Assignment 1 - Big Data Analysis - Melbourne House dataset EDA

Alban Tchikladzé

26/10/2021

Part 1 : Description and loading libraries

The housing dataset contains the prices and other attributes of almost 35,000 houses in the city of Melbourne. Your task is to perform an Exploratory Data Analysis on the dataset

```
#Library ggplot2, to plot graphics about the data
library(ggplot2)

#Library corrplot, to plot graphic about the correlations
library(corrplot)
```

```
## corrplot 0.90 loaded
```

Part 2 : Loading the data

First download the dataset from Moodle's module page.

Below is some brief details of the dataset variables:

- - -

X: Primary Key to identify houses

Rooms: Number of rooms

Price: Price in Australian dollars

Date: Date sold

Type: House type (h=house, u=unit/duplex, t=townhouse)

Distance: Distance from Central Business District in KMs

Regionname: General Region (West, North West, North, etc.)

Propertycount: Number of properties that exist in the suburb

Bathroom: Number of bathrooms

Car: Number of carspots

Landsize: Land size in Metres

BuildingArea: Building size in Metres

YearBuilt: Year the house was built

- - -

```
# checking if the file exist
file.exists("melbourne_data.csv")
```

```
## [1] TRUE
```

```
# load the csv file from a local PC file
housing.dataset <- read.csv("melbourne_data.csv")

#exploring the structure of the data
#str(housing.dataset)

#overview of the dataset's values
#head(housing.dataset)
summary(housing.dataset)
```

```
##        X              Date               Type               Price         
##  Min.   :    1   Length:34857       Length:34857       Min.   :   85000  
##  1st Qu.: 8715   Class :character   Class :character   1st Qu.:  635000  
##  Median :17429   Mode  :character   Mode  :character   Median :  870000  
##  Mean   :17429                                         Mean   : 1050173  
##  3rd Qu.:26143                                         3rd Qu.: 1295000  
##  Max.   :34857                                         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  
##                    
##                    
##                    
##                    
```

Part 3.1 : Clean the dataset and prepare it for analysis

Your first task is to clean the dataset and prepare it for analysis by e.g. removing/replacing NAs, outliers, and incorrect values.

Let's first remove the outliers and incorrect values.

```
#First column : X

#length(unique(housing.dataset$X))
#X is ranging form 1 to 34 857 with no NA so it does exactly his primary key job and there are no incoherence



#Second column : Date

#We print all the unique date values to have a look at them
#unique(housing.dataset$Date)
#Then we transform them to regular date type
housing.dataset$Date <- as.Date(housing.dataset$Date, "%d/%m/%Y")
#we get all the value having a date below the beginning of 2016
datesBelow2016 <- ifelse(housing.dataset$Date < as.Date("2016-01-01"), TRUE, FALSE)
#we get all the value having a date later than the beginning of 2020
allDates <- c(datesBelow2016, ifelse(housing.dataset$Date > as.Date("2020-01-01"), TRUE, FALSE))
#is there any value corresponding to the pattern that we where hoping not to find ?
#isTRUE(allDates)
#as they are all false, there is no incoherence in this column then and all the dates are between 2016 and 20219



#Third column : Type

housing.dataset$Type <- as.factor(housing.dataset$Type)
#unique(housing.dataset$Type)
#we can see that there is only 3 types in Types, as expected in the data description from the start
#no incoherence in this column



#To get rid of the outliers I have chosen to use the statistics from the box plot as it delete approximately 300
values each time, which is so number you should expect.


#I have chose this compare to the quartile method "by hand", which was deleting to many data, and I found myself
```

```
with less than 10 000 after it. So that why I'm using the boxplot statistics, as they are more precise, even thos
e at the end it's still a quartile discrimination.



#Fourth Column : Price

outliers <- boxplot.stats(housing.dataset$Price)$out
housing.dataset <- housing.dataset[-which(housing.dataset$Price %in% outliers),]



#Fifth Column : Landsize

outliers <- boxplot.stats(housing.dataset$Landsize)$out
housing.dataset <- housing.dataset[-which(housing.dataset$Landsize %in% outliers),]



#Sixth Column  : BuildingArea

outliers <- boxplot.stats(housing.dataset$BuildingArea)$out
housing.dataset <- housing.dataset[-which(housing.dataset$BuildingArea %in% outliers),]



#Seventh Column : Rooms

outliers <- boxplot.stats(housing.dataset$Rooms)$out
housing.dataset <- housing.dataset[-which(housing.dataset$Rooms %in% outliers),]



#Eighth Column : Bathroom

outliers <- boxplot.stats(housing.dataset$Bathroom)$out
housing.dataset <- housing.dataset[-which(housing.dataset$Bathroom %in% outliers),]



#Ninth Column : Car

outliers <- boxplot.stats(housing.dataset$Car)$out
housing.dataset <- housing.dataset[-which(housing.dataset$Car %in% outliers),]



#Tenth Column : YearBuilt

outliers <- boxplot.stats(housing.dataset$YearBuilt)$out
housing.dataset <- housing.dataset[-which(housing.dataset$YearBuilt %in% outliers),]



#Eleventh Column : Distance


housing.dataset$Distance <- as.numeric(housing.dataset$Distance)
```

```
## Warning: NAs introduits lors de la conversion automatique
```

```
outliers <- boxplot.stats(housing.dataset$Distance)$out
housing.dataset <- housing.dataset[-which(housing.dataset$Distance %in% outliers),]



#Twelvth Column : Regionname

housing.dataset$Regionname <- as.factor(housing.dataset$Regionname)



#Thirteenth Column : Propertycount

housing.dataset$Propertycount <- as.integer(housing.dataset$Propertycount)
```

```
## Warning: NAs introduits lors de la conversion automatique
```

```r
outliers <- boxplot.stats(housing.dataset$Propertycount)$out
housing.dataset <- housing.dataset[-which(housing.dataset$Propertycount %in% outliers),]
```

Then we will get rid of the NA

```r
#this one is a bit hard, it make us cut 25 962 values over the 34 857 values that are in the dataset, so we are l
eft with only 8 895 values, so let's find a more subtle way of replacing the N/A
#housing.dataset <- na.omit(housing.dataset)


#we will check how many data get deleted by the different thresholds as we don't want to get rid of two many usef
ul data
nrow(housing.dataset[rowSums(is.na(housing.dataset))>4,])
```

```
## [1] 7482
```

```r
nrow(housing.dataset[rowSums(is.na(housing.dataset))>5,])
```

```
## [1] 1683
```

```r
nrow(housing.dataset[rowSums(is.na(housing.dataset))>6,])
```

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

```r
#so we will finally deleted the rows that have more than 6 NA, as it will allow us to not loose so many data and
7 out of 13 data could still be useful
housing.dataset <- housing.dataset[rowSums(is.na(housing.dataset))<=5,]


#Find The number of NA values in each column
naCcount<-sapply(housing.dataset,function(x) sum(is.na(x)==TRUE))
naCcount
```

```
##             X          Date          Type         Price      Landsize
##             0             0             0          4835          8661
##  BuildingArea         Rooms      Bathroom           Car     YearBuilt
##         16435             0          5788          6237         14848
##      Distance    Regionname Propertycount
##             0             0             0
```

```r
# Find percent of nulls in each column

for(i in 1:ncol(housing.dataset)) {
  colName <- colnames(housing.dataset[i])
  pctNull <- sum(is.na(housing.dataset[,i]))/length(housing.dataset[,i])
  if (pctNull > 0.20) {
  print(paste("Column ", colName, " has ", round(pctNull*100, 3), "% of nulls"))
  }
}
```

```
## [1] "Column  Landsize  has  31.735 % of nulls"
## [1] "Column  BuildingArea  has  60.219 % of nulls"
## [1] "Column  Bathroom  has  21.208 % of nulls"
## [1] "Column  Car  has  22.853 % of nulls"
## [1] "Column  YearBuilt  has  54.404 % of nulls"
```

```r
#we can maybe drop columns with more than 50 percent NA values
#housing.dataset[,c("BuildingArea","YearBuilt")]<-NULL
#but we will not as it's not useful to dismiss valuable data, that can still be useful
```

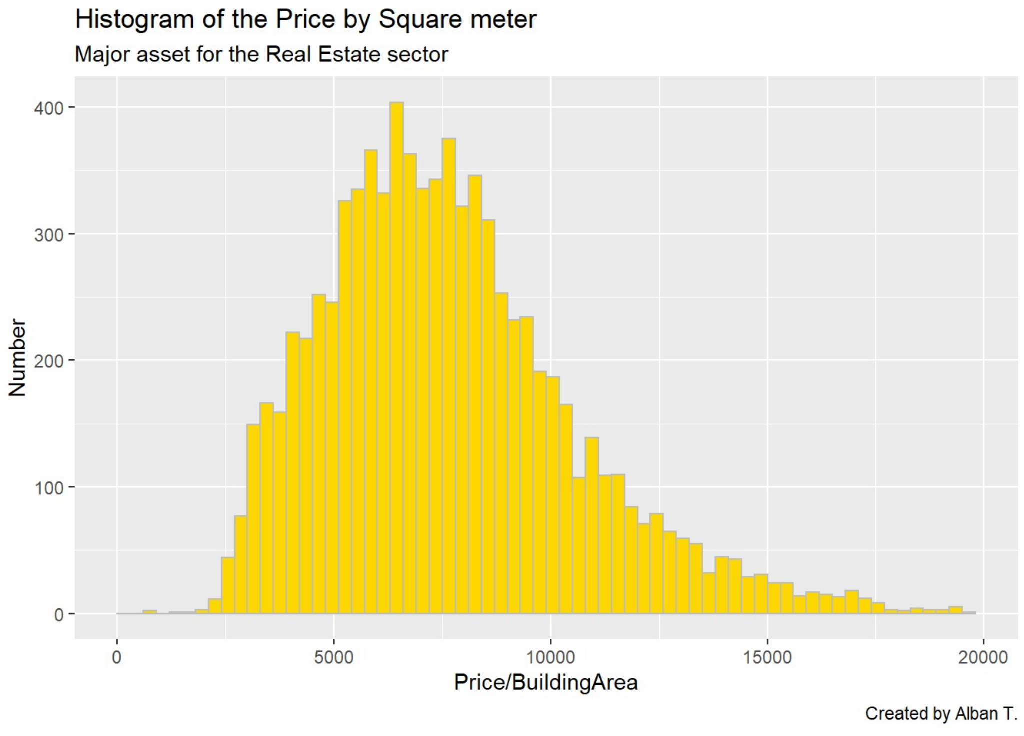
Part 3.2 : Summary of the variables and the 4 plots

```r
summary(housing.dataset)
```

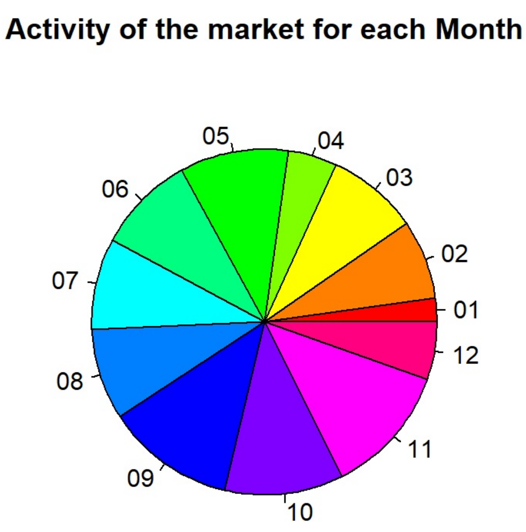
```
##        X               Date               Type         Price
##  Min.   :    1   Min.   :2016-01-28   h:18206   Min.   :  85000
##  1st Qu.: 8155   1st Qu.:2016-11-12   t: 2999   1st Qu.: 640000
##  Median :16831   Median :2017-06-24   u: 6087   Median : 867500
##  Mean   :17021   Mean   :2017-05-15             Mean   : 966501
##  3rd Qu.:25575   3rd Qu.:2017-10-28             3rd Qu.:1242000
##  Max.   :34857   Max.   :2018-03-17             Max.   :2285000
##                                                 NA's   :4835
##     Landsize       BuildingArea       Rooms          Bathroom
##  Min.   :   0.0   Min.   :  0.0   Min.   :1.000   Min.   :0.000
##  1st Qu.: 189.0   1st Qu.: 98.0   1st Qu.:2.000   1st Qu.:1.000
##  Median : 430.0   Median :129.0   Median :3.000   Median :1.000
##  Mean   : 420.3   Mean   :136.3   Mean   :2.928   Mean   :1.532
##  3rd Qu.: 638.0   3rd Qu.:170.0   3rd Qu.:3.000   3rd Qu.:2.000
##  Max.   :1332.0   Max.   :298.2   Max.   :7.000   Max.   :3.000
##  NA's   :8661     NA's   :16435                   NA's   :5788
##       Car          YearBuilt        Distance
##  Min.   :0.000   Min.   :1863    Min.   : 0.00
##  1st Qu.:1.000   1st Qu.:1940    1st Qu.: 6.20
##  Median :2.000   Median :1970    Median : 9.70
##  Mean   :1.524   Mean   :1964    Mean   :10.28
##  3rd Qu.:2.000   3rd Qu.:1999    3rd Qu.:13.80
##  Max.   :3.000   Max.   :2018    Max.   :25.20
##  NA's   :6237    NA's   :14848
##                       Regionname    Propertycount
##  Southern Metropolitan     :9195   Min.   :  389
##  Northern Metropolitan     :7551   1st Qu.: 4280
##  Western Metropolitan      :5803   Median : 6543
##  Eastern Metropolitan      :3690   Mean   : 7192
##  South-Eastern Metropolitan:1021   3rd Qu.:10160
##  Northern Victoria         :  31   Max.   :17496
##  (Other)                   :   1
```

```
#This plot represent a significant value in the real estate sector, it's the histogram of the Price by Square met
ers
ggplot(data=housing.dataset, aes(Price/(BuildingArea))) + geom_histogram(breaks=seq(0, 20000, by=300), color = "g
rey", fill = "#ffd700") + labs(title="Histogram of the Price by Square meter ", subtitle="Major asset for the Rea
l Estate sector", caption="Created by Alban T.", x="Price/BuildingArea", y="Number")
```

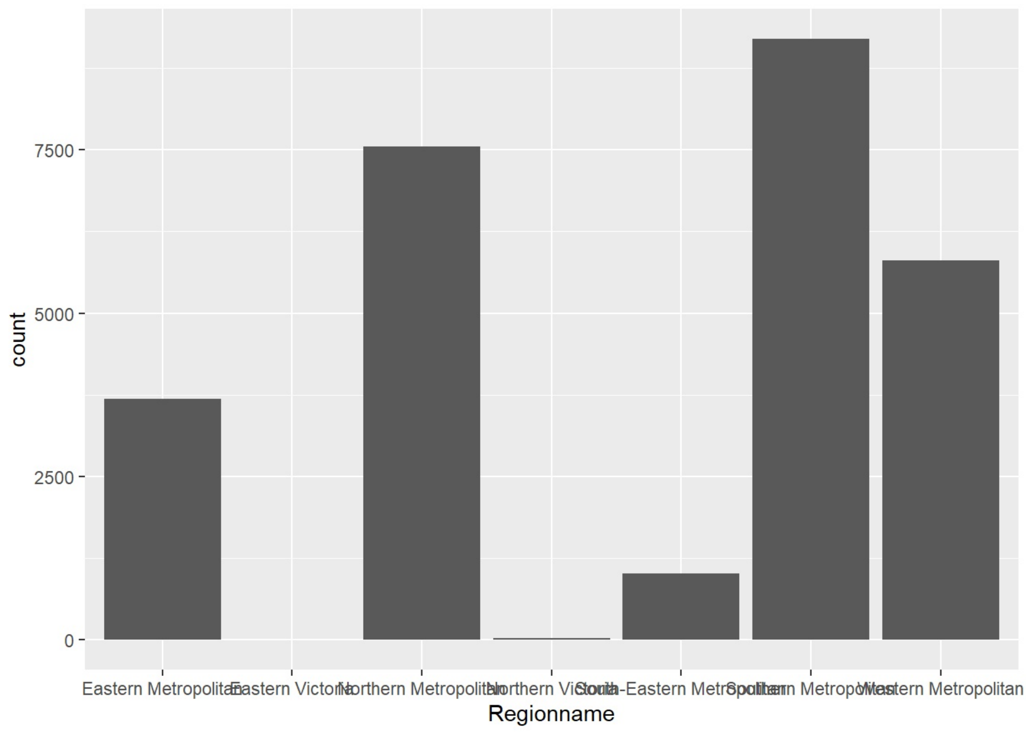
```
## Warning: Removed 18961 rows containing non-finite values (stat_bin).
```




```
#This is a pie chart showing the number of sales by month, the activity of the market
pie(table(format(as.Date(housing.dataset$Date), "%m")),col=rainbow(12), main= "Activity of the market for each Mo
nth")
```




```
#I had to use the pie function because ggplot pie was killing my R session, I don't know why but ggplot is not we
ll adapted for pie.
#ggplot(housing.dataset, aes(x = "", y = format(as.Date(Date), "%m"))) + geom_col() + coord_polar(theta = "y")

#This plot shows the number of house by regionname
ggplot(data = housing.dataset, aes(Regionname), aes(color = Regionname)) + geom_bar()
```



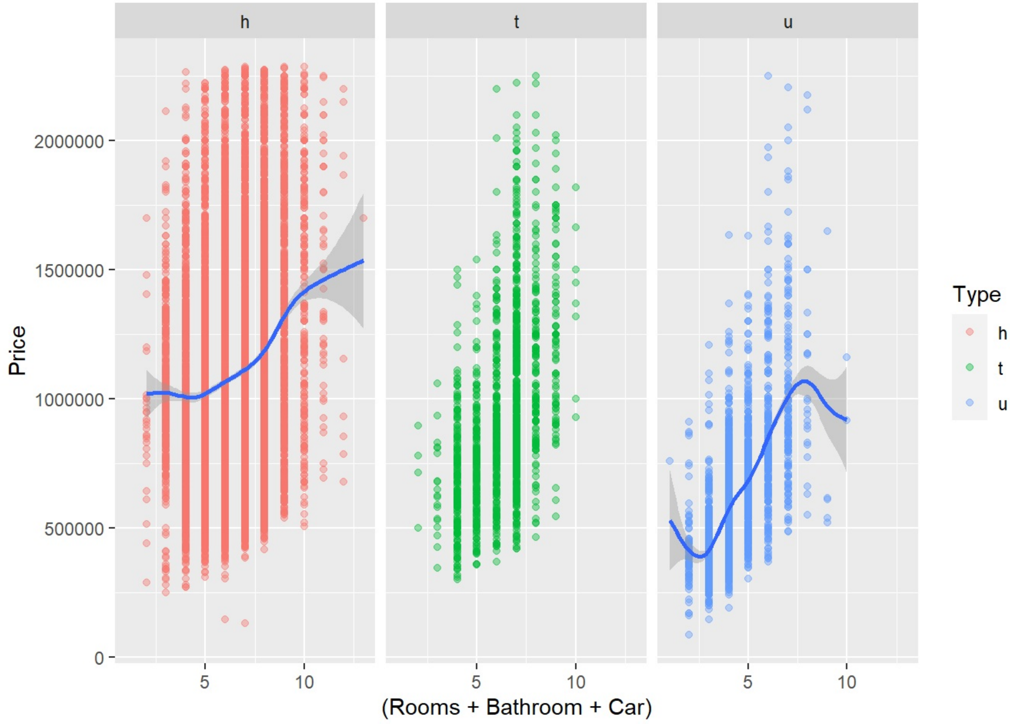

```
#This plot show the total number of house parts related to the price for each house type
ggplot(data = housing.dataset, aes(x=(Rooms + Bathroom + Car), y=Price)) + geom_point(alpha = 0.4, aes(color = Ty
pe)) + facet_wrap(~Type)+ geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 10956 rows containing non-finite values (stat_smooth).
```

```
## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.
```

```
## Warning: Removed 10956 rows containing missing values (geom_point).
```

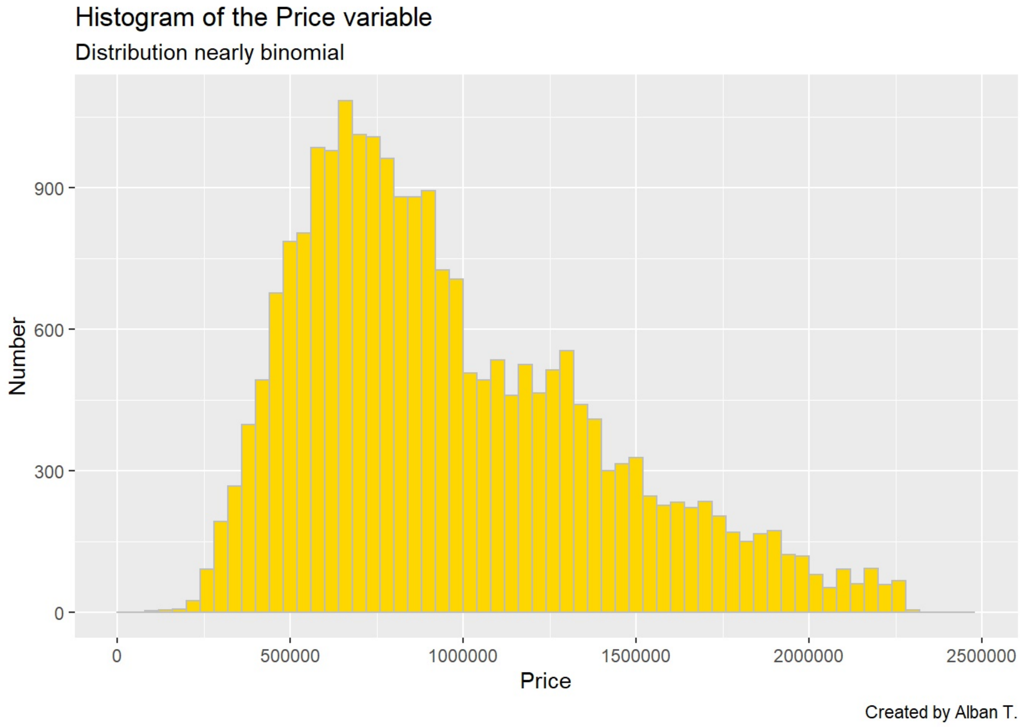



```
#We can see that their is a trend emerging for this and, in average, the more parts the higher the price.
```

Part 3.3.a : histogram of the price variable

```
#Here is the historgram of the price variable
ggplot(data=housing.dataset, aes(Price)) + geom_histogram(breaks=seq(0, 2500000, by=40000), color = "grey", fill
= "#ffd700") + labs(title="Histogram of the Price variable", subtitle= "Distribution nearly binomial", caption="C
reated by Alban T.", x="Price", y="Number")
```

```
## Warning: Removed 4835 rows containing non-finite values (stat_bin).
```




```
#We can  see that the Price variable is almost following a binomial distribution with p=0.2
```

Part 3.3.b : Group houses by some price ranges & summarize them

```
#taking the Price column  and the primary key from the data set
HouseGroups = subset(housing.dataset, select = c(Price, X))

#cutting the data in 3 parts according to the primary key and transforming them into numeric
HouseGroups$Groups = as.numeric(cut(HouseGroups$X, 3))

#Summarize data from the group 1 : "Low Price"
print("Group 1 : Low Price")
```

```
## [1] "Group 1 : Low Price"
```

```
summary(HouseGroups$Price[which(HouseGroups$Groups=="1")])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   85000  626000  875000  971465 1260000 2285000    1757
```

```
#Summarizing data from the group 2 : "Middle Price"
print("Group 2 : Middle Price")
```

```
## [1] "Group 2 : Middle Price"
```

```
summary(HouseGroups$Price[which(HouseGroups$Groups=="2")])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  121000  637000  870000  964288 1240000 2285000    1727
```

```
#summarizing Data from the group 3 : "High Price"
print("Group 3 : High Price")
```

```
## [1] "Group 3 : High Price"
```

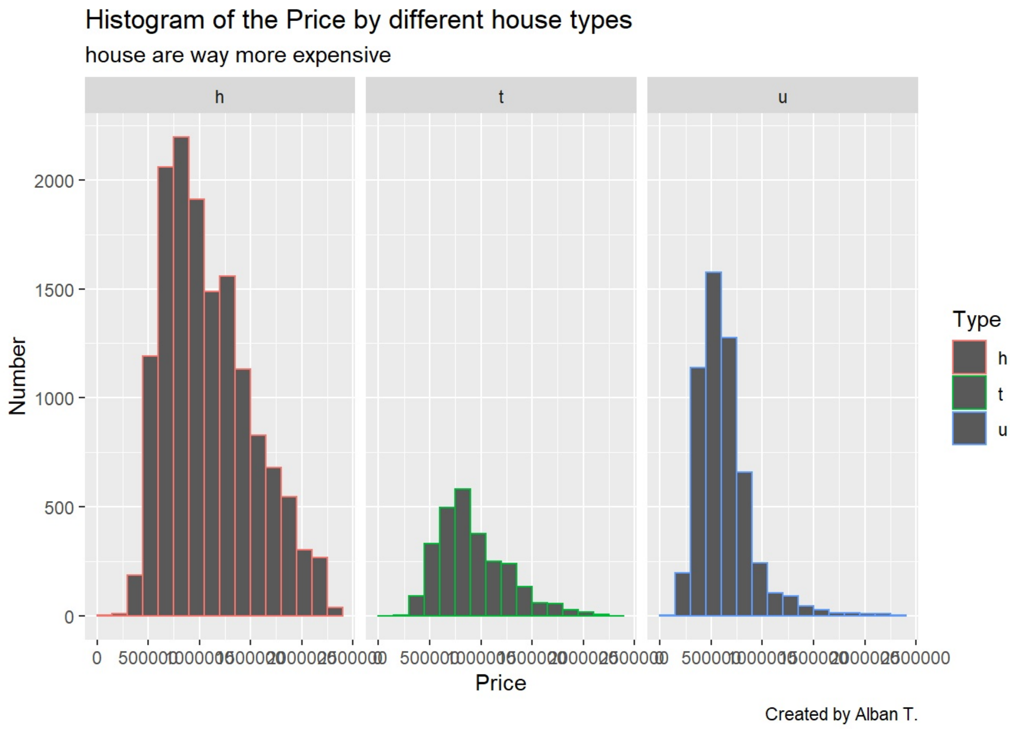
```
summary(HouseGroups$Price[which(HouseGroups$Groups=="3")])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  112000  650000  860000  963496 1220000 2285000    1351
```

Part 3.3.c : Explore prices for different house types

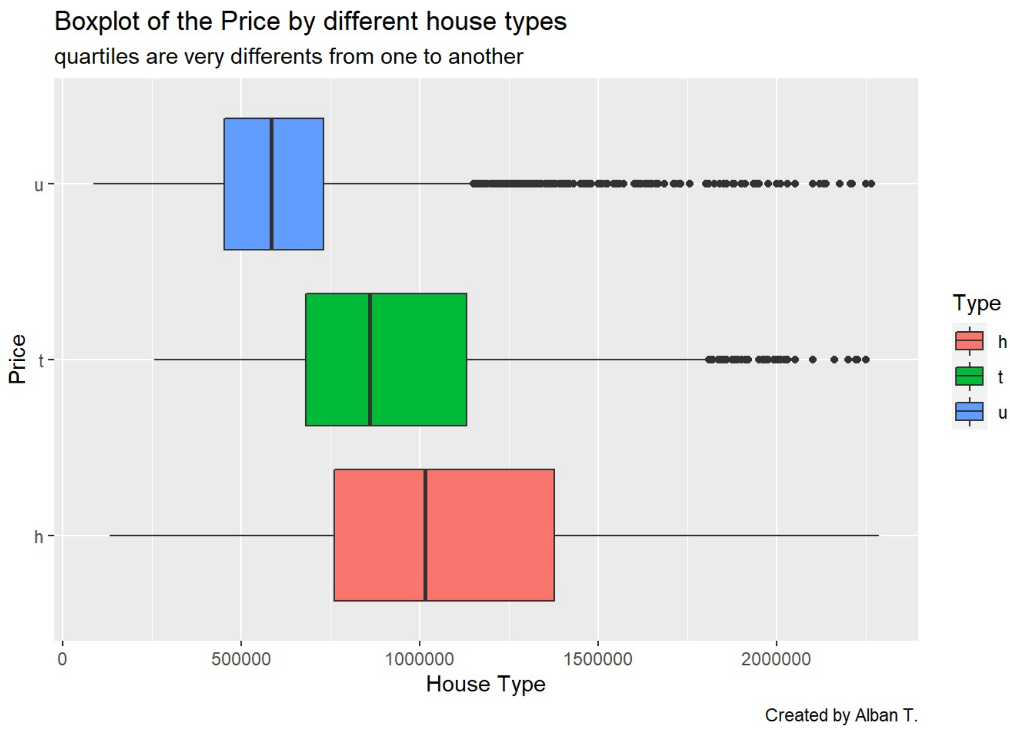
```
#This graph show the histogram of the price by different house types
ggplot(data = housing.dataset, aes(Price)) + geom_histogram(breaks=seq(0, 2500000, by=150000), aes(color=Type)) +
facet_wrap(~Type) + labs(title="Histogram of the Price by different house types", subtitle= "house are way more e
xpensive", caption="Created by Alban T.", x="Price", y="Number")
```

```
## Warning: Removed 4835 rows containing non-finite values (stat_bin).
```




```
#This plot show the prices differences between the different house types in the form of a boxplot
ggplot(data=housing.dataset, aes(y=Price, x=Type, fill=Type)) + geom_boxplot() + coord_flip()+ labs(title="Boxplo
t of the Price by different house types", subtitle= "quartiles are very differents from one to another", caption=
"Created by Alban T.", x="Price", y="House Type")
```

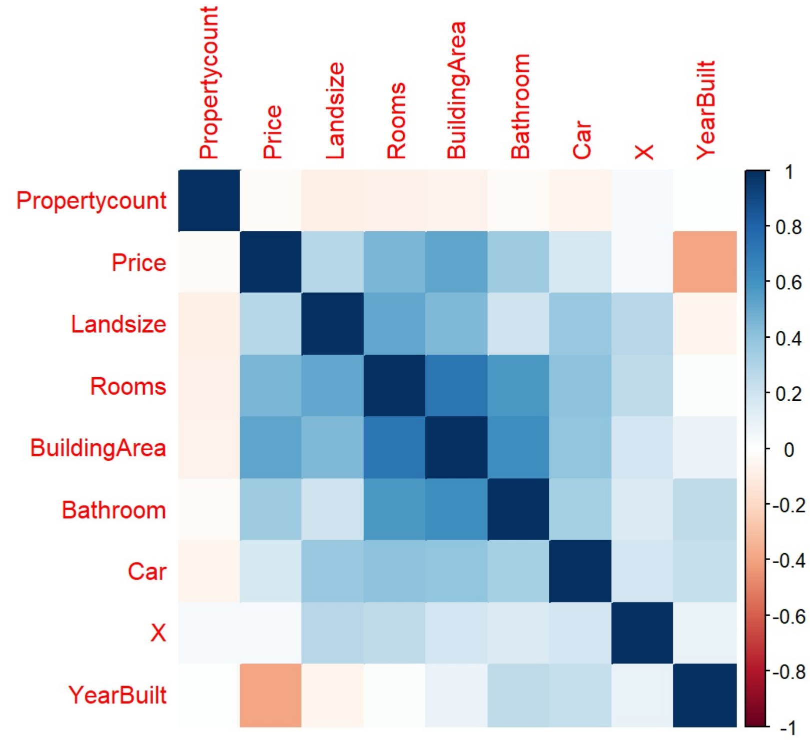
```
## Warning: Removed 4835 rows containing non-finite values (stat_boxplot).
```




```
#We have in here the different quartiles of each house types
```

Part 3.3.d : Correlation with price

```
#We exclude non-numeric data from the data set, as they are not fitted for the correlation
CorData <- housing.dataset[,c("X","Price","Landsize","BuildingArea","Rooms","Bathroom","Car","YearBuilt","Propert
ycount")]

#then we omit the NA, because they prevent us to do the correlation
CorData <- na.omit(CorData)

#We finally have the correlation matrix which is a general matrix for all the data set
CorMatrix <- cor(CorData)

#We can print it to see what it looks like
corrplot(CorMatrix, method = 'color', order = 'AOE')
```



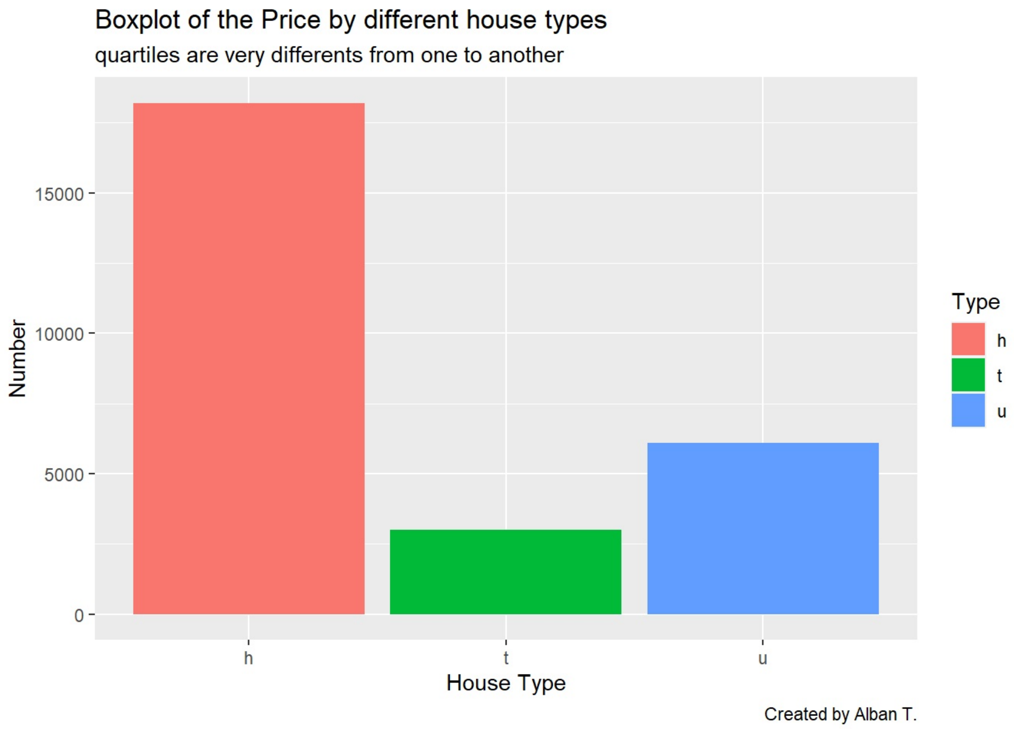

```
#Then let's take only the column for the Price
CorMatrix[,2]
```

```
##             X         Price      Landsize  BuildingArea         Rooms
##    0.03026125    1.00000000    0.28064201    0.52764544    0.45639407
##      Bathroom           Car     YearBuilt Propertycount
##    0.35586878    0.17782308   -0.39746188   -0.02679703
```

```
#We clearly see that their is three column with a reasonable amount of correlation with Price.
#Those are BuildingArea, Rooms and Bathroom, which make sense as it's what people primary look at when choosing a
house.

#So, it was a good idea not to get rid of Building Area, despite the high number of NA
```

Part 4 : Frequencies of houses for various types & 2 scatter plots

```
#This plot show the frequencies of houses for various types
ggplot(data = housing.dataset, aes(Type)) + geom_bar(aes(fill = Type)) + labs(title="Boxplot of the Price by diff
erent house types", subtitle= "quartiles are very differents from one to another", caption="Created by Alban T.",
x="House Type", y="Number")
```

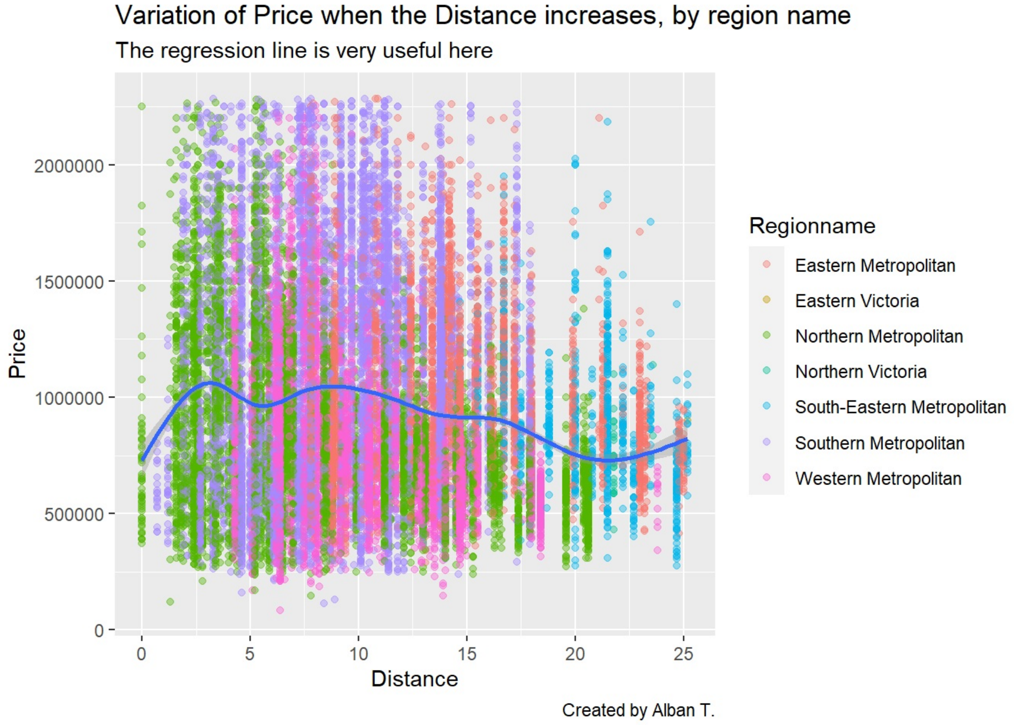



```
#This plot show the variations of prices when the distances from the district center increase for each of the reg
ion names
ggplot(data = housing.dataset, aes(x=Distance, y=Price)) + geom_point(alpha = 0.4, aes(color = Regionname))+ geom
_smooth() + labs(title="Variation of Price when the Distance increases, by region name", subtitle= "The regressio
n line is very useful here", caption="Created by Alban T.", x="Distance", y="Price")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 4835 rows containing non-finite values (stat_smooth).
```

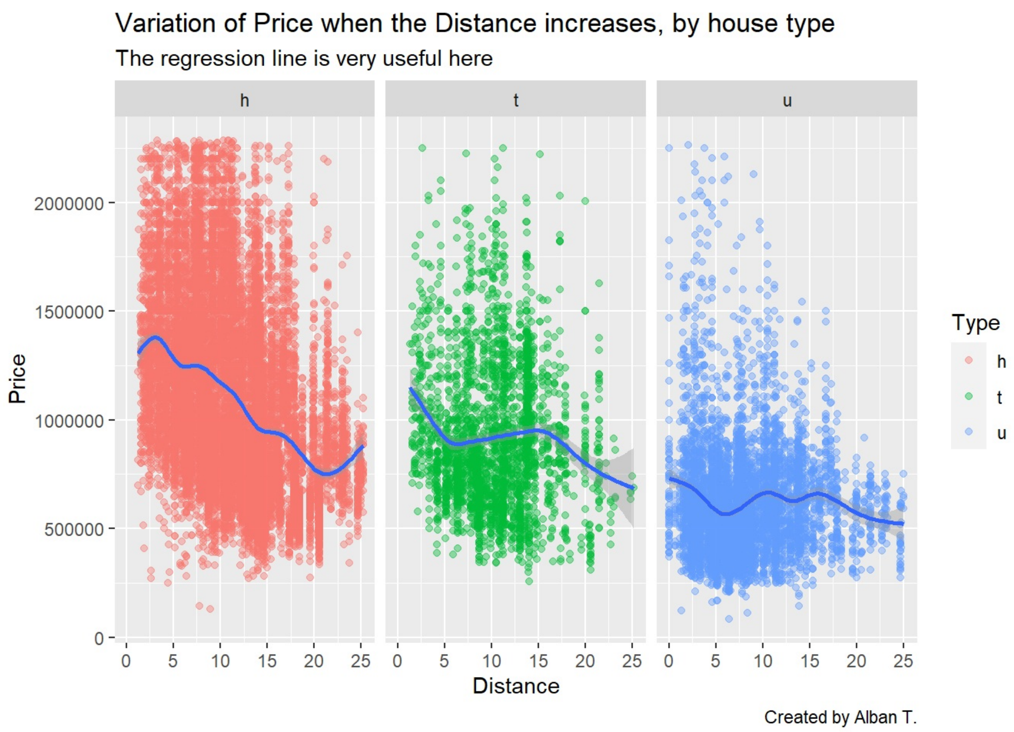
```
## Warning: Removed 4835 rows containing missing values (geom_point).
```



```
#You can see that the more we get closer to the district center, the more variations their is in term of prices b
ut the average seems to be almost constant



#This plot show the variations of prices when the distances from the district center increase for each of the hou
se types
ggplot(data = housing.dataset, aes(x=Distance, y=Price)) + geom_point(alpha = 0.4, aes(color = Type))+ geom_smoot
h() + facet_wrap(~Type)+ labs(title="Variation of Price when the Distance increases, by house type", subtitle= "T
he regression line is very useful here", caption="Created by Alban T.", x="Distance", y="Price")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 4835 rows containing non-finite values (stat_smooth).

## Warning: Removed 4835 rows containing missing values (geom_point).
```




```
#You can see that the more we get closer to the district center, the more the average goes up, especially for the
houses.
#However, their is a little ascent in the middle for townhouses, it's maybe because it's where you can find the m
ost of townhouses so there is much competition
```