Project
Objective: Aim of this analysis is to answer a question of "Which ones are the best for investments?"

```
library(ggplot2)
library(tidyverse)
library(dplyr)
library(scales)
library(rpart)
library(rpart.plot)
library(randomForest)
library(caret)
library(Metrics)
library(ggthemes)

# Load data
rm(list=ls())
df<-read.csv(file.choose())

# Check data structure and summary
dim(df)
[1] 3164    6
> str(df)
'data.frame':	3164 obs. of  6 variables:
 $ Id       : Factor w/ 2967 levels "","C4230142",..: 2317 2654 2536 355 2460 2311 2389 2535 2284 2829
 ...
 $ Address  : Factor w/ 2946 levels "","1 Aberfoyle Cres 1109, Toronto",..: 2163 1171 1348 1310 950 162
7 2675 1347 925 2227 ...
 $ Bedrooms : int  1 1 1 3 1 1 5 1 1 1 ...
 $ Bathrooms: int  1 1 1 1 1 1 3 1 1 1 ...
 $ Type     : Factor w/ 99 levels "","Att/Row/Twnhouse 2-Storey",..: 75 4 81 33 28 19 33 81 88 19 ...
 $ Price    : int  650 700 700 799 800 800 800 800 950 1000 ...
> summary(df)
        Id                          Address        Bedrooms     Bathrooms                        Type
 C4311344:   2   101 Peter St 516, Toronto    :   3   Min.   :1.0   Min.   :1.000   Condo Apt Apartment        :
2066
 C4320832:   2   18 Kenaston Gdns 1605, Toronto:   3   1st Qu.:1.0   1st Qu.:1.000   Detached 2-Storey
      : 183
 C4322238:   2   55 Stewart St 932, Toronto    :   3   Median :2.0   Median :1.000   Detached Bungalow
     : 112
 C4327202:   2   65 St Mary St 2503, Toronto   :   3   Mean   :2.1   Mean   :1.626   Comm Element Condo
Apartment:  56
 C4327328:   2   1 Arundel Ave Main, Toronto   :   2   3rd Qu.:3.0   3rd Qu.:2.000   Semi-Detached 2-Store
y      :  54
 C4329247:   2   1 Bloor St E 1603, Toronto    :   2   Max.   :8.0   Max.   :8.000   Condo Townhouse 3-Stor
ey    :  48
 (Other) :3152   (Other)                       :3148   NA's   :1     NA's   :1      (Other)                  : 645
     Price
 Min.   :  650
 1st Qu.: 2150
 Median : 2500
 Mean   : 3001
 3rd Qu.: 3200
 Max.   :22500
```

NA's :1

```
# Change datatype
df$Price <- as.numeric(df$Price)
df$Bedrooms <- as.numeric(df$Bedrooms)
df$Bathrooms <- as.numeric(df$Bathrooms)
df$Type <- as.character(df$Type)
df$Address <- as.character(df$Address)
df$Id<- as.character(df$Id)

# Check duplicates and remove duplicates
duplicated(df$Id)
df <- df[!duplicated(df$Id), ]
dim(df)

# Checking missing values and remove them
colSums(is.na(df)|df=='')
df<-df[complete.cases(df),]

# Stats information about the Price,Bedrooms,Bathrooms after duplicates removed
dim(df)
summary(df$Price)
summary(df$Bedrooms)
summary(df$Bathrooms)
```

```
> dim(df)
[1] 2966    6
> summary(df$Price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    650    2100    2475    3022    3200   22500 
> summary(df$Bedrooms)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   1.000   2.000   2.104   3.000   8.000 
> summary(df$Bathrooms)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   1.000   1.000   1.634   2.000   8.000 
```

```
# Change datatype of Bedrooms and Bathrooms for plotting
df$Bedrooms <- as.character(df$Bedrooms)
df$Bathrooms <- as.character(df$Bathrooms)

# Count the total number of properties by type
df %>% group_by(Type) %>% summarize(count=n())
```

```
Type                                count
   <chr>                              <int>
 1 Att/Row/Twnhouse 2-Storey             21
 2 Att/Row/Twnhouse 2 1/2 Storey          3
 3 Att/Row/Twnhouse 3-Storey             32
 4 Att/Row/Twnhouse Apartment             3
```

```
 5 Att/Row/Twnhouse Other                  2
 6 Co-Op Apt Apartment                     6
 7 Co-Ownership Apt 2-Storey               1
 8 Co-Ownership Apt Apartment              1
 9 Co-Ownership Apt Bachelor/Studio        1
10 Comm Element Condo 2-Storey             1
# ... with 88 more rows
```

Total number of Type of Properties : 98

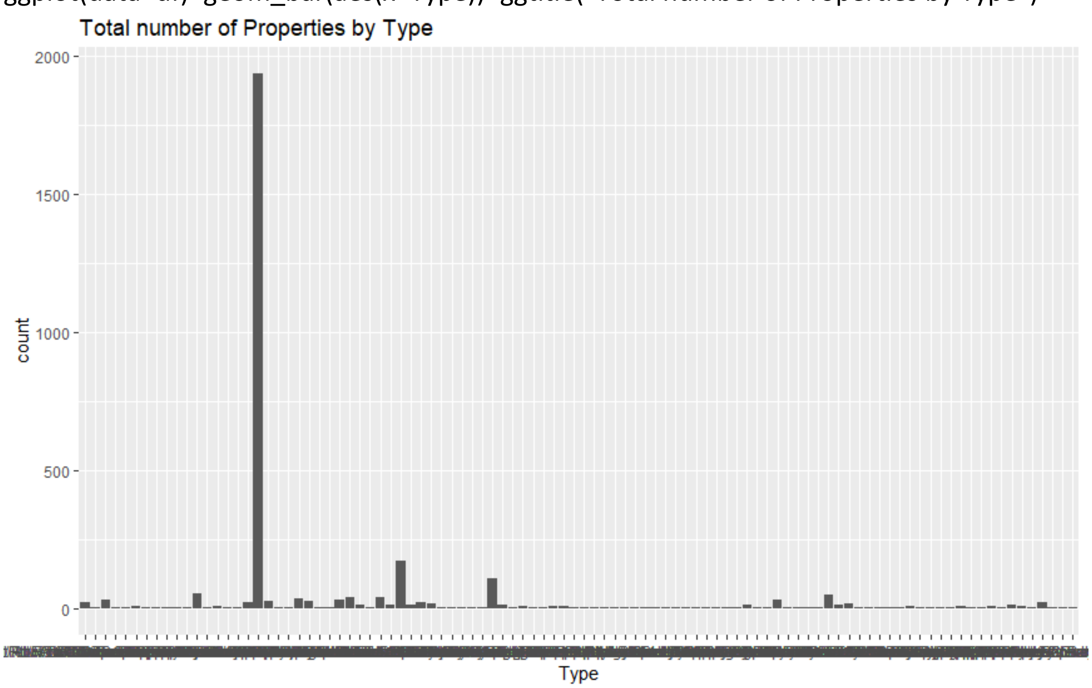
List unique Type : total 98 types

unique(df$Type)

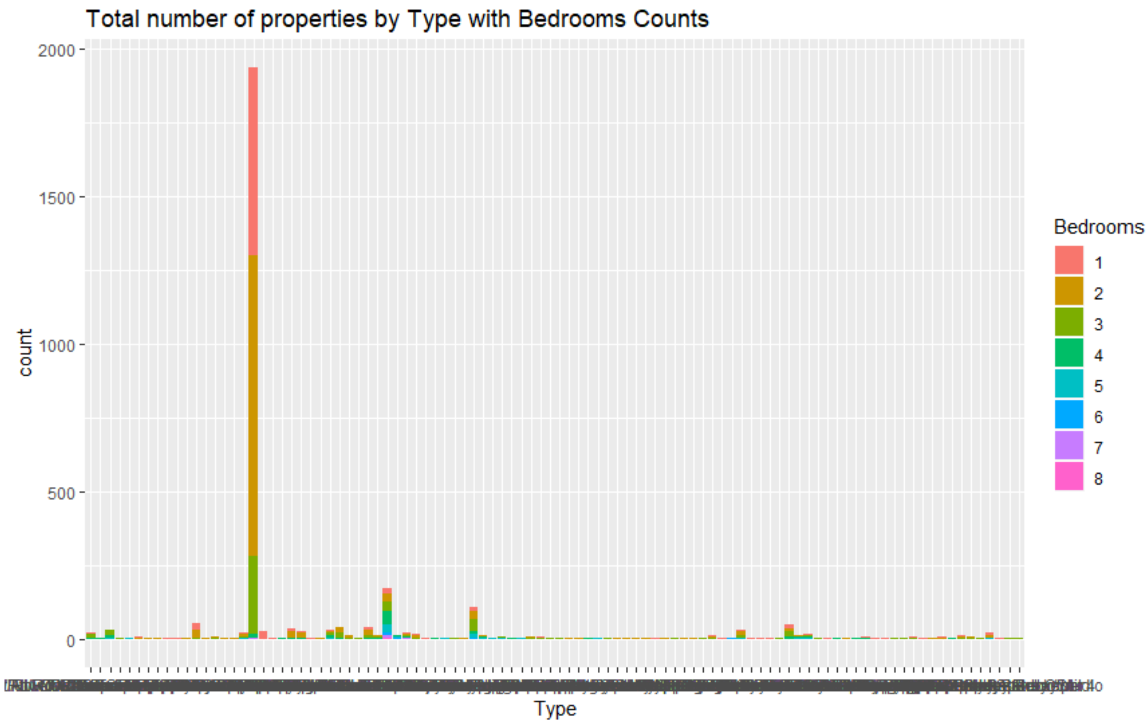
```
> unique(df$Type)
 [1] "Semi-Detached 2-Storey"                "Att/Row/Twnhouse 3-Storey"
  "Semi-Detached Backsplit 5"
 [4] "Detached 2-Storey"                     "Condo Townhouse 3-Storey"
  "Condo Apt Apartment"
 [7] "Store W/Apt/Offc Apartment"            "Lower Level Bachelor/Studio"
  "Multiplex Apartment"
[10] "Detached Bungalow"                     "Att/Row/Twnhouse Apartment"
  "Att/Row/Twnhouse 2-Storey"
[13] "Fourplex Apartment"                    "Shared Room Apartment"
  "Semi-Detached Other"
[16] "Detached 1 1/2 Storey"                 "Triplex Apartment"
  "Upper Level Apartment"
[19] "Detached Bungalow-Raised"              "Detached Apartment"
  "Lower Level 2 1/2 Storey"
[22] "Semi-Detached Bachelor/Studio"         "Detached Bungaloft"
  "Multiplex Bachelor/Studio"
[25] "Room 3-Storey"                         "Lower Level 1 1/2 Storey"
  "Detached Bachelor/Studio"
[28] "Semi-Detached Apartment"               "Lower Level 2-Storey"
  "Multiplex 3-Storey"
[31] "Duplex 2-Storey"                       "Semi-Detached Bungalow"
  "Lower Level Bungalow-Raised"
[34] "Upper Level Bachelor/Studio"           "Store W/Apt/Offc 2-Storey"
  "Other Apartment"
[37] "Condo Townhouse Stacked Townhse"       "Detached 2 1/2 Storey"
  "Condo Apt Bungalow"
[40] "Lower Level Apartment"                 "Condo Apt Bachelor/Studio"
  "Lower Level Bungalow"
[43] "Semi-Detached Bungalow-Raised"         "Detached Sidesplit 4"
  "Detached Backsplit 3"
[46] "Comm Element Condo Apartment"          "Detached 3-Storey"
  "Co-Ownership Apt Bachelor/Studio"
[49] "Semi-Detached 3-Storey"                "Lower Level Backsplit 4"
  "Detached Backsplit 4"
[52] "Semi-Detached 2 1/2 Storey"            "Triplex 2-Storey"
  "Duplex 2 1/2 Storey"
[55] "Comm Element Condo Multi-Level"        "Condo Apt Loft"
  "Condo Apt Multi-Level"
[58] "Condo Townhouse 2-Storey"              "Other Multi-Level"
  "Co-Op Apt Apartment"
[61] "Detached Other"                        "Duplex Bungalow"
  "Semi-Detached 1 1/2 Storey"
```

```
[64] "Upper Level 2-Storey"                    "Upper Level Backsplit 4"
  "Upper Level 3-Storey"
[67] "Triplex 1 1/2 Storey"                    "Condo Townhouse Apartment"
  "Condo Apt Stacked Townhse"
[70] "Condo Apt 2-Storey"                      "Duplex Apartment"
  "Att/Row/Twnhouse 2 1/2 Storey"
[73] "Detached Sidesplit 3"                    "Upper Level Other"
  "Co-Ownership Apt Apartment"
[76] "Multiplex 2-Storey"                      "Triplex 3-Storey"
  "Store W/Apt/Offc 3-Storey"
[79] "Co-Ownership Apt 2-Storey"               "Detached Backsplit 5"
  "Condo Apt Other"
[82] "Condo Townhouse Multi-Level"             "Duplex 3-Storey"
  "Other 2-Storey"
[85] "Comm Element Condo Stacked Townhse" "Fourplex 3-Storey"
  "Comm Element Condo Loft"
[88] "Fourplex 1 1/2 Storey"                   "Other Other"
  "Att/Row/Twnhouse Other"
[91] "Fourplex 2-Storey"                       "Store W/Apt/Offc Other"
  "Comm Element Condo Other"
[94] "Semi-Detached Backsplit 3"               "Detached Sidesplit 5"
  "Condo Apt Industrial Loft"
[97] "Comm Element Condo 2-Storey"             "Comm Element Condo 3-Storey"
```

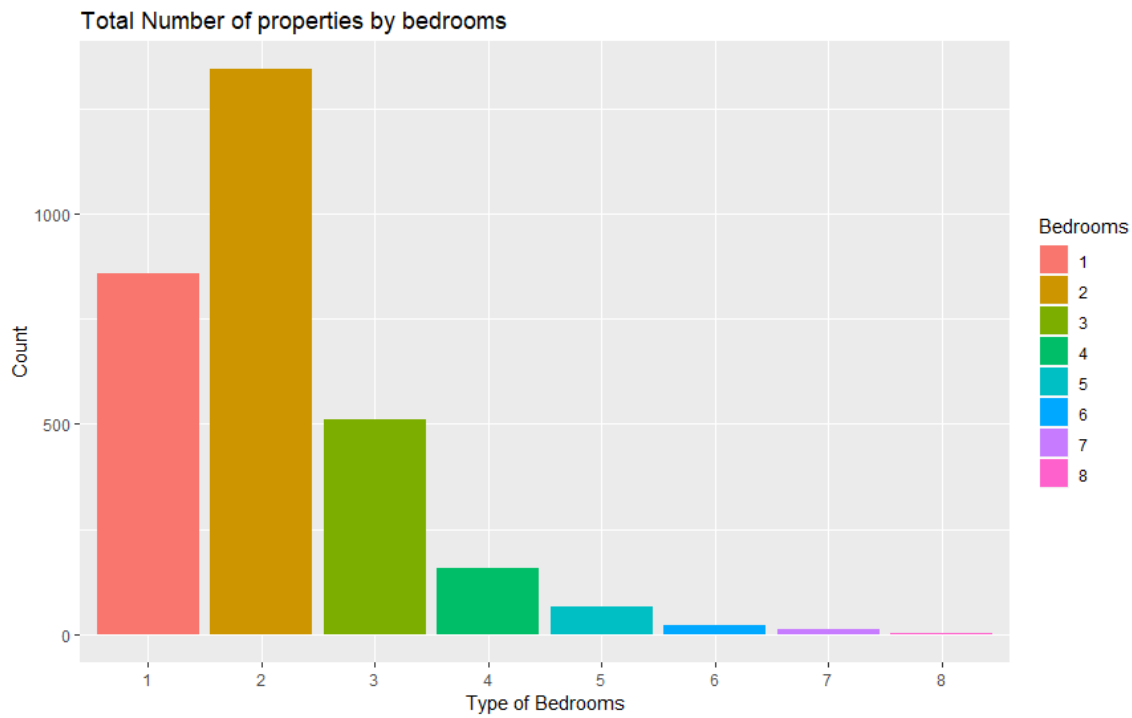
# Visualize the total number of properties by Type
ggplot(data=df)+geom_bar(aes(x=Type))+ggtitle("Total number of Properties by Type")



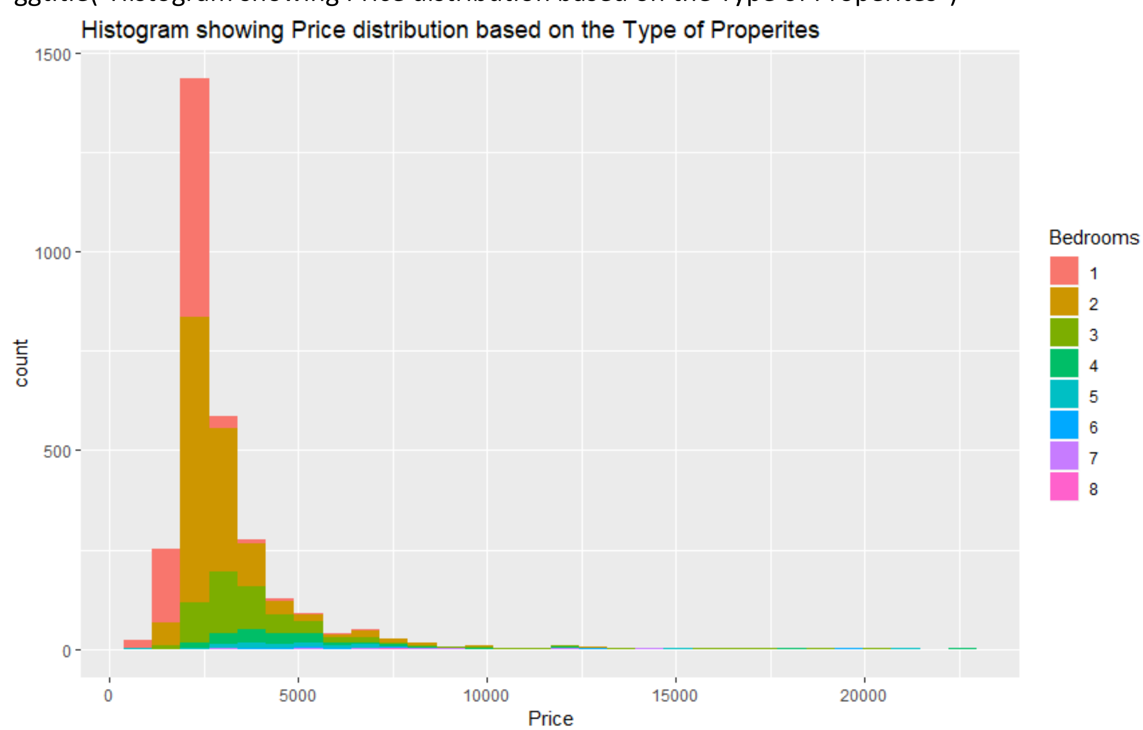
# Visualize the total number of propertiesby Type with fill "Bedrooms"
ggplot(data=df)+geom_bar(aes(x=Type,fill=Bedrooms))+ggtitle("Total number of properties by Type with Bedrooms Counts")



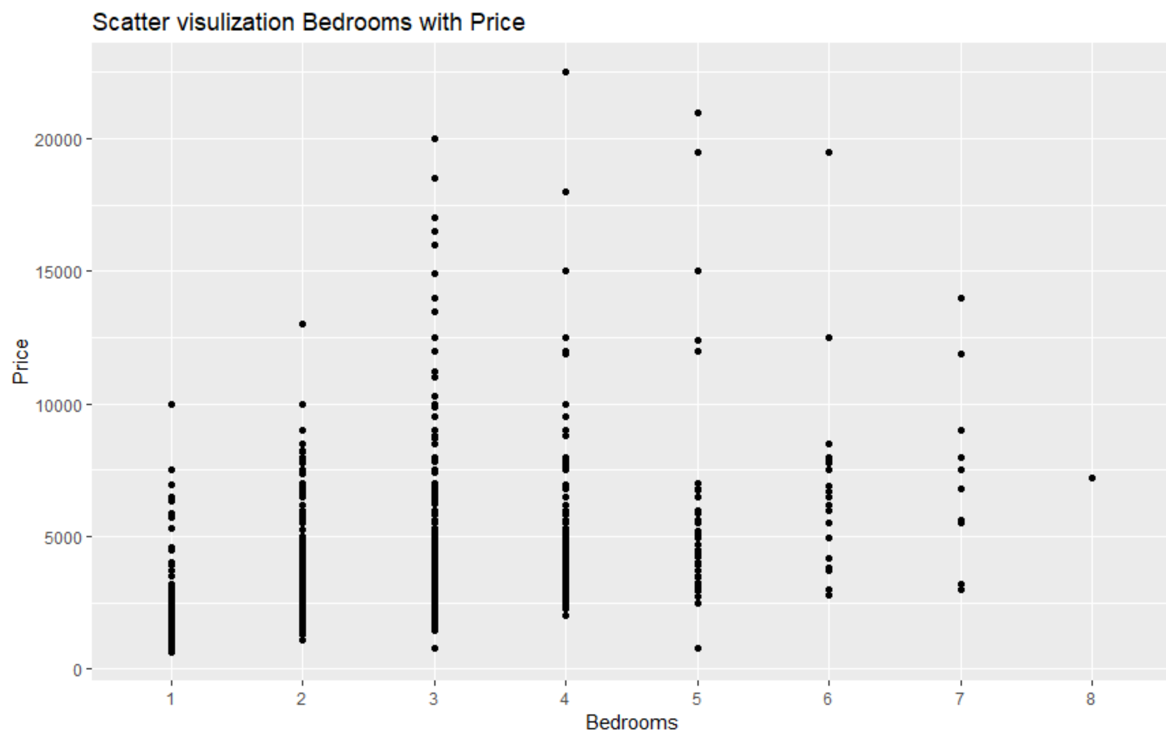
Visualize the number of properties by Bedrooms
ggplot(data = df, aes(x= Bedrooms, fill = Bedrooms))+
geom_bar()+ggtitle("Total Number of properties by bedrooms")+
xlab("Type of Bedrooms")+ ylab("Count")



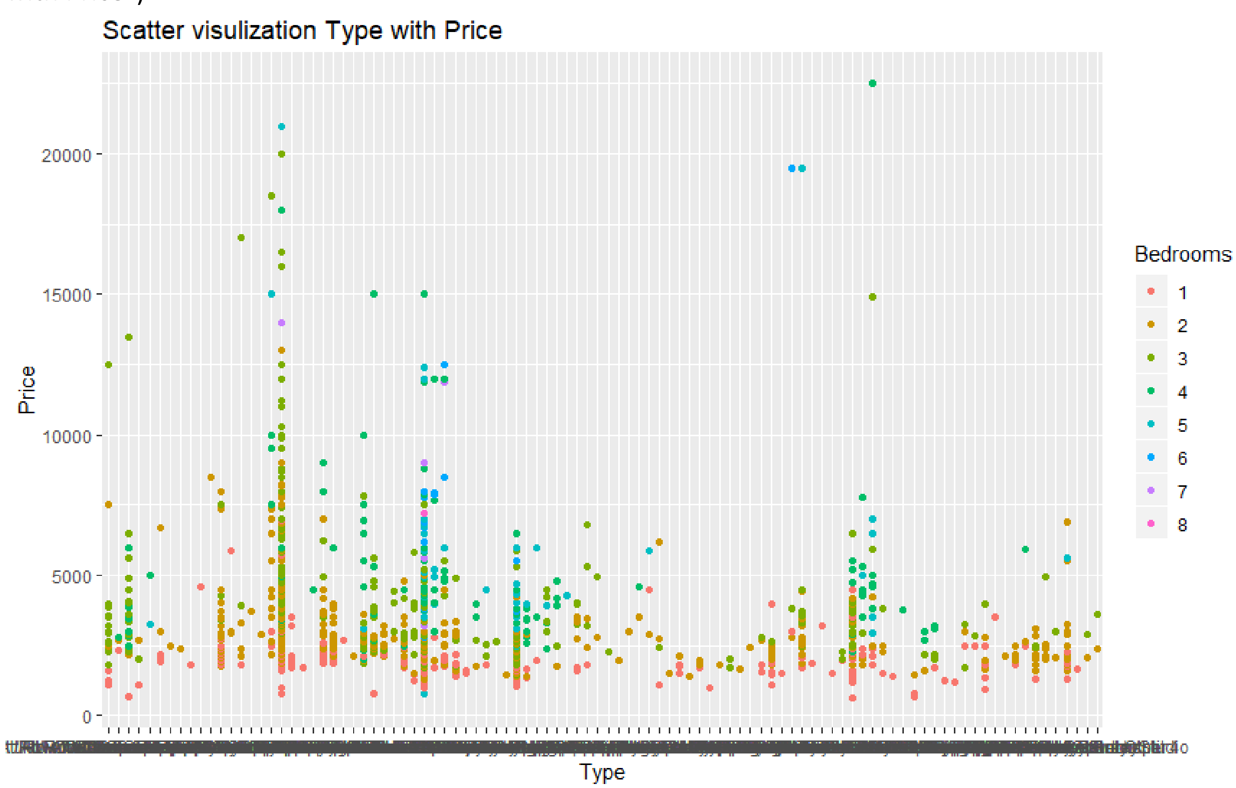

```
# Histogram visulize Price distribution based on type of properties
ggplot(data = df, aes(x= Price, bins=10, fill= Bedrooms))+
  geom_histogram()+
  ggtitle("Histogram showing Price distribution based on the Type of Properites")
```




```
# Visualize Price with Bedrooms
ggplot(data = df, aes(x=Bedrooms, y=Price))+geom_point()+ggtitle("Scatter visulization Bedrooms with Price")
```

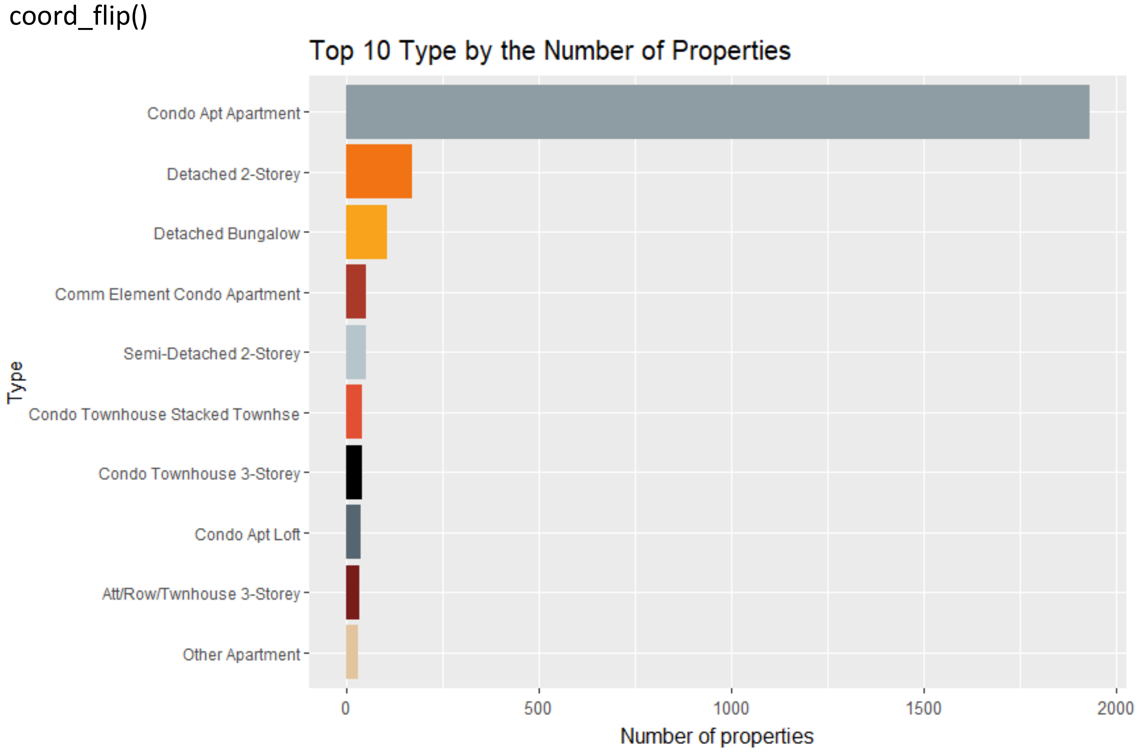



```
# Visualize Price with Type
ggplot(data=df)+geom_point(aes(x=Type,y=Price,color=Bedrooms))+ggtitle("Scatter visulization Type with Price")
```



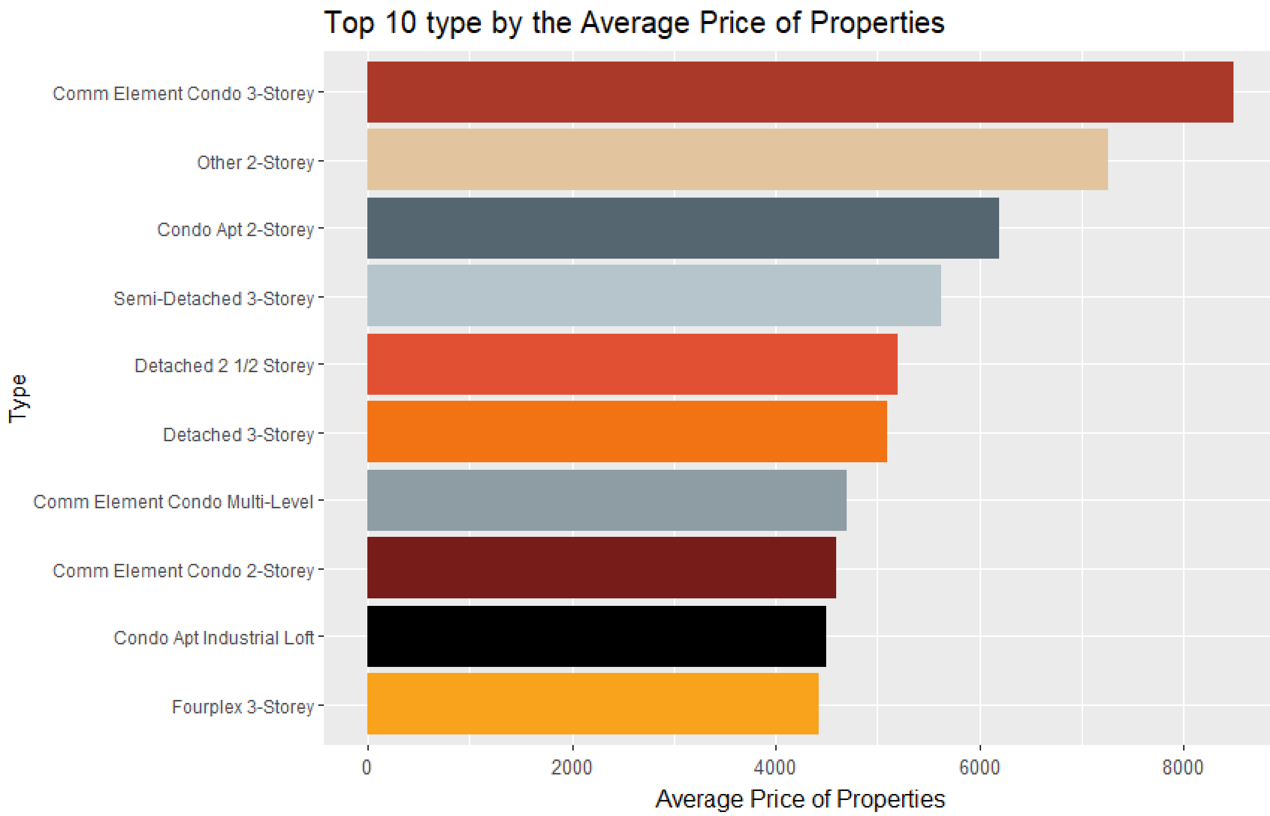

```
# Settings
```

```
mycolors <- c("#771C19", "#AA3929", "#8E9CA3", "#556670", "#000000",
         "#E25033", "#F27314", "#F8A31B", "#E2C59F", "#B6C5CC",
         "#99CCCC","#FFCC99")

mytheme <- theme(axis.text.x = element_text(angle = 90, size = 10, vjust = .4),
           plot.title = element_text(size = 15, vjust = 2),
           axis.title.x = element_text(size = 12, vjust = -.35))

mytheme2 <- theme(axis.text.x = element_text(size = 10, vjust = .4),
            plot.title = element_text(size = 15, vjust = 2),
            axis.title.x = element_text(size = 12, vjust = -.35))

# Top 10 Type by the Number of Properties
top10_type <- df %>% group_by(Type) %>%
  summarise(Number = n()) %>%
  arrange(desc(Number)) %>%
  head(10)
ggplot(top10_type, aes(reorder(Type, Number), Number, fill = Type))+
  geom_bar(stat = "identity")+mytheme2+
  theme(legend.position = "none")+
  labs(x = "Type", y = "Number of properties",
     title = "Top 10 Type by the Number of Properties")+
  scale_fill_manual(values = mycolors)+
  coord_flip()
```




```
#Top 10 Type by the Average Price of Properties
type_vs_price <- df[c("Type","Price")] %>%na.omit()
top10type_by_averprice <- type_vs_price %>%
  group_by(Type) %>%
```

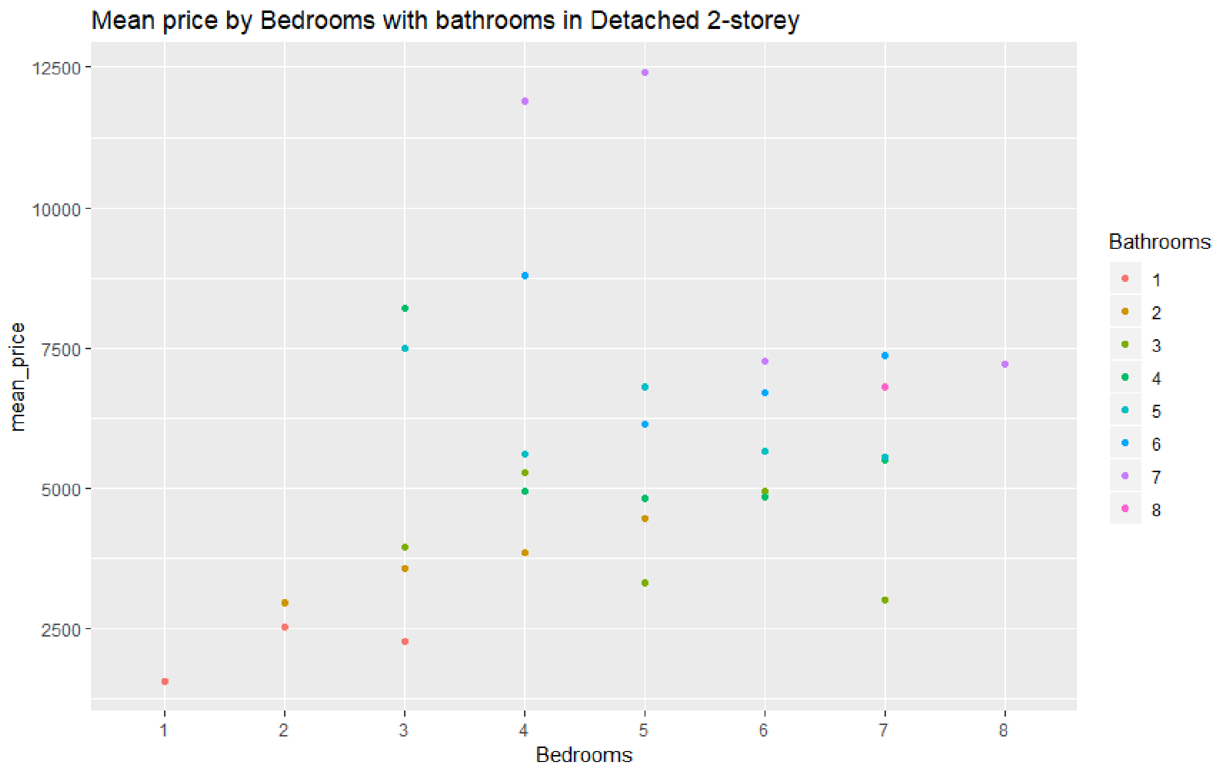
```
  summarise(Average = sum(Price)/n()) %>%
  arrange(desc(Average)) %>%
  head(10)
ggplot(top10type_by_averprice, aes(reorder(Type, Average), Average, fill = Type))+
  geom_bar(stat = "identity")+mytheme2+theme(legend.position = "none")+
  labs(x = "Type", y = "Average Price of Properties",
     title = "Top 10 type by the Average Price of Properties")+
  scale_fill_manual(values = mycolors)+
  coord_flip()
```




```
# Summarize the Price with Type, Bedrooms, Bathrooms and Look at Price Trend
df1<-df%>%
group_by(Type,Bedrooms,Bathrooms)%>%
summarize(mean_price=mean(Price,na.rm=TRUE))
write.csv(df1, file = "Summary_Type_Beds_Baths.csv",row.names=TRUE)
#see output "Summary_type_Beds_Baths.csv" file
```

```
# Look  the price trend in Type in "Detached 2-storey"
df1 %>%
filter(Type=="Detached 2-Storey")%>%
```

```
ggplot(aes(x=Bedrooms,y=mean_price))+geom_point(aes(color=Bathrooms))+geom_smooth(se=FALSE)
+ ggtitle("Mean price by Bedrooms with bathrooms in Detached 2-storey")
```




```
# Summarize the Price with Bedrooms, Bathrooms and Look at Price Trend
df2<-df1%>%
  group_by(Bedrooms,Bathrooms)%>%
  summarize(mean_price1=mean(mean_price,na.rm=TRUE))
write.csv(df, file = "Summary_Beds_Baths.csv",row.names=TRUE)
# see output "Summary_Beds_Baths.csv" file

# Summarize the Price with Bedrooms and Look at Price Trend
df3<-df1%>%
  group_by(Bedrooms)%>%
  summarize(mean_price2=mean(mean_price,na.rm=TRUE))
write.csv(df, file = "Summary_Beds.csv",row.names=TRUE)
#See output"Summary_Beds.csv"file


# Modelling Building
# Decision Tree and Random Forest
# Compare perfromance of a single decison tree and random forest with 500 trees towards predicitng
rental Price.

#Split Dataset into train(80%) and test(20%)
set.seed(12345)
```

```
d<-sample(x=nrow(df),size=nrow(df)*0.8)
tree_train<-df[d,]
tree_test<-df[-d,]
dim(tree_train)
dim(tree_test)
colnames(df)
sum(is.na(df))
```

```
> dim(tree_train)
[1] 2372    6
> dim(tree_test)
[1] 594   6
> colnames(df)
[1] "Id"        "Address"   "Bedrooms"  "Bathrooms" "Type"      "Price"
> sum(is.na(df))
[1] 0
```

Decision Tree

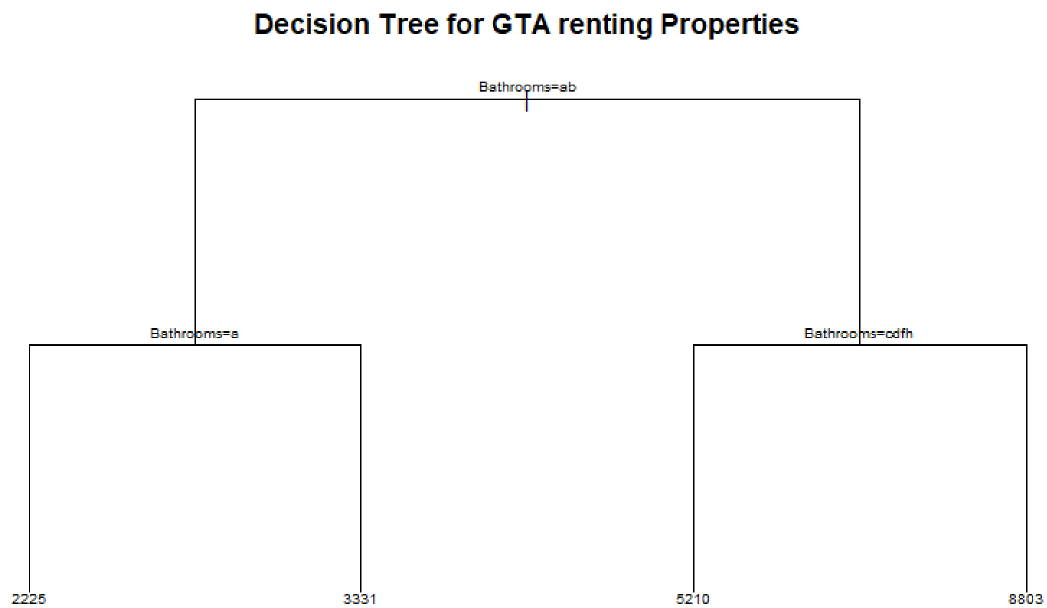
```
fit <- rpart(Price ~  Bedrooms + Bathrooms,data=tree_train)
printcp(fit)
rsq.rpart(fit)
summary(fit)
```

```
Regression tree:
rpart(formula = Price ~ Bedrooms + Bathrooms, data = tree_train)

Variables actually used in tree construction:
[1] Bathrooms

Root node error: 6696755621/2372 = 2823253

n= 2372

        CP nsplit rel error  xerror     xstd
1 0.298086      0   1.00000 1.00061 0.115491
2 0.091923      1   0.70191 0.70341 0.080966
3 0.049537      2   0.60999 0.61177 0.079485
4 0.010000      3   0.56045 0.57176 0.069251
> rsq.rpart(fit)

Regression tree:
rpart(formula = Price ~ Bedrooms + Bathrooms, data = tree_train)

Variables actually used in tree construction:
[1] Bathrooms

Root node error: 6696755621/2372 = 2823253

n= 2372

        CP nsplit rel error  xerror     xstd
1 0.298086      0   1.00000 1.00061 0.115491
2 0.091923      1   0.70191 0.70341 0.080966
3 0.049537      2   0.60999 0.61177 0.079485
```

```
4 0.010000      3   0.56045 0.57176 0.069251
> summary(fit)
Call:
rpart(formula = Price ~ Bedrooms + Bathrooms, data = tree_train)
  n= 2372

          CP nsplit rel error    xerror       xstd
1 0.29808612      0 1.0000000 1.0006110 0.11549110
2 0.09192337      1 0.7019139 0.7034103 0.08096581
3 0.04953719      2 0.6099905 0.6117734 0.07948481
4 0.01000000      3 0.5604533 0.5717643 0.06925078

Variable importance
Bathrooms  Bedrooms
       77        23

Node number 1: 2372 observations,    complexity param=0.2980861
  mean=2975.414, MSE=2823253
  left son=2 (2117 obs) right son=3 (255 obs)
  Primary splits:
      Bathrooms splits as  LLRRRRRR, improve=0.2980861, (0 missing)
      Bedrooms  splits as  LLRRRRRR, improve=0.1989460, (0 missing)
  Surrogate splits:
      Bedrooms splits as  LLLRRRRR, agree=0.929, adj=0.341, (0 split)

Node number 2: 2117 observations,    complexity param=0.09192337
  mean=2657.027, MSE=1012126
  left son=4 (1290 obs) right son=5 (827 obs)
  Primary splits:
      Bathrooms splits as  LR------, improve=0.2872997, (0 missing)
      Bedrooms  splits as  LRRRRR--, improve=0.1353624, (0 missing)
  Surrogate splits:
      Bedrooms splits as  LLRRRR--, agree=0.731, adj=0.312, (0 split)

Node number 3: 255 observations,    complexity param=0.04953719
  mean=5618.651, MSE=1.003089e+07
  left son=6 (226 obs) right son=7 (29 obs)
  Primary splits:
      Bathrooms splits as  --LLRLRL, improve=0.12969290, (0 missing)
      Bedrooms  splits as  LLLLRRRR, improve=0.01167821, (0 missing)
  Surrogate splits:
      Bedrooms splits as  LLLLLLLR, agree=0.89, adj=0.034, (0 split)

Node number 4: 1290 observations
  mean=2225.267, MSE=228694.2

Node number 5: 827 observations
  mean=3330.51, MSE=1489802

Node number 6: 226 observations
  mean=5210.075, MSE=7023888

Node number 7: 29 observations
  mean=8802.724, MSE=2.202548e+07
```

```
# plot a single decision tree
plot(fit, uniform=TRUE,main="Decision Tree for GTA renting Properties")
```

```
text(fit,use.n=TRUE,cex=.6)
# prune the tree
prune(fit,cp=0.0001)
```



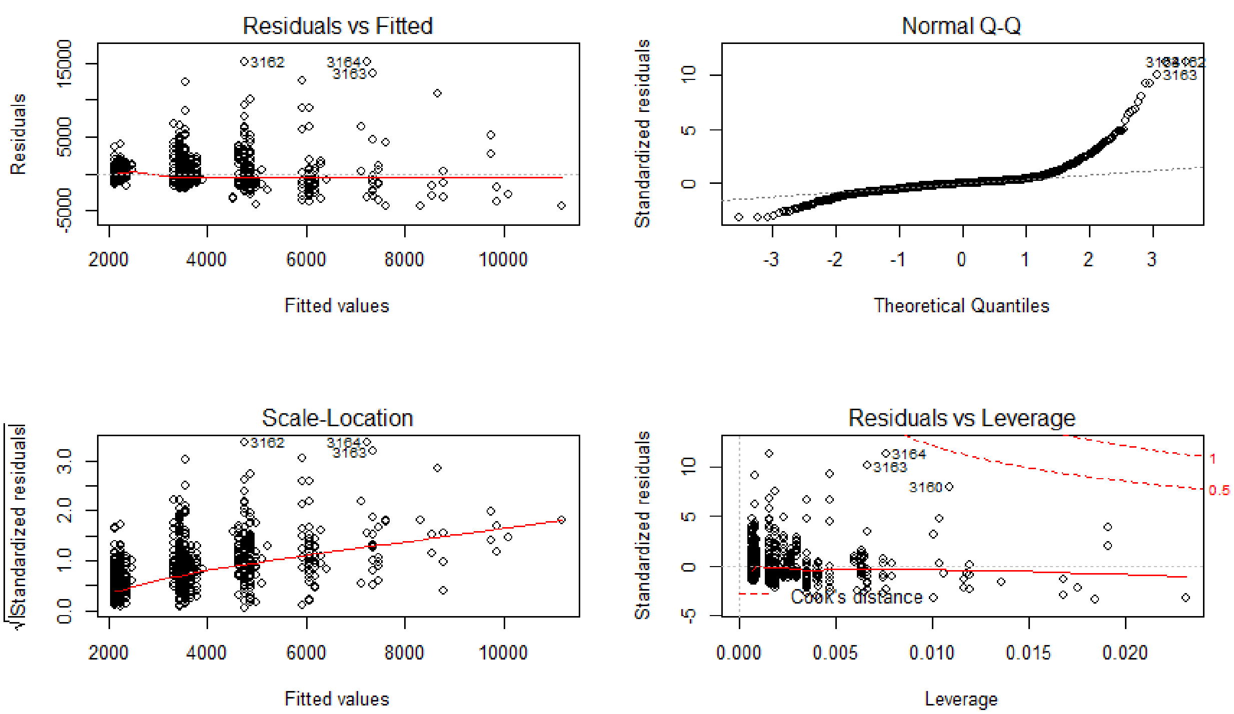

```
n= 2372

node), split, n, deviance, yval
      * denotes terminal node

1) root 2372 6696756000 2975.414
  2) Bathrooms=1,2 2117 2142670000 2657.027
    4) Bathrooms=1 1290  295015500 2225.267 *
    5) Bathrooms=2 827 1232066000 3330.510 *
  3) Bathrooms=3,4,5,6,7,8 255 2557876000 5618.651
    6) Bathrooms=3,4,6,8 226 1587399000 5210.075 *
    7) Bathrooms=5,7 29  638738800 8802.724 *

# Model Validation
# Calculating accuracy:rmse or mae
test_predictions<-predict(fit,tree_test)
rmse_decision_tree<-rmse(actual=tree_test$Price,predicted=test_predictions)
mae_decision_tree<-mae(actual=tree_test$Price,predicted=test_predictions)

# Random Forest:
model <- randomForest(Price ~ Bedrooms + Bathrooms,data = tree_train,ntree=50
0)
#Plot Variable of Importance
varImpPlot(model)
#Model Validation (calculating accuracy:rmse or mae)
summary(model)
```

```
forest_predictions <- predict(model, tree_test)
rmse_random_forest<-rmse(actual=tree_test$Price,predicted=forest_predictions)
mae_random_forest<-mae(actual=tree_test$Price,predicted=forest_predictions)

#Comparing the Errors (MAE)of decision tree and random forest
print(rmse_decision_tree)
print(rmse_random_forest)
print(mae_decision_tree)
print(mae_random_forest)
> #Comparing the Errors (MAE)of decision tree and random forest
> print(rmse_decision_tree)
[1] 1910.903
> print(rmse_random_forest)
[1] 1838.928
> print(mae_decision_tree)
[1] 882.7005
> print(mae_random_forest)
[1] 838.6105

# Conslusion: It can be concluded random forest performs better than a single
 decision tree.
```

```
# Regression Model to predict rental price
rent_regression <- select(df,-c(Id,Address,Type))
rent_regression$Bedrooms <- as.numeric(rent_regression$Bedrooms)
rent_regression$Bathrooms <- as.numeric(rent_regression$Bathrooms)

#Split dataset for Regression Model
set.seed(1)
d<-sample(x=nrow(rent_regression),size=nrow(df)*0.8)
regression_train<-rent_regression[d,]
regression_test<-rent_regression[-d,]
dim(regression_train)
dim(regression_test)
colnames(rent_regression)
str(rent_regression)

#Linear regression (adjusted R =0.4113 )
linear_model1 <- lm(Price ~., data = regression_train)
summary(linear_model1)

#set graphic output
par(mfrow=c(2,2))
# Create residual plots
plot(linear_model1)
```

```
> summary(linear_model1)

Call:
lm(formula = Price ~ ., data = regression_train)
```

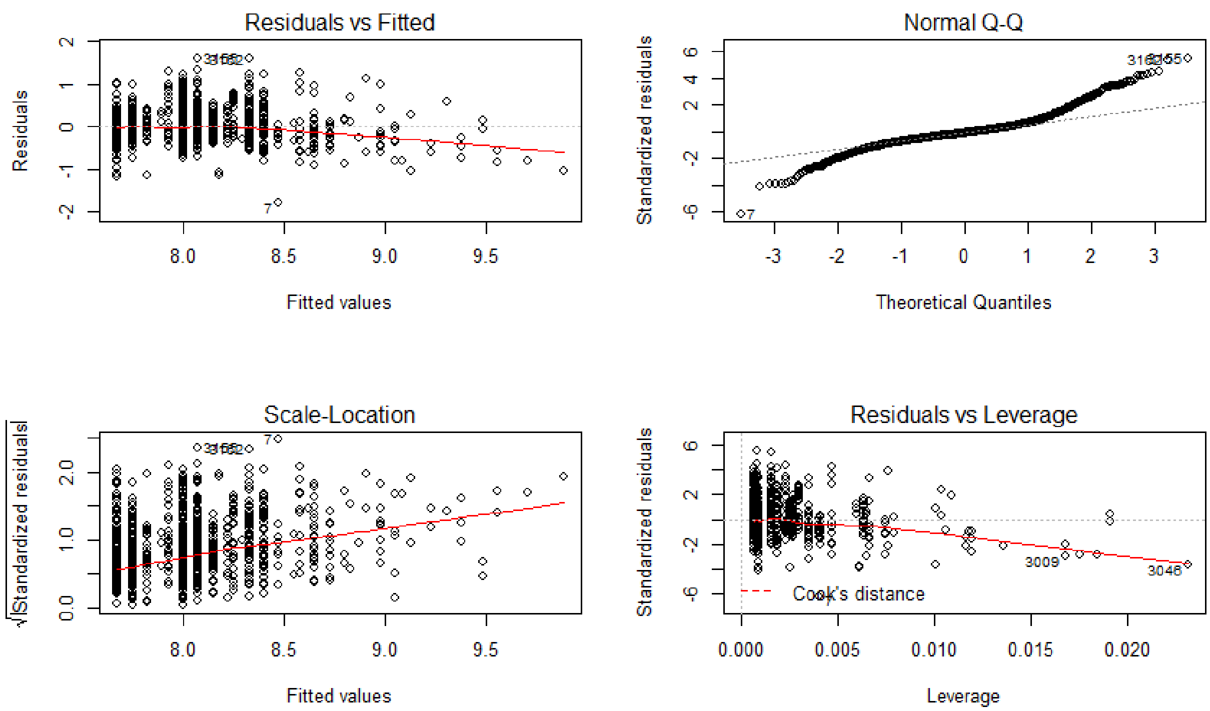
```
Residuals:
    Min      1Q  Median      3Q     Max 
-4406.4  -542.0  -105.7   213.6 15260.5 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   830.75      64.97  12.787   <2e-16 ***
Bedrooms      119.35      40.31   2.961   0.0031 ** 
Bathrooms    1186.27      47.39  25.033   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1355 on 2369 degrees of freedom
Multiple R-squared:  0.4118,	Adjusted R-squared:  0.4113 
F-statistic: 829.4 on 2 and 2369 DF,  p-value: < 2.2e-16
```




```
# To overcome heteroskedasiticity with building log(Price) (Adjusted R-squared:0.487 )
linear_model2 <- lm(log(Price)~., data = regression_train)
summary(linear_model2)
plot(linear_model2)
#(Adjusted R obtained =0.4870)
test_predictions<-predict(linear_model2,data = regression_test)
test_predictions<-exp(test_predictions)
rmse(actual=regression_test$Price,predicted=test_predictions)
```

```
> summary(linear_model2)

Call:
lm(formula = log(Price) ~ ., data = regression_train)
```

```
Residuals:
     Min       1Q   Median       3Q      Max 
-1.78924 -0.14093 -0.02291  0.10311  1.60711 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  7.34605    0.01389 528.679   <2e-16 ***
Bedrooms     0.07407    0.00862   8.593   <2e-16 ***
Bathrooms    0.25249    0.01013  24.912   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2897 on 2369 degrees of freedom
Multiple R-squared:  0.4874,	Adjusted R-squared:  0.487 
F-statistic:  1126 on 2 and 2369 DF,  p-value: < 2.2e-16
```




```
> rmse(actual=regression_test$Price,predicted=test_predictions)
[1] 2312.589
```

It shows linear_model2 is better than linear_model1. Linear_model 2 has Adjusted R-squared: 0.487, p-value: < 2.2e-16.

#The relationship shows price with bedrooms and bathrooms is
log(Price)=0.07407*number of Bedrooms +0.25247* number of Bathrooms +7.34605.

This shows that the number of Bedrooms has stronger positive relationships with the renting Price than the number of bathrooms.

Aim of this analysis is to answer a question of "Which ones are the best for investments?"

Finding "BEST" is hard and it is subjective matter.
Therefore, rather than concluding which ones are the best for investments,
it is much wiser to perform further research about the area Toronto
since in this analysis we are missing some potentially important variables
related to properties. It is possible that the properties have higher rental prices because of low crime rate, convenient transportation, and higher standard interior decorations of property etc.