801.

   Use in Data Frame : Now we will plot a Pie Chart to show a relation between the number of the houses and house type so that we can observe what percentage of houses are in which type of houses.

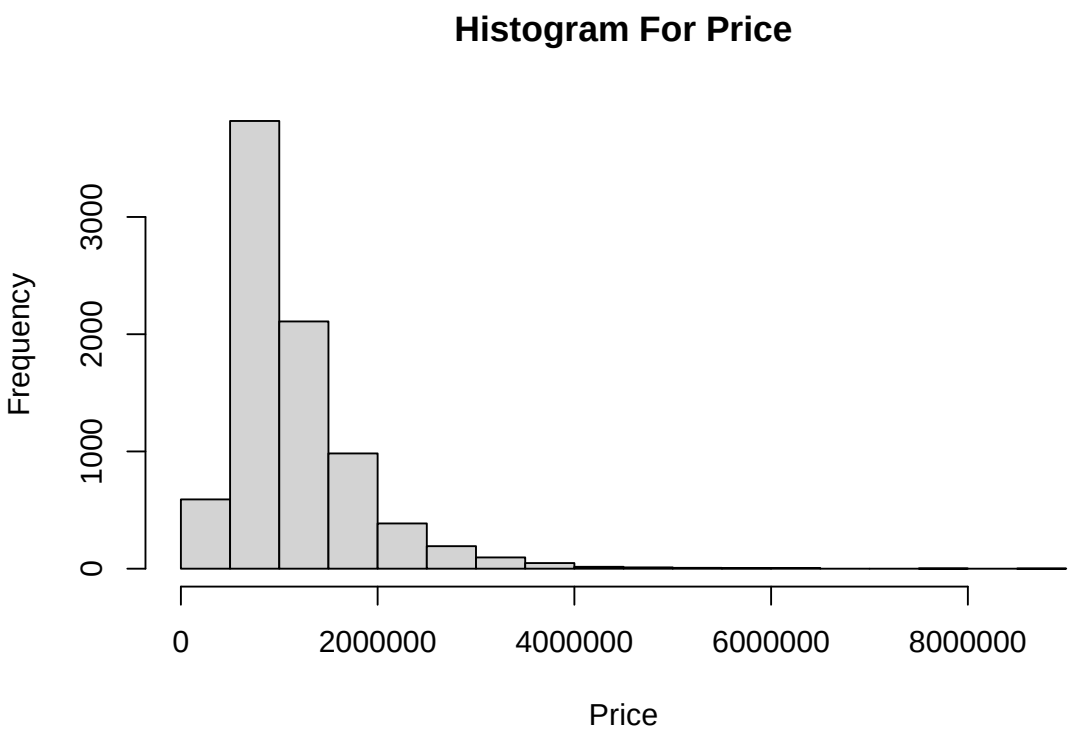
```
PieByType<- f %>% group_by(Type) %>% summarise(counts = n(),
                                               percentage = n()/nrow(f))
ggplot(data = PieByType,aes(x = "", y = percentage, fill=Type))+geom_col(colour = "Black")+coord_polar(
```



Main Focus

1. Histogram of Price Variable with summary

```
options(scipen = 999)
hist(f$Price, main = "Histogram For Price", xlab = "Price")
```



summary of Price variable where we can observe Minimum Price of a house , Maximum Price of a house , Mean Price of houses , Median Price of houses and Variance which is "471970520767" of houses present in our data frame.

Also "options(scipen = 999)" render's r studio compiler not to show hexadecimal values on graphs we will plot next .

```
summary(f$Price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  131000  700000  960000 1154030 1401000 9000000
```

```
var(f$Price)
```

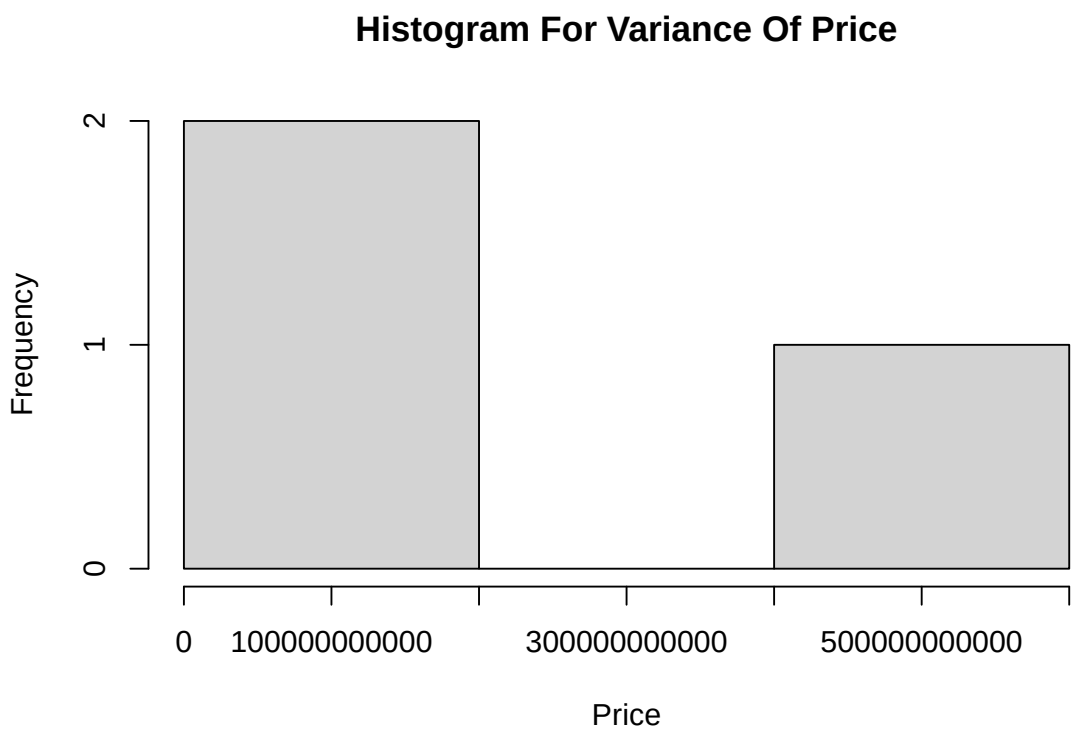
```
## [1] 471970520767
```

1. Histogram For Median Of Price (Here we group prices with type of houses take their sum and then take median out.)

```
Medianby_Type<-f %>% group_by(Type) %>% summarise( Count=n(),PriceSum = sum(Price),Median_Price = m
hist(Medianby_Type$Median_Price, main = "Histogram For Median Of Price", xlab = "Price")
```

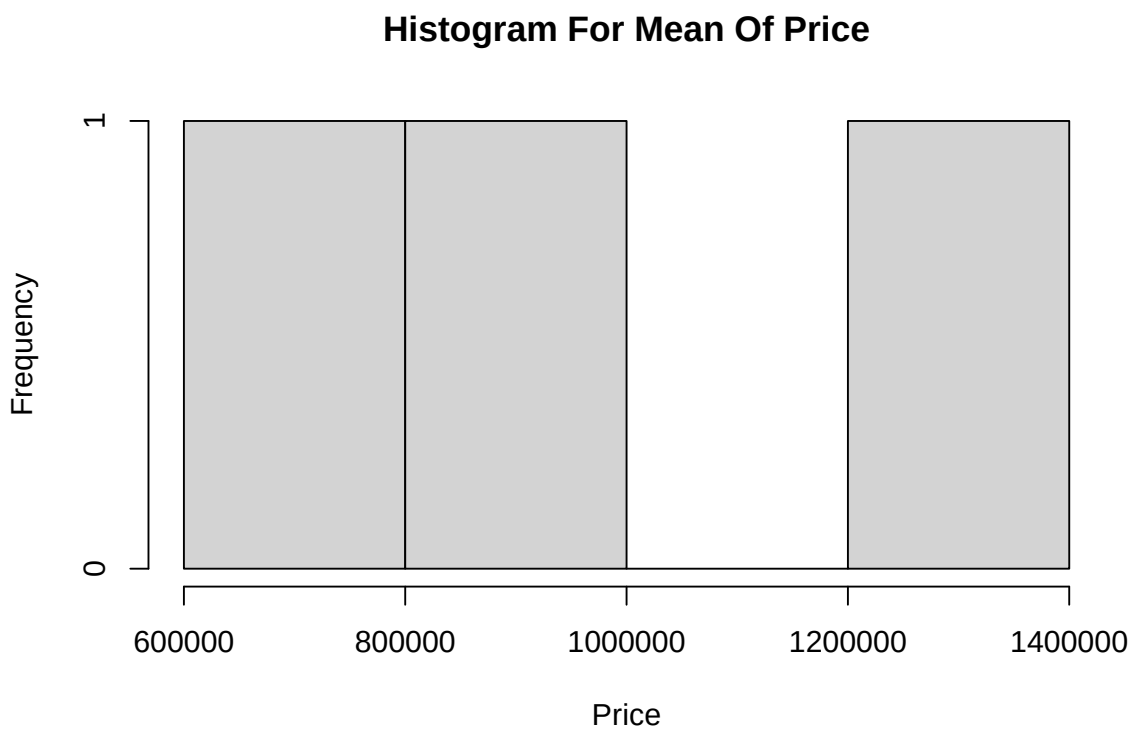
2. Histogram For Variance Of Price (Here we group prices with type of houses take their sum and then take variance out.)

```
options(scipen = 999)
Varianceby_Type<-f %>% group_by(Type) %>% summarise( Count=n(),PriceSum = sum(Price),Variance_Price
hist(Varianceby_Type$Variance_Price, main = "Histogram For Variance Of Price", xlab = "Price")
```



3. Histogram For Mean Of Price (Here we group prices with type of houses take their sum and then take mean out.)

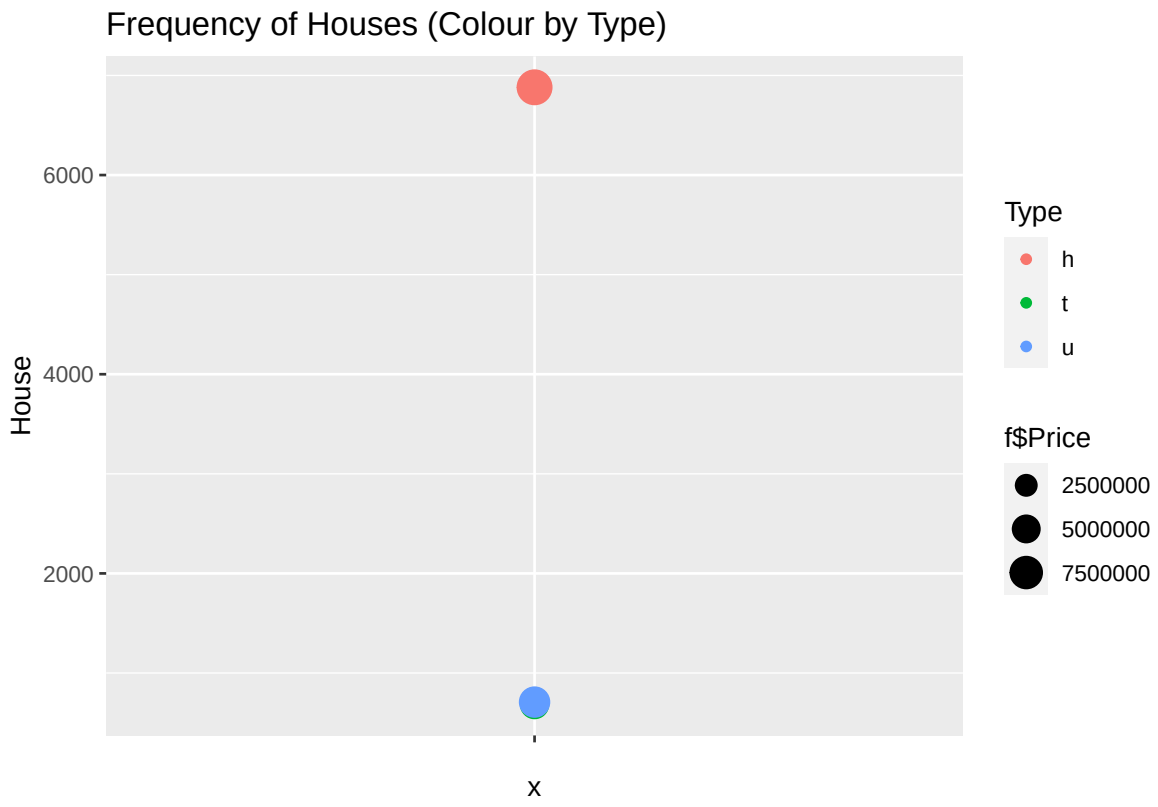
```
Meanby_Type<-f %>% group_by(Type) %>% summarise( Count=n(),PriceSum = sum(Price),Mean_Price = mean(
hist(Meanby_Type$Mean_Price, main = "Histogram For Mean Of Price", xlab = "Price")
```



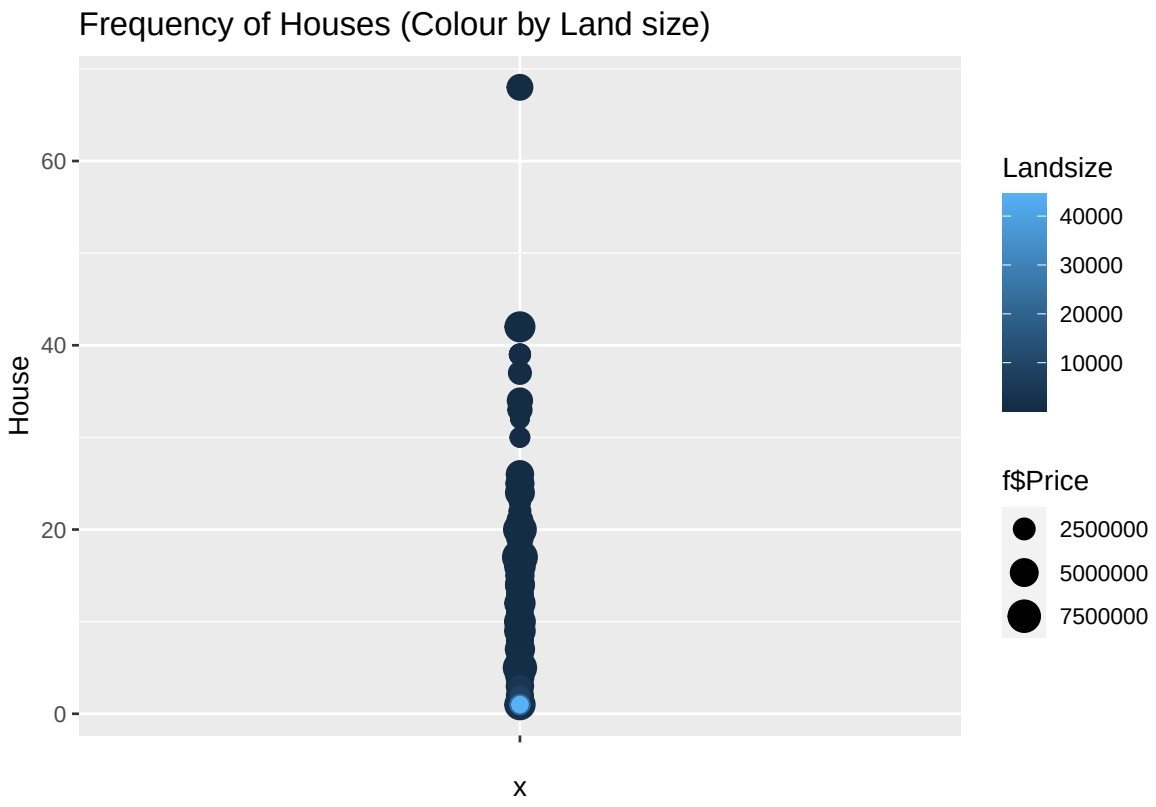
4. We will now group houses by some price ranges like low, medium, high, etc. and plot them separately .

5. Now we list the frequencies of houses for various types. Create 2 scatter plots and color the house price by land size and type.

```
Frequency_House<- f %>%  group_by(Type) %>%
  summarise(House = n(), countSum = sum(as.integer(Propertycount)), Mean_count = mean(Propertycount
ggplot(Frequency_House,aes(x = "", y = House ,size = f$Price  , colour = Type))+ geom_point() + ggt
```



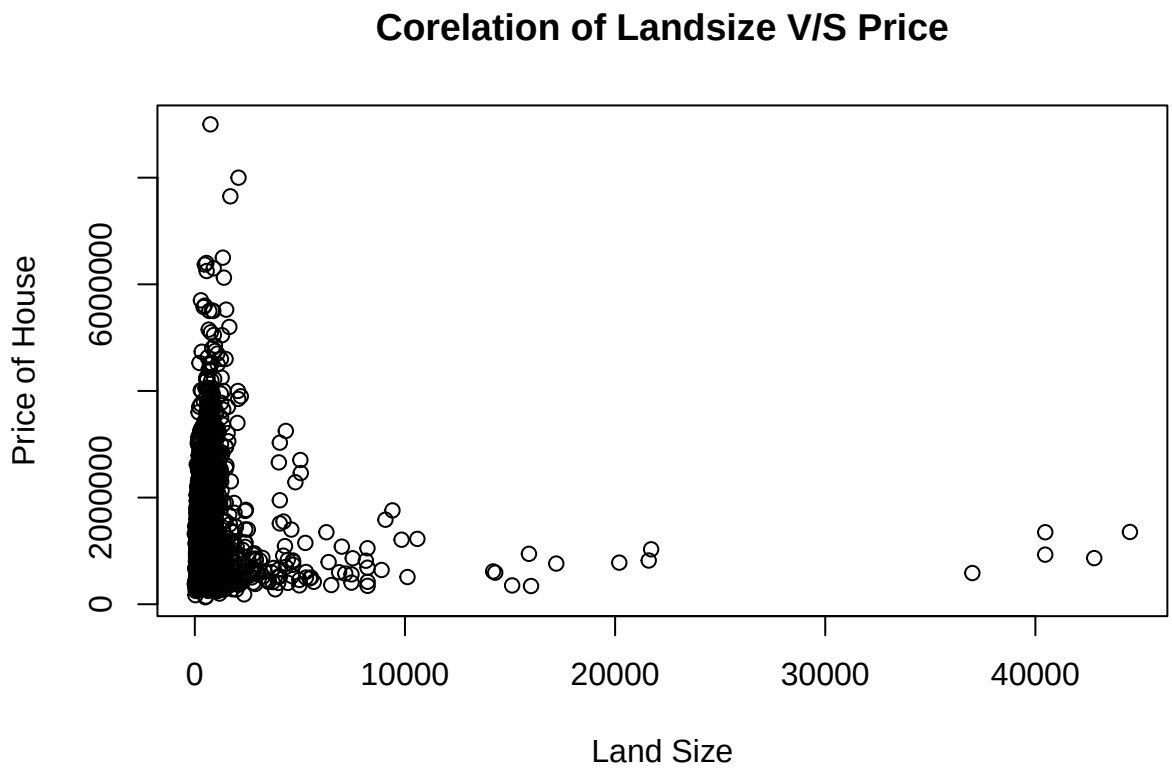

```
Frequency_House_1<- f %>%  group_by(Landsize) %>%
  summarise(House = n(), countSum = sum(as.integer(Propertycount)), Mean_count = mean(Propertycount
ggplot(Frequency_House_1,aes(x = "", y = House ,size = f$Price  , colour = Landsize))+ geom_point()
```



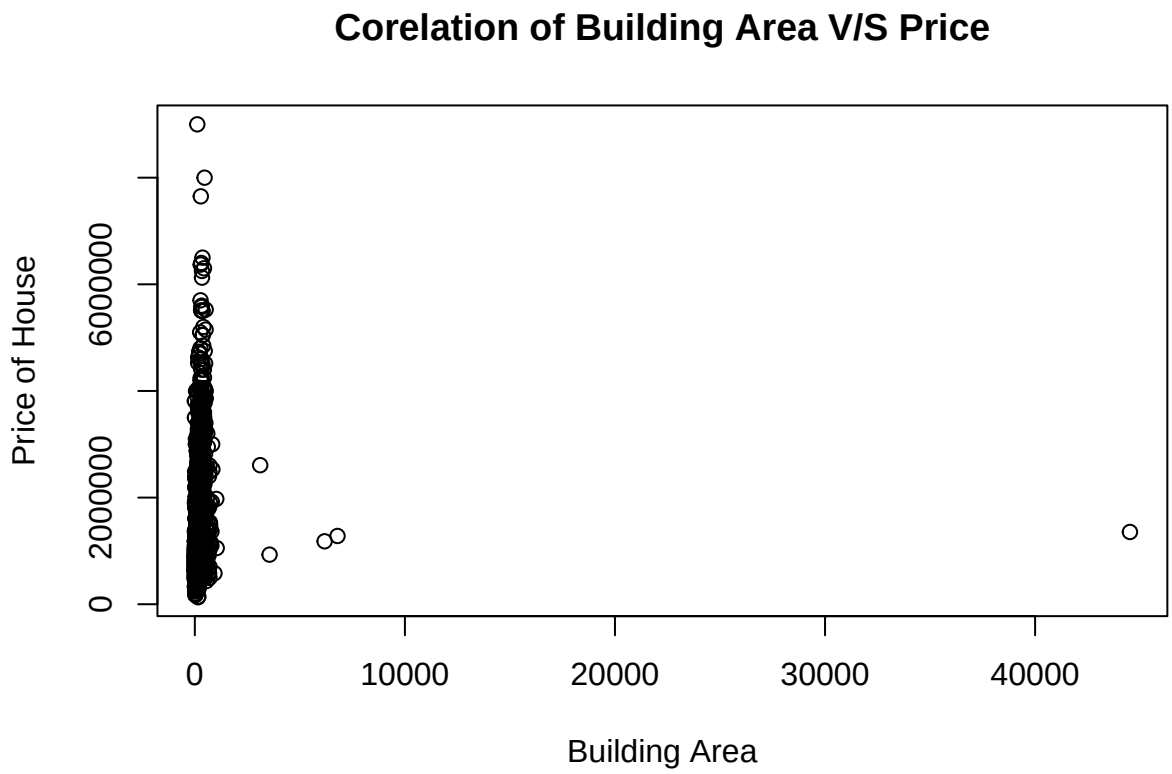
6. What all different attributes are correlated with the price and which 3 variables are correlated the most with the price.

    1. Land Size
    2. Building Area
    3. Number of Rooms

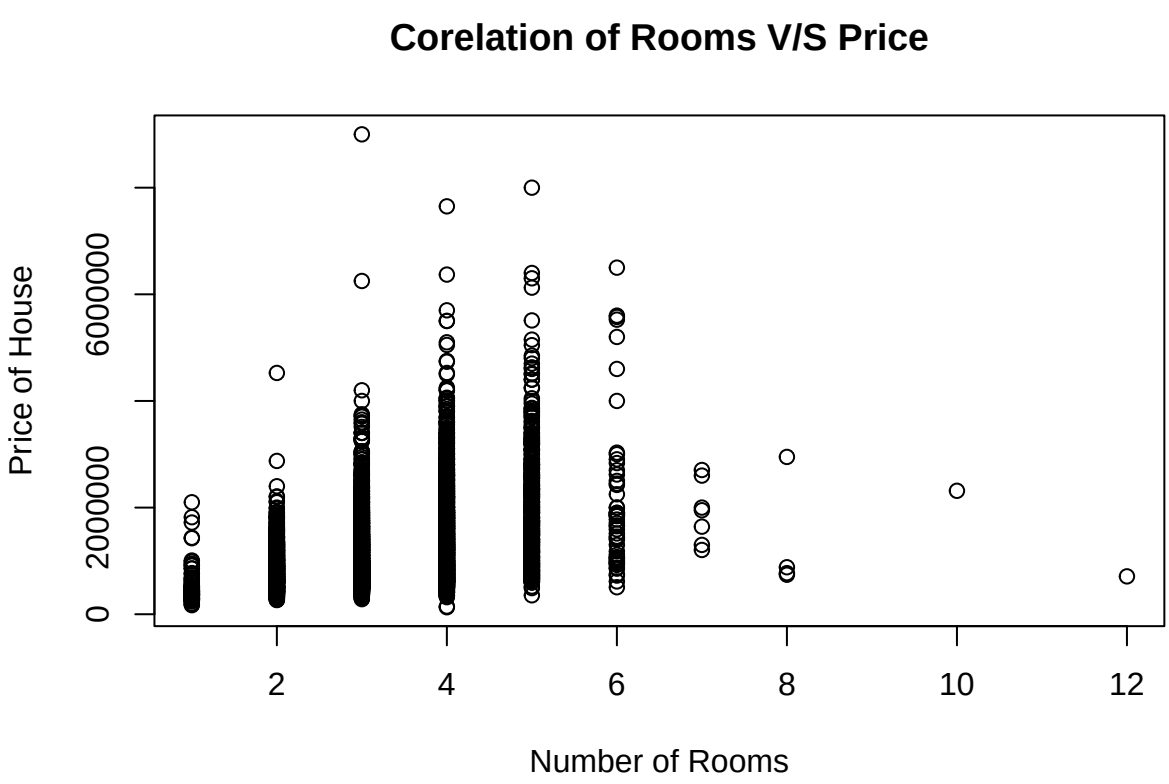
```
plot(f$Landsize,f$Price,xlab = "Land Size",ylab = "Price of House",main = "Corelation of Landsize V/S P
```




```
plot(f$BuildingArea,f$Price,xlab = "Building Area",ylab = "Price of House",main = "Corelation of Buildi
```




```
plot(f$Rooms,f$Price,xlab = "Number of Rooms",ylab = "Price of House",main = "Corelation of Rooms V/S P
```



we conclude the work by saying that as we go higher in land size we should expect increase in price but its not certain that number of room will be greater in a higher costing house.