ADS 506 Final Project

2023-11-04

```
library(readr)
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
house=read.csv("/Users/amyou/Desktop/ADS 506/ADS 506 Final Project/raw_sales.csv")
head(house)
```

```
##              datesold postcode  price propertyType bedrooms
## 1 2007-02-07 00:00:00     2607 525000        house        4
## 2 2007-02-27 00:00:00     2906 290000        house        3
## 3 2007-03-07 00:00:00     2905 328000        house        3
## 4 2007-03-09 00:00:00     2905 380000        house        4
## 5 2007-03-21 00:00:00     2906 310000        house        3
## 6 2007-04-04 00:00:00     2905 465000        house        4
```

```
#check for missing data and data inspection
sum(is.na(house))
```

```
## [1] 0
```

```
str(house)
```

```
## 'data.frame':    29580 obs. of  5 variables:
##  $ datesold    : chr  "2007-02-07 00:00:00" "2007-02-27 00:00:00" "2007-03-07 00:00:00" "2007-03-09 0
##  $ postcode    : int  2607 2906 2905 2905 2906 2905 2607 2606 2902 2906 ...
##  $ price       : int  525000 290000 328000 380000 310000 465000 399000 1530000 359000 320000 ...
##  $ propertyType: chr  "house" "house" "house" "house" ...
##  $ bedrooms    : int  4 3 3 4 3 4 3 4 3 3 ...
```

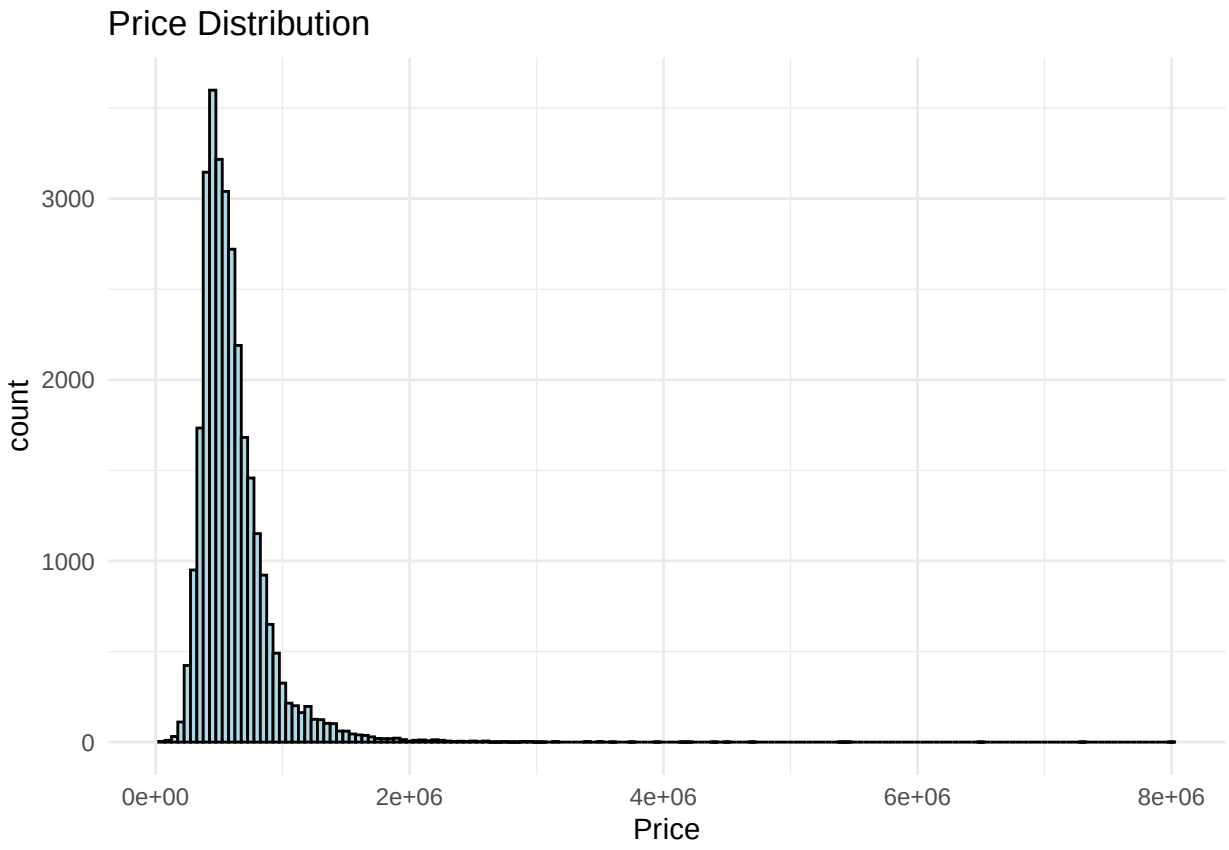
```
summary(house)
```

```
##    datesold            postcode        price          propertyType
##  Length:29580       Min.   :2600   Min.   :  56500   Length:29580
##  Class :character   1st Qu.:2607   1st Qu.: 440000   Class :character
##  Mode  :character   Median :2615   Median : 550000   Mode  :character
##                     Mean   :2730   Mean   : 609736
##                     3rd Qu.:2905   3rd Qu.: 705000
##                     Max.   :2914   Max.   :8000000
##     bedrooms
##  Min.   :0.00
##  1st Qu.:3.00
##  Median :3.00
##  Mean   :3.25
##  3rd Qu.:4.00
##  Max.   :5.00
```

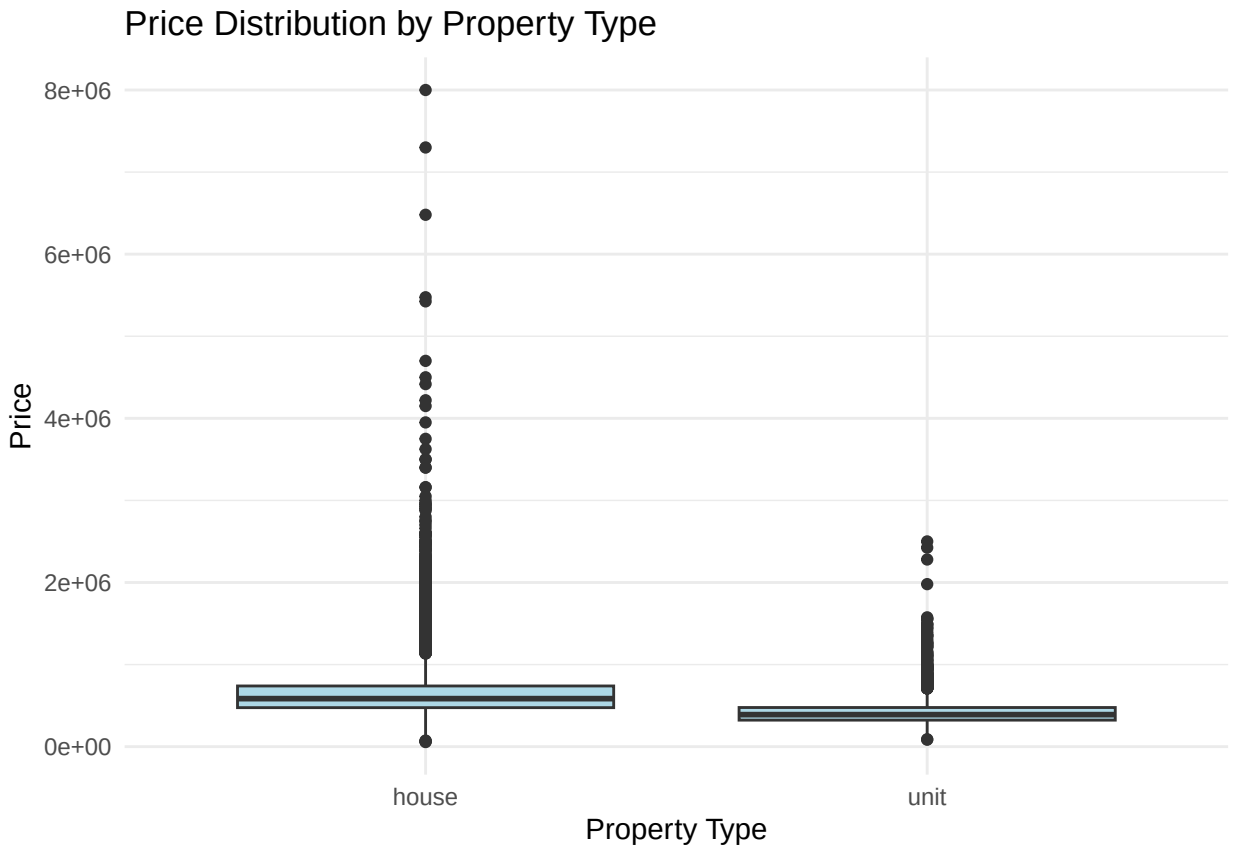
```r
#converting to proper time frame
house$datesold <- as.POSIXct(house$datesold, format = "%Y-%m-%d %H:%M:%S")

property_count <- table(house$propertyType)
```

```r
#time analysis
house<- house[order(house$datesold), ]

#time series plot
house$Month <- format(house$datesold, format = "%Y-%m")
monthly_counts <- aggregate(house$postcode, by = list(house$Month), FUN = length)
monthly_amounts <- aggregate(house$price, by = list(house$Month), FUN = sum)
```

```r
# Data Visualization
ggplot(house, aes(x = price)) +
  geom_histogram(binwidth = 50000, fill = "lightblue", color = "black") +
  labs(title = "Price Distribution", x = "Price") +
  theme_minimal()
```



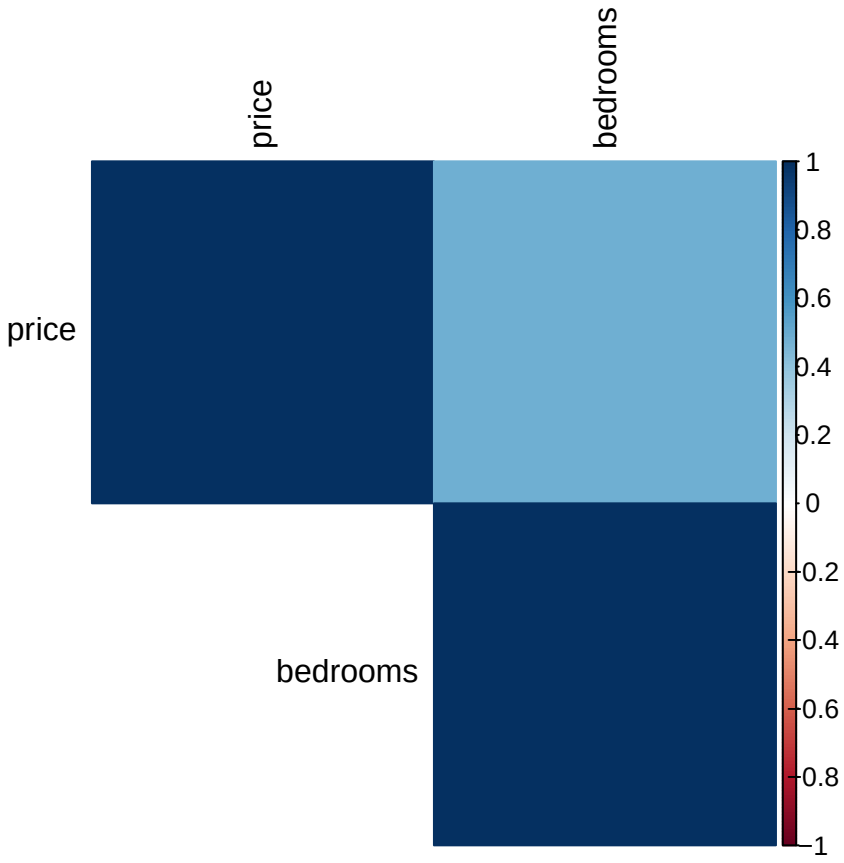

```r
ggplot(house, aes(x = propertyType, y = price)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Price Distribution by Property Type", x = "Property Type", y = "Price") +
  theme_minimal()
```




```
corr <- cor(house[, c("price", "bedrooms")])
corr
```

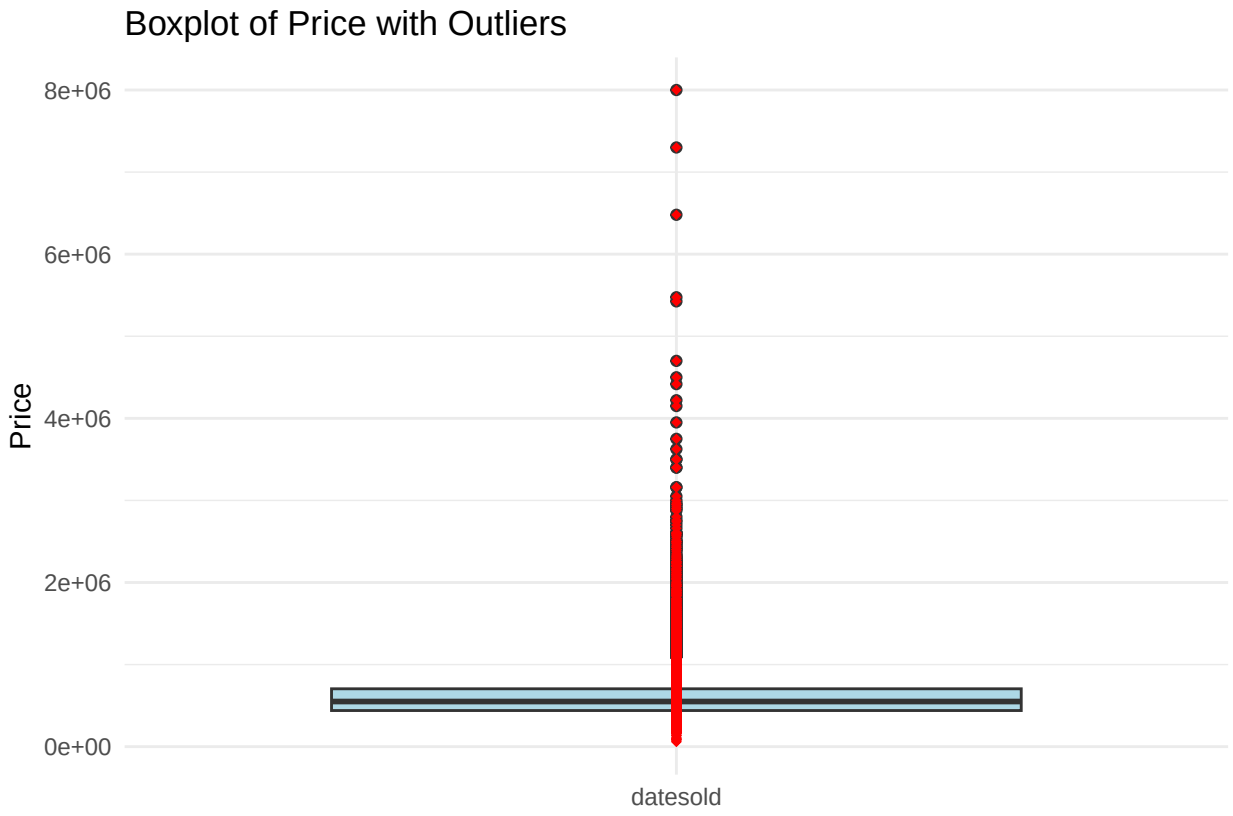
```
##              price  bedrooms
## price    1.0000000 0.4842117
## bedrooms 0.4842117 1.0000000
```

```
corrplot(corr, method = "color", type = "upper", tl.col = "black")
```




```
# detecting outliers
price_q1 <- quantile(house$price, 0.25)
price_q3 <- quantile(house$price, 0.75)
iqr <- price_q3 - price_q1
lower_bound <- price_q1 - 1.5 * iqr
upper_bound <- price_q3 + 1.5 * iqr
outliers <- house[house$price < lower_bound | house$price > upper_bound, ]

ggplot(house, aes(x = "datesold", y = price)) +
  geom_boxplot(fill = "lightblue") +
  geom_point(house = house[house$price < lower_bound | house$price > upper_bound, ],
             aes(x = 1, y = price), color = "red", shape = 18) +
  labs(title = "Boxplot of Price with Outliers", x = "", y = "Price") +
  theme_minimal() +
  scale_x_discrete()
```

```
## Warning in geom_point(house = house[house$price < lower_bound | house$price > :
## Ignoring unknown parameters: `house`
```



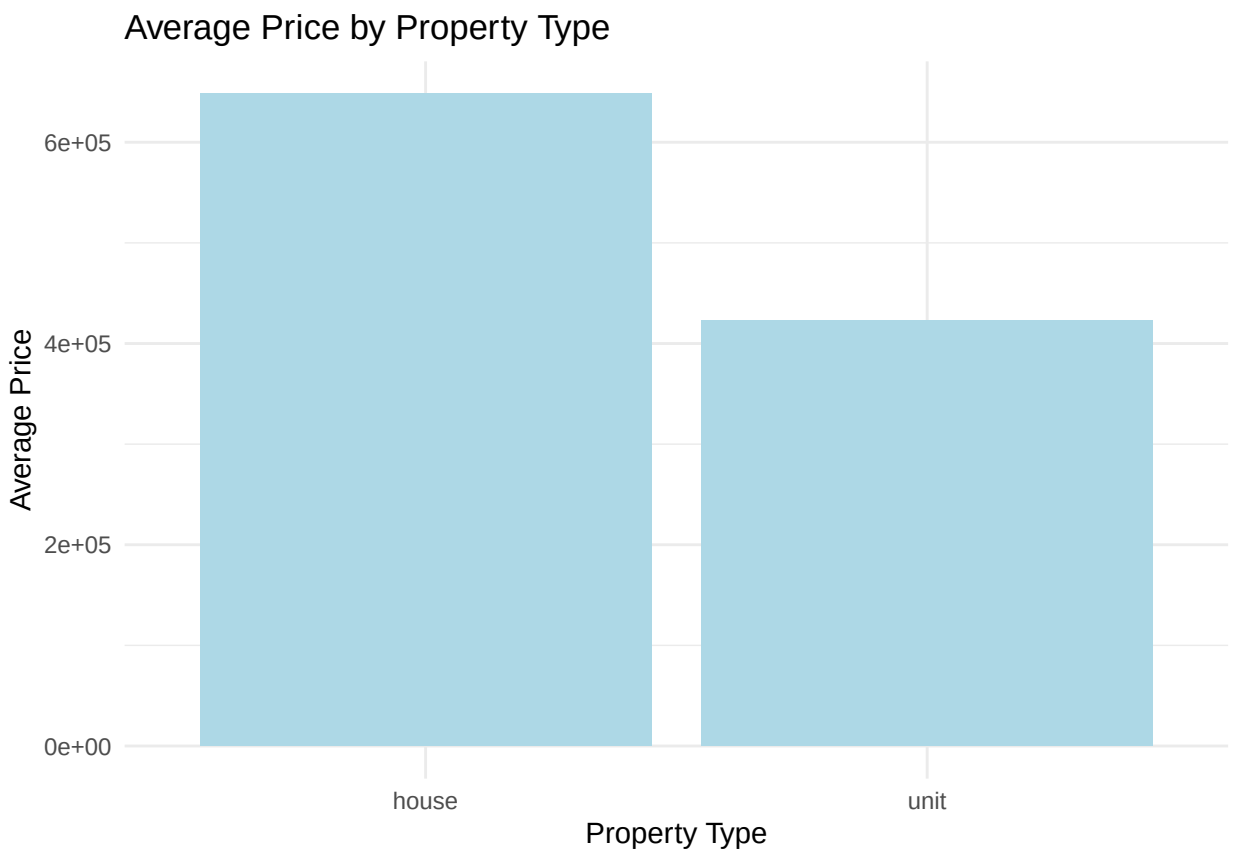

```
# Property Type Analysis
property_type_prices <- aggregate(house$price, by = list(house$propertyType), FUN = mean)
property_type_bedrooms <- aggregate(house$bedrooms, by = list(house$propertyType), FUN = mean)

print(property_type_bedrooms)
```

```
##   Group.1        x
## 1   house 3.539467
## 2    unit 1.837510
```

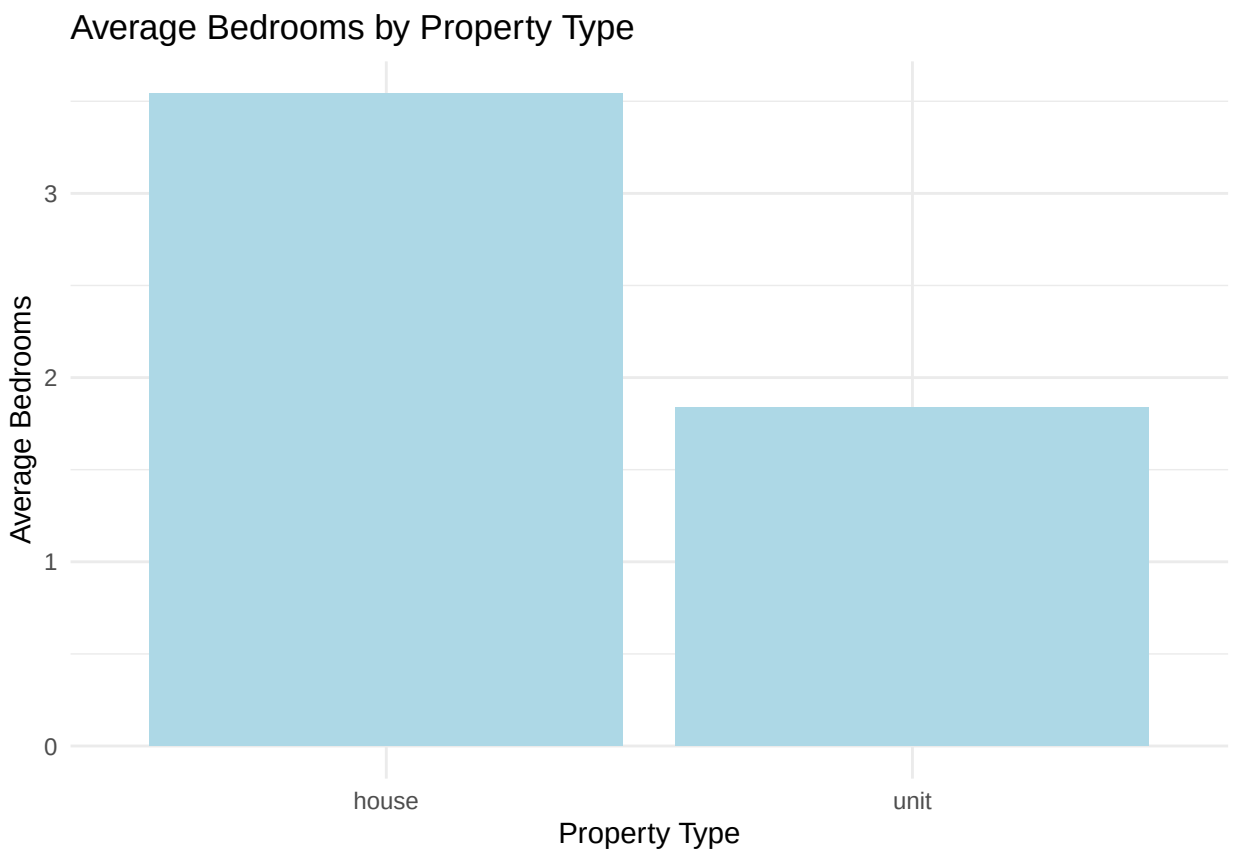
```
print(property_type_prices)
```

```
##   Group.1        x
## 1   house 647956.1
## 2    unit 423106.6
```

```
# Bar plot for price
ggplot(property_type_prices, aes(x = Group.1, y = x)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  labs(title = "Average Price by Property Type", x = "Property Type", y = "Average Price") +
  theme_minimal()
```




```
# Bar plot for bedrooms
ggplot(property_type_bedrooms, aes(x = Group.1, y = x)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  labs(title = "Average Bedrooms by Property Type", x = "Property Type", y = "Average Bedrooms") +
  theme_minimal()
```




```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(fpp2)
```

```
## -- Attaching packages --------------------------------------------- fpp2 2.5 --
```

```
## v fma       2.5     v expsmooth 2.3
```

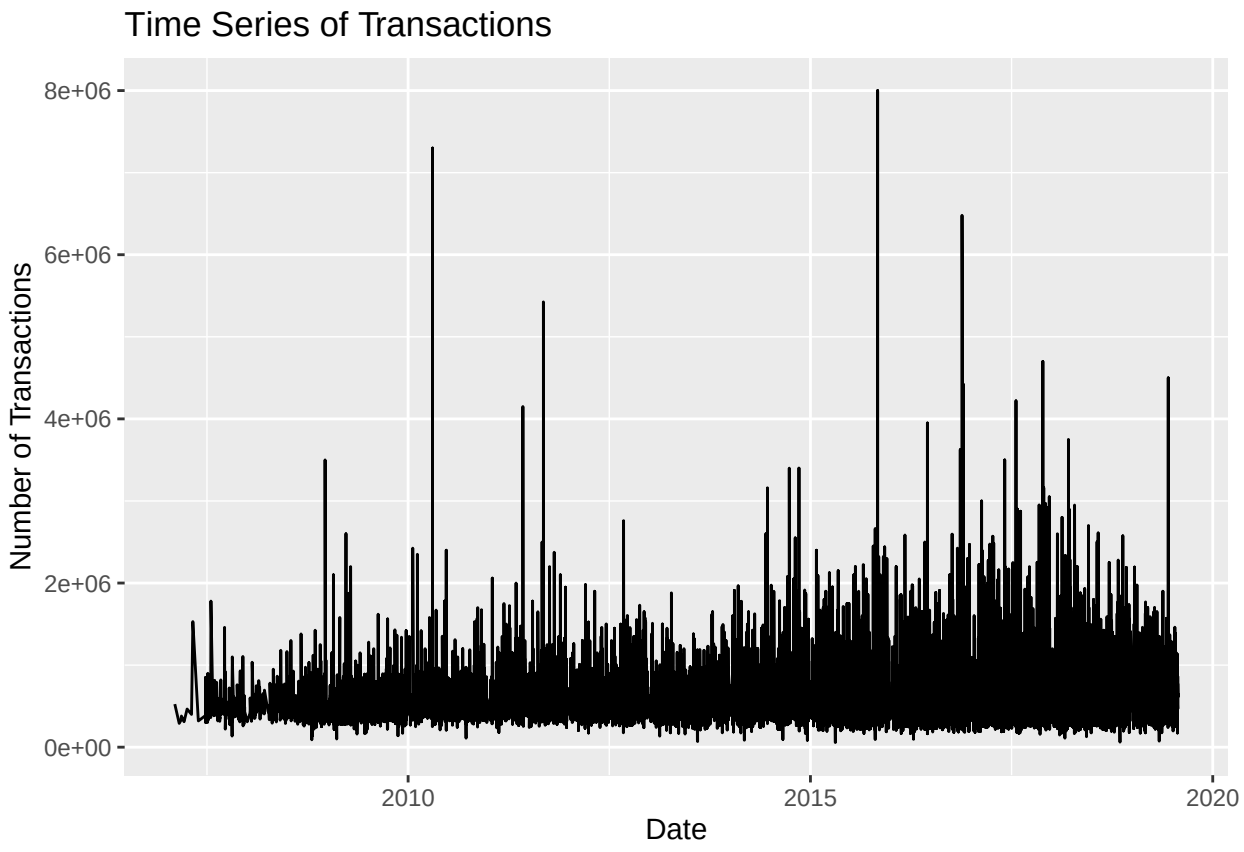
```
##
```

```
library(xts)
house_ts <- xts(house$price, order.by = house$datesold)

# Create a time series plot using autoplot
library(ggfortify)
```

```
## Registered S3 methods overwritten by 'ggfortify':
##   method                 from
```

```
##   autoplot.Arima         forecast
##   autoplot.acf           forecast
##   autoplot.ar            forecast
##   autoplot.bats          forecast
##   autoplot.decomposed.ts forecast
##   autoplot.ets           forecast
##   autoplot.forecast      forecast
##   autoplot.stl           forecast
##   autoplot.ts            forecast
##   fitted.ar              forecast
##   fortify.ts             forecast
##   residuals.ar           forecast
```

```r
autoplot(house_ts) +
  ggtitle("Time Series of Transactions") +
  xlab("Date") +
  ylab("Number of Transactions")
```




```r
correlation_postal_price <- cor(house$postcode, house$price)
corr_post_bed <- cor(house$bedrooms, house$postcode)
# Print the correlation result
cat("Correlation between PostalCode and Price: ", correlation_postal_price, "\n")
```

```
## Correlation between PostalCode and Price:  -0.1505482
```

```r
cat("Correlation between Postalcode and Bedroom: ", corr_post_bed, "\n")
```

```
## Correlation between Postalcode and Bedroom:  0.2257614
```