Stats 15 Project

Team 13: Wanyue Dong, Michelle Pang, Liaohan Wang

10/31/2022

Section 1 - Introduction

1.1 - Motivating Questions and Scope of Analysis

As one of the biggest cities in China, Beijing currently houses a population of 21.33 million residents. With the rapid rate of globalization, the movement of people selling and buying their property has risen as well. However, with the help of Lianjia, buyers and sellers are able to view and list properties with ease. In this project, we seek to answer the following motivating question: **what affects the price per square meters of a property?**

1.2 - Background on Lianjia

Basic Structure: Lianjia is a Chinese real-estate company that allows buyers and sellers to view and list properties all over China on the Lianjia website. In our data set, we narrowed down our listings to those that were traded in 2017 in the city of Beijing. This means that any property that was listed or sold in 2017 through Lianjia will be included in this dataset. We found our data set on a website called Kaggle which is a data science company which provides a variety of public data sets. Before cleaning the data, our data set had 318,851 listings and 26 variables.

How a Seller can Post a Listing: To post a listing on Lianjia, the owner or the real estate agent of the property will have to follow the guidelines and steps provided by filling out the necessary information regarding the property such as price of listing, number of bathrooms, square footage of the entire property and more listed below in the explanatory variables. They will also need to provide accompanying photographs of the property.

How a Buyer can Purchase a Listing: To purchase a listing on Lianjia, the buyer can simply just reach out to the contact number or email of the real estate angent provided to arrange a viewing. After viewing the property and possible negotiations, both parties will agree on a price for the transaction and legal paperwork will be organized by the real estate agency to facilitate the payment process and finally legalize the change of ownership of the property.

1.3 - Variable Explanation

Explanatory Variables

1. ***DOM***: (*numerical*) The active days on market. The number of days since the property is posted until it is sold and removed from the website.
2. ***followers***: (*numerical*) The number of people who follow the transaction by bookmarking the property to revisit later on. This indicates the popularity of a property compared to the other listings. Once the transaction is completed, users can not follow it anymore. It is counted from the property is listed until the transaction is completed.

3. ***livingRoom***: (*numerical*) The number of bedrooms.
4. ***drawingRoom***: (*numerical*) The number of living rooms.
5. ***kitchen***: (*numerical*) The number of kitchens.
6. ***bathRoom***: (*numerical*) The number of bathrooms.
7. ***floor***: (*numerical*) The total number of floors the building has. This usually also refers to the height of the building.
8. ***constructionTime***: (*categorical*) The year the building was constructed.
9. ***renovationCondition***: (*categorical*) The quality of the renovation with 4 being the best and 1 being the worst.
10. ***buildingStructure***: (*categorical*) The material that is used the most to construct the building. More specifically, the main material used to build the exterior of the building. In this case, 1 refers to the material being unknown, 2 refers to the material being a mix of more than 3 distinguishable materials, 3 refers to brick and wood, 4 refers to brick and concrete, 5 refers to steel and 6 refers to steel-concrete composite.
11. ***ladderRatio***: (*categorical*) The proportion between the number of elevators or ladders that particular floor has to the number of households who live on the same floor. It describes how many ladders a household have on average. In this case, 0.5 means two households share 1 elevator/ladder. The higher the number, the more elevators or ladders a household can use.
12. ***elevator***: (*categorical*) The presence of an elevator or not. In this case, 1 refers to the presence of an elevator in the building and 0 refers to the absence of an elevator in the building.
13. ***fiveYearsProperty***: (*categorical*) Refers to whether or not the previous owner has owned the property for less than five years. In this case, 1 refers to the property being owned by the previous owner for less than five years and 0 refers to the property being owned by the previous owner for more than five years. In Beijing, if a house/apartment has been owned for 5 years, the tax will be less than owning it for less than five years.
14. ***subway***: (*categorical*) Based on the knowledge of the seller posting the listing, this variable shows if a subway is located near the property. In this case, 1 refers to the presence of a subway nearby and 0 refers to the absence of a subway nearby. Whether or not a property is near a subway is self reported by the seller.

Figure 1: Beijing District Map

15. ***district***: (*categorical*) There are 13 main districts in Beijing and each number from 1 to 13 refers

to a different district in which the property lies in. In this case, 1 refers to DongCheng, 2 refers to FengTai, 3 refers to YiZhuang (a sub-region of DaXing, we will consider it as DaXing in the following analysis), 4 refers to DaXing, 5 refers to FangShan, 6 refers to ChangPing, 7 refers to ChaoYang, 8 refers to HaiDian, 9 refers to ShiJingShan, 10 refers to XiCheng, 11 refers to TongZhou, 12 refers to MenTouGou, and 13 refers to ShunYi.

As shown in the map above, there are a total of 16 districts in Beijing, but we mainly focuses on the listed 12 districts which are shown on the map as yellow, red, and purple. Green districts are not included in this Lianjia dataset. Yellow and red parts are considered the core districts. In other words, "the city". They contains the main functions of Beijing in political, cultural and economic ways. The yellow parts, DongCheng and XiCheng contain the most history of the city. They have a relative small area but most ancient buildings preserved. Thus, there are actually not that much area could be used for residency nor new structures. Among these red parts, ChaoYang and HaiDian are a bit special. ChaoYang is the largest among the core districts. It has the majority of foreign embassies and it is also the business center. HaiDian is famous for education. It contains the most education institutions. Purple parts are considered suburb areas where airports are located. Green parts are excluded from our data because these are relative rural that are mostly made up of mountains, Great Walls, manors for travelers, etc.

Response Variable

1. ***price***: (*integer*) The price per square foot of the property (price = totalPrice / square). To ensure that the price is a fair comparison across all the properties listed, we decided to compare the price per square foot of all properties instead of totalPrice. Since some properties may be bigger than others, comparing price per square foot provides us with a better understanding of how valuable a property is.

Section 2: Data Cleaning

```r
# loading libraries
library(lubridate)
library(dplyr)
library(VIM)
library(stringr)
library(ggplot2)
library(naniar)
library(waffle)
library(tidyverse)
```

```r
# the following code is used to import dataset that includes Chinese character
data <- read.csv("new.csv", fileEncoding = "GBK", encoding = "GBK")
```

2.1 Cleaning Trade Time Variable

The `tradeTime` variable is in the format of y-m-d, we wanted to look at year, month, and day separately. Therefore, we created new variables `year`, `month`, `day` based on `tradeTime`.
Since the current data has 318851 rows, we decided to use part of the data where the trade time was in 2017.

```r
data2017 <- data %>%
  mutate(year = year(tradeTime), month = month.name[month(tradeTime)], day = day(tradeTime)) %>%
  filter(year == 2017)
```

2.2 Cleaning Floor Variable

The `floor` variable two piece of information. The first part, which is in Chinese character, informs the floor range of the apartment/house. The second part, which is a number, tells us the total number of floors the building has. Therefore, we need to separate these two information into two new variables (`totalFloor` and `floorRange`) to better analyze them. At the same time, we converted Chinese characters into English.

I hide code for this section because the file doesn't knit with Chinese characters in the code.

2.3 Cleaning variable format

Some variables such as living room and bedroom should be numerical variable, while other variables such as subway and elevator should be categorical variables. We also recoded the district variable into district names, which will be easier to see for later analysis. Moreover, the living room in this dataset should be bedroom and drawing room in this dataset should be living room. Finally, we recoded some categorical variables such as subway and elevator to have text in the values to see them clearly.

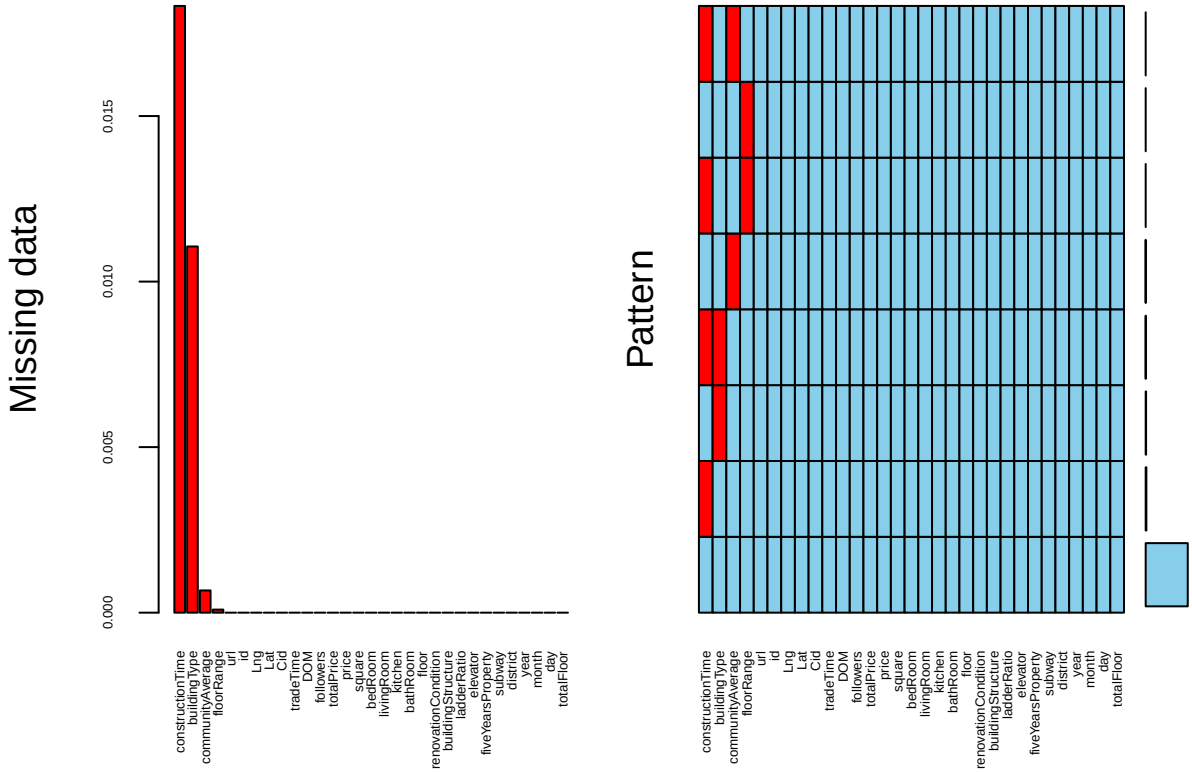
```
data2017 <- data2017 %>%
  rename(livingRoom = drawingRoom, bedRoom = livingRoom) %>%
  mutate(constructionTime = as.numeric(constructionTime),
         livingRoom = as.numeric(livingRoom),
         bedRoom = as.numeric(bedRoom),
         bathRoom = as.numeric(bathRoom),
         kitchen = as.numeric(kitchen),
         subway = as.factor(subway),
         elevator = as.factor(elevator),
         fiveYearsProperty = as.factor(fiveYearsProperty),
         district = recode(district, "1" = "DongCheng",
                           "2" = "FengTai", "3" = "DaXing", "4" = "DaXing",
                           "5" = "FangShan", "6" = "ChangPing", "7" = "ChaoYang",
                           "8" = "HaiDian", "9" = "ShiJingShan", "10" = "XiCheng",
                           "11" = "TongZhou", "12" = "MenTouGou", "13" = "ShunYi"),
         subway = recode(subway, "0" = "not_near_subway", "1" = "near_subway"),
         elevator = recode(elevator, "0" = "no", "1" = "yes"),
         fiveYearsProperty = recode(fiveYearsProperty, "0" = "no", "1" = "yes"),
         renovationCondition = recode(renovationCondition, "1" = "other", "2" = "rough",
                                      "3" = "simplicity", "4" = "hardcover"),
         buildingStructure = recode(buildingStructure, "1" = "other", "2" = "mixed",
                                    "3" = "brick&wood", "4" = "brick&concrete",
                                    "5" = "steel", "6" = "steel-concrete"))
```

2.4 Missing Values

```
table(is.na(data2017))
```

```
##
##   FALSE    TRUE
## 1338424    1303
```

```r
na_plot <- aggr(data2017, col=c('skyblue','red'), numbers=TRUE, sortVars=TRUE,
                labels=names(data2017), cex.axis=.4, gap=3, ylab=c("Missing data","Pattern"))
```




```
##
##  Variables sorted by number of missings:
##             Variable        Count
##     constructionTime 1.832612e-02
##         buildingType 1.106046e-02
##     communityAverage 6.710322e-04
##           floorRange 9.255617e-05
##                  url 0.000000e+00
##                   id 0.000000e+00
##                  Lng 0.000000e+00
##                  Lat 0.000000e+00
##                  Cid 0.000000e+00
##            tradeTime 0.000000e+00
##                  DOM 0.000000e+00
##            followers 0.000000e+00
##           totalPrice 0.000000e+00
##                price 0.000000e+00
##               square 0.000000e+00
##              bedRoom 0.000000e+00
##            livingRoom 0.000000e+00
##               kitchen 0.000000e+00
##              bathRoom 0.000000e+00
```

```
##                floor 0.000000e+00
##  renovationCondition 0.000000e+00
##    buildingStructure 0.000000e+00
##          ladderRatio 0.000000e+00
##             elevator 0.000000e+00
##    fiveYearsProperty 0.000000e+00
##               subway 0.000000e+00
##             district 0.000000e+00
##                 year 0.000000e+00
##                month 0.000000e+00
##                  day 0.000000e+00
##           totalFloor 0.000000e+00
```

Our data had 1303 missing values. All missing values are in variables construction time, building type, community average, and floor range. The above graph shows the proportion of missing values in each variable with the highest being 0.018 for construction time.
We decided to remove all missing values for two reasons. First, 1303 missing values is a relatively small amount of data compared to 43217 observations we have. Second, we did not plan to use the variable buildingType and community average, which had a large portion of missing values.

```
data2017 <- na.omit(data2017)
```

Now we have 42710 observations.

2.5 Remove unnecessary variables

The final step of data cleaning is removing variables that we do not plan to use.

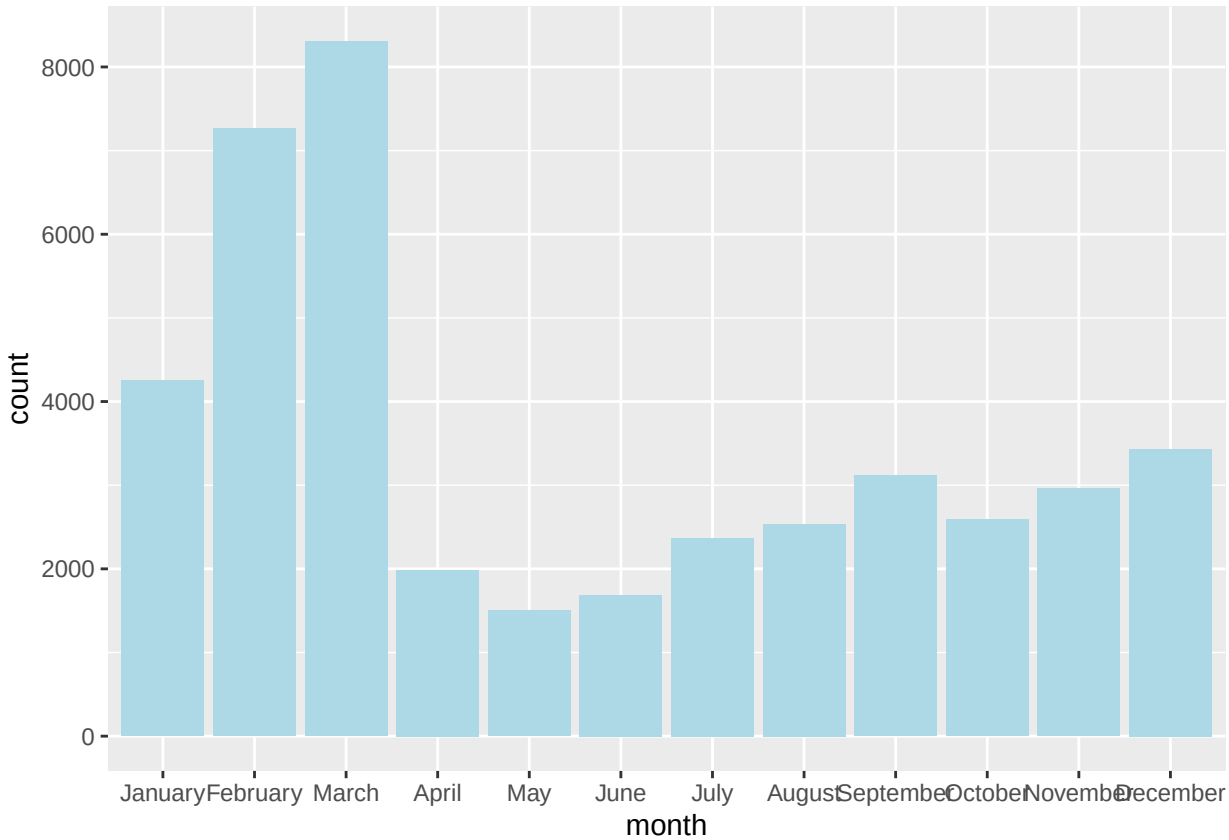
```
data2017 <- data2017 %>%
  select(-c(url, Lng, Lat, Cid, tradeTime, floor, buildingType, communityAverage, year))
```

Section 3: Exploratory Analysis

3.1 Distribution of variables and outliers

3.1.1 Month

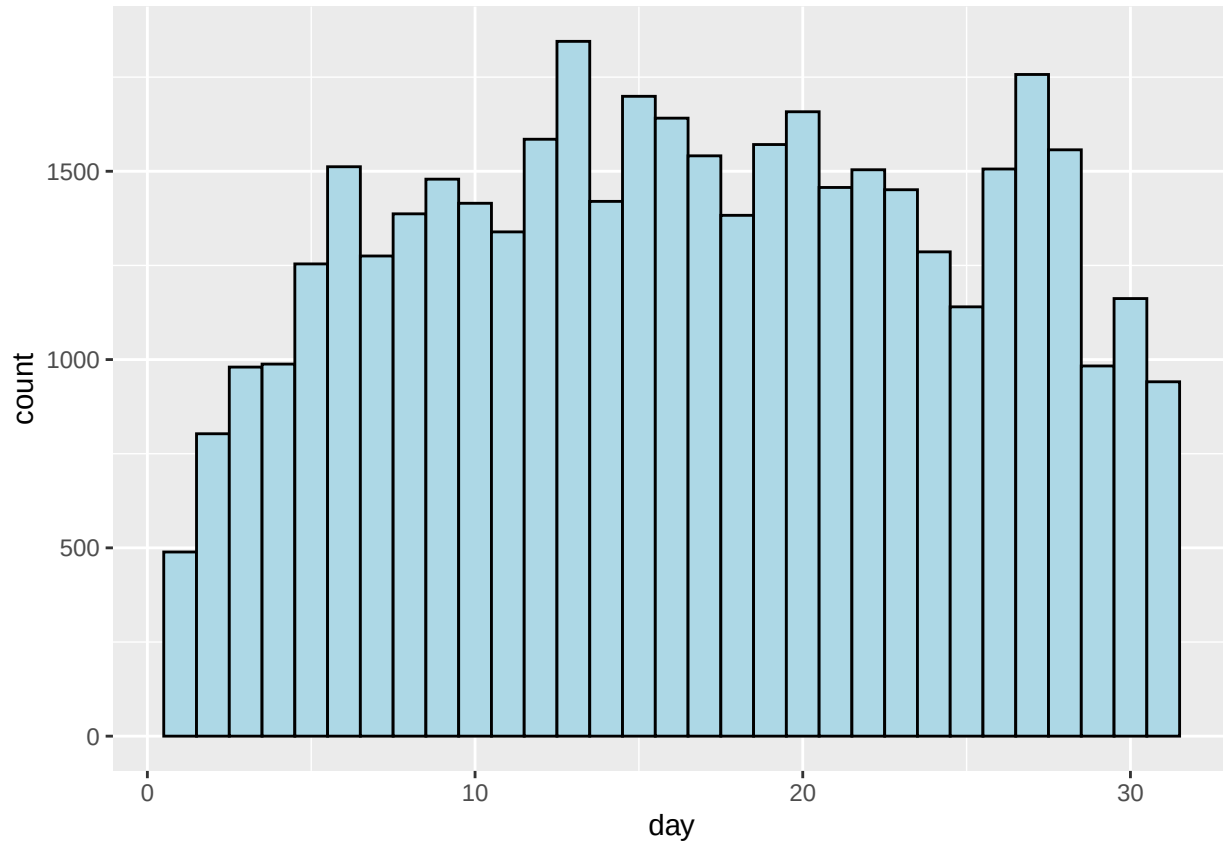
```
ggplot(data2017, aes(x = month)) +
  geom_bar(fill = "light blue") +
  scale_x_discrete(limits = month.name)
```



The first three months of year 2017 had the most trading of houses, with March being the highest number of trading, more than 8000 tradings. Apirl, May, and June each only had less than 1/4 of the number of trading compared to March. The amount of transactions for the rest of the year remained below 4000. One possible explanation for this trend is that Chinese people like to buy houses in the beginning of the year as a way of starting a fresh year.

3.1.2 Day

```
ggplot(data2017, aes(x = day)) +
  geom_histogram(fill = "light blue", color = "black", bins = 31)
```




```r
quantile(data2017$day)
```

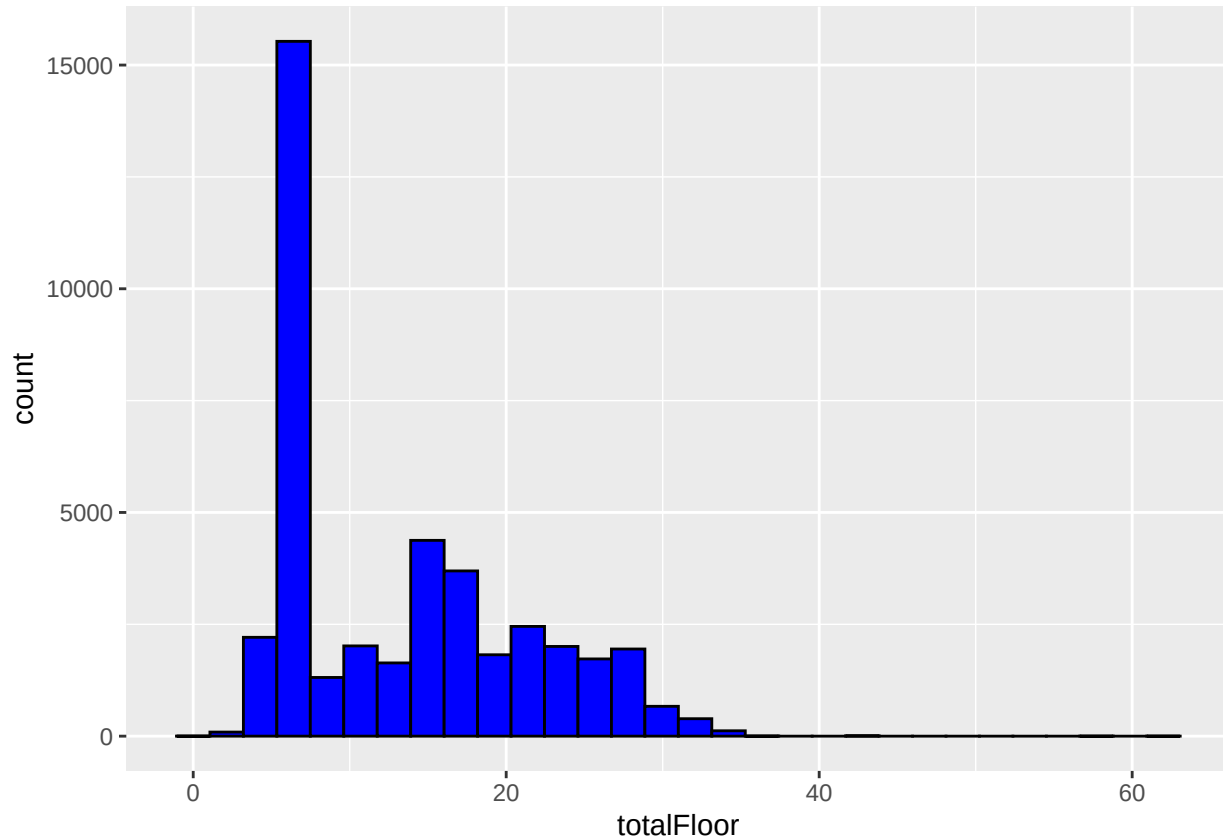
```
##   0%  25%  50%  75% 100%
##    1   10   16   23   31
```

Number of transactions of a day is uniform. There isn't a unique day of the month that people liked to buy houses, but people relatively like to buy houses in the middle of the month. Few trades happened in the beginning of the month. Approximately 25% of the houses are sold on and before 10th. 50% of the houses are sold on or before 17th. 25% of the houses are sold on or after 24th.

3.1.3 Total Floor

```r
ggplot(data2017, aes(x = totalFloor)) + geom_histogram(color = "black", fill = "blue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
quantile(data2017$totalFloor, c(0, .025, .25, .5, .75, .975, 1))
```

```
##    0%  2.5%   25%   50%   75% 97.5%  100%
##     1     5     6    11    20    29    63
```

```
data2017 %>% arrange(desc(totalFloor)) %>% select(id, totalFloor, floorRange) %>% head()
```

```
##             id totalFloor floorRange
## 1 101101127418         63        low
## 2 101101690389         57     medium
## 3 101091696453         42        low
## 4 101100641538         42        low
## 5 101100768834         42       high
## 6 101100791635         42     medium
```

```
data2017 <- data2017 %>%
  filter(totalFloor < 60)
```

The distribution of total floor is unimodel and skewed to the right. The median number of floors is 6. 75% of the buildings less than or equal to 6 floors. Only 2.5% of the buildings had more than 18 floors. Some high numbers could be an error in data recording. The url for the apartment located in a building with 63 floors is invalid; however, we checked the apartment located in the building with 57 floors using url and, surprisingly, the building does exist and is called Yu Jin Tai. Therefore, we only removed the transaction with total floor of 63.

3.1.4 Floor Range

```
data2017$floorRange <- factor(data2017$floorRange,
                  levels = c("bottom", "low", "medium", "high", "top"))
summary_table <- data2017 %>% group_by(floorRange) %>%
  summarise(total = n()) %>%
  mutate(percent = round(100 * total / sum(total)))
counts <- summary_table$percent
names(counts) <- summary_table$floorRange
print(counts)
```

```
## bottom    low medium   high    top
##      8     20     39     22     11
```

```
waffle(counts)
```

Apartments located in the middle of a building were traded the most in 2017. Bottom and top both have lowest percentage, while low and high have relatively middle percentage.

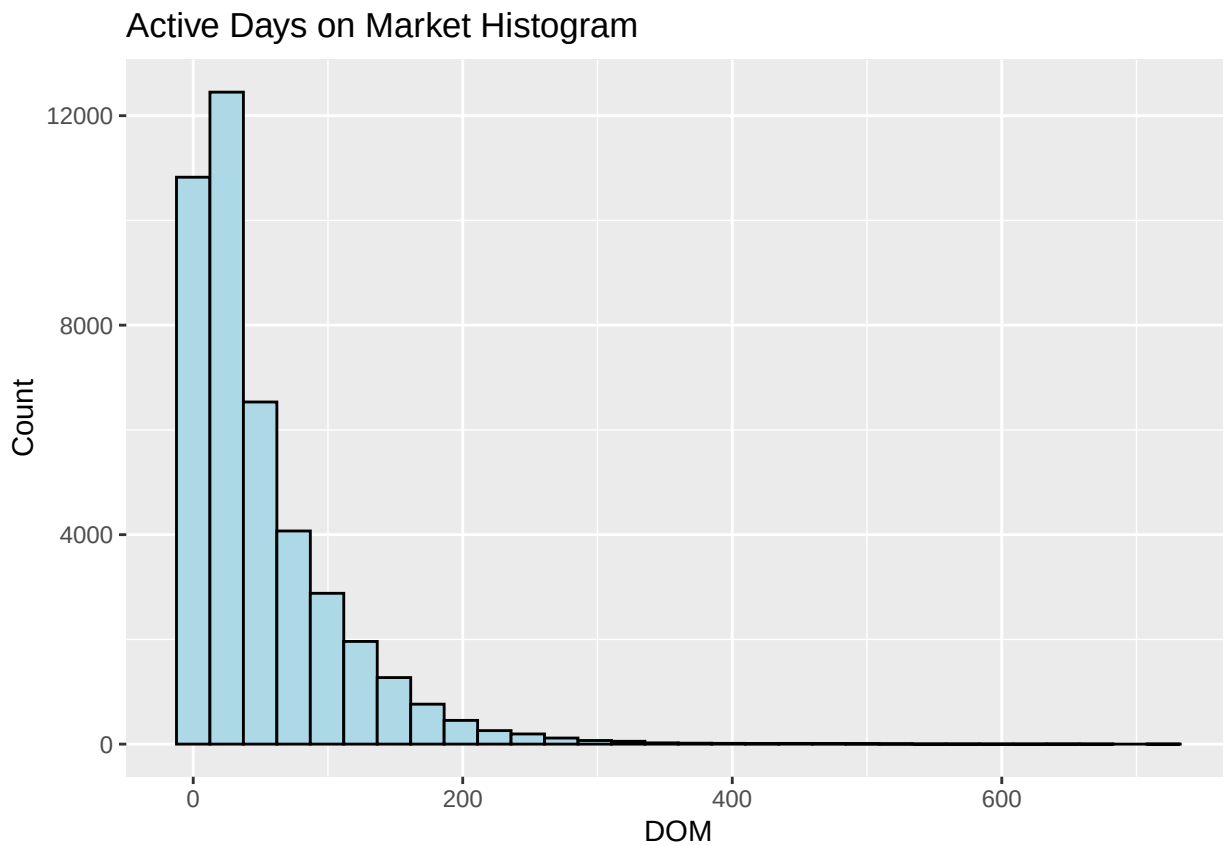
3.1.5 DOM

The distribution of number of days on market is unimodal and heavily skewed to the right. The mean number of days on market is around 51 and the median is 31 days. The difference between mean and median shows that there are more houses with relative longer days on market. Approximately 2.5% of values are

below 2. Approximately 2.5% of values are above 196. The longest day on market is 721. Whereas there are total of 474 houses have less or equal to 1 day on market until sold.

```
ggplot(data2017, aes(x=DOM))+
  geom_histogram(fill='lightblue', color='black') +
  xlab("DOM")+ylab("Count")+ggtitle("Active Days on Market Histogram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
mean(data2017$DOM)
```

```
## [1] 50.858
```

```
quantile(data2017$DOM, c(0, 0.025, 0.25, 0.5, 0.75, 0.975, 1))
```

```
##    0%  2.5%   25%   50%   75% 97.5%  100%
##     1     2    12    31    71   196   721
```

```
data2017 %>%
  filter(DOM<=1) %>%
  nrow()  # number of houses have less than 1 day on market
```

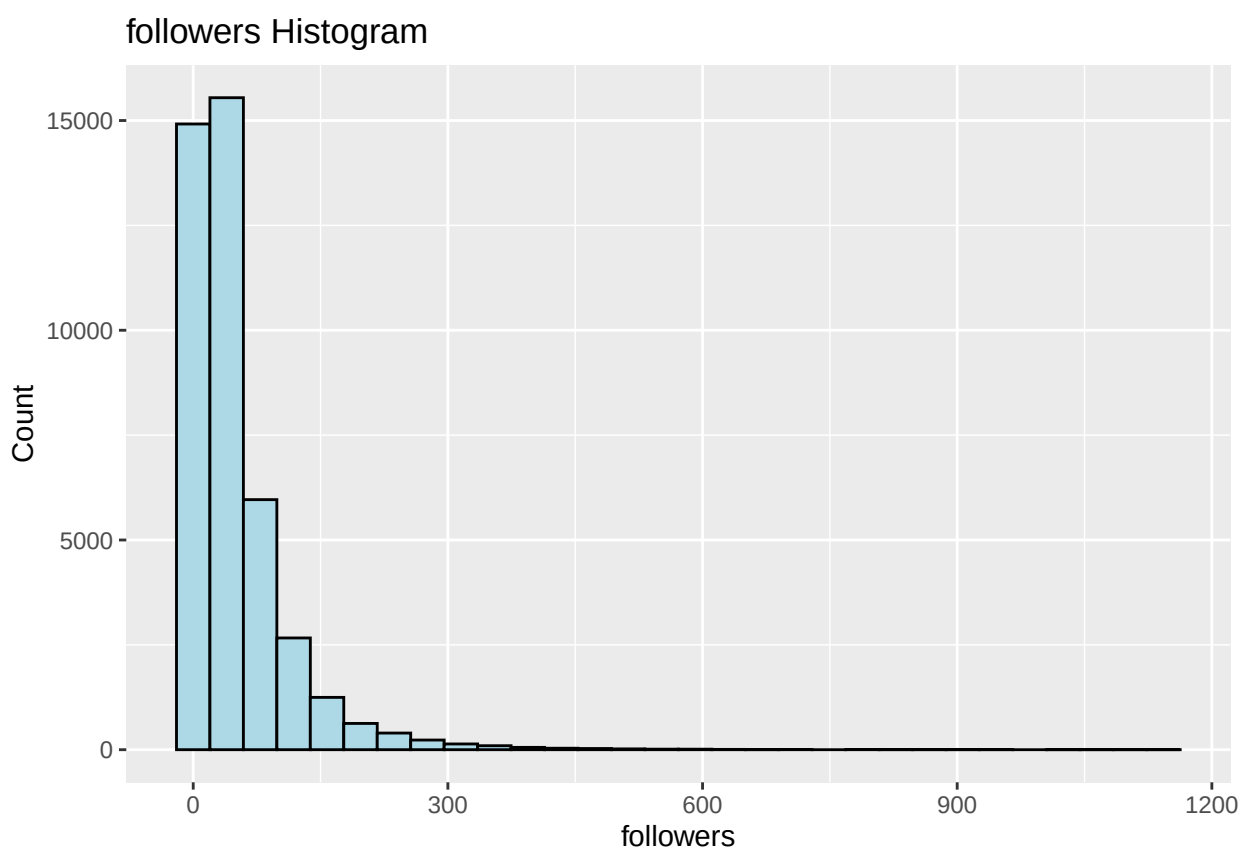
```
## [1] 474
```

3.1.6 followers

The distribution of number of followers is unimodal and heavily skews right. The mean and median number of followers are all around 49 people. Approximately 2.5% of values are below or equal to 0. However, the highest 2.5% are between 228 and 1045, which suggests that there are a few significantly popular houses the great amounts of followers. There are 2021 houses with 0 follower and also a total of 5 houses with more than 1000 followers. Among houses with more than 1000 followers, 1 house has significant low total price compare with others. However, it also has relative small squares and located in the suburb, which makes this reasonable.

```
ggplot(data2017, aes(x=followers))+
  geom_histogram(fill='lightblue', color='black') +
  xlab("followers")+ylab("Count")+ggtitle("followers Histogram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
mean(data2017$followers)
```

```
## [1] 49.41241
```

```
quantile(data2017$followers, c(0, 0.025, 0.25, 0.5, 0.75, 0.975, 1))
```

```
##    0%  2.5%   25%   50%   75% 97.5%  100%
##     0     0    11    31    64   216  1143
```

```r
table(data2017$followers==0) # 2021 houses with 0 follower
```

```
##
## FALSE  TRUE
## 36514  5493
```

```r
data2017 %>% filter(followers>1000)
```

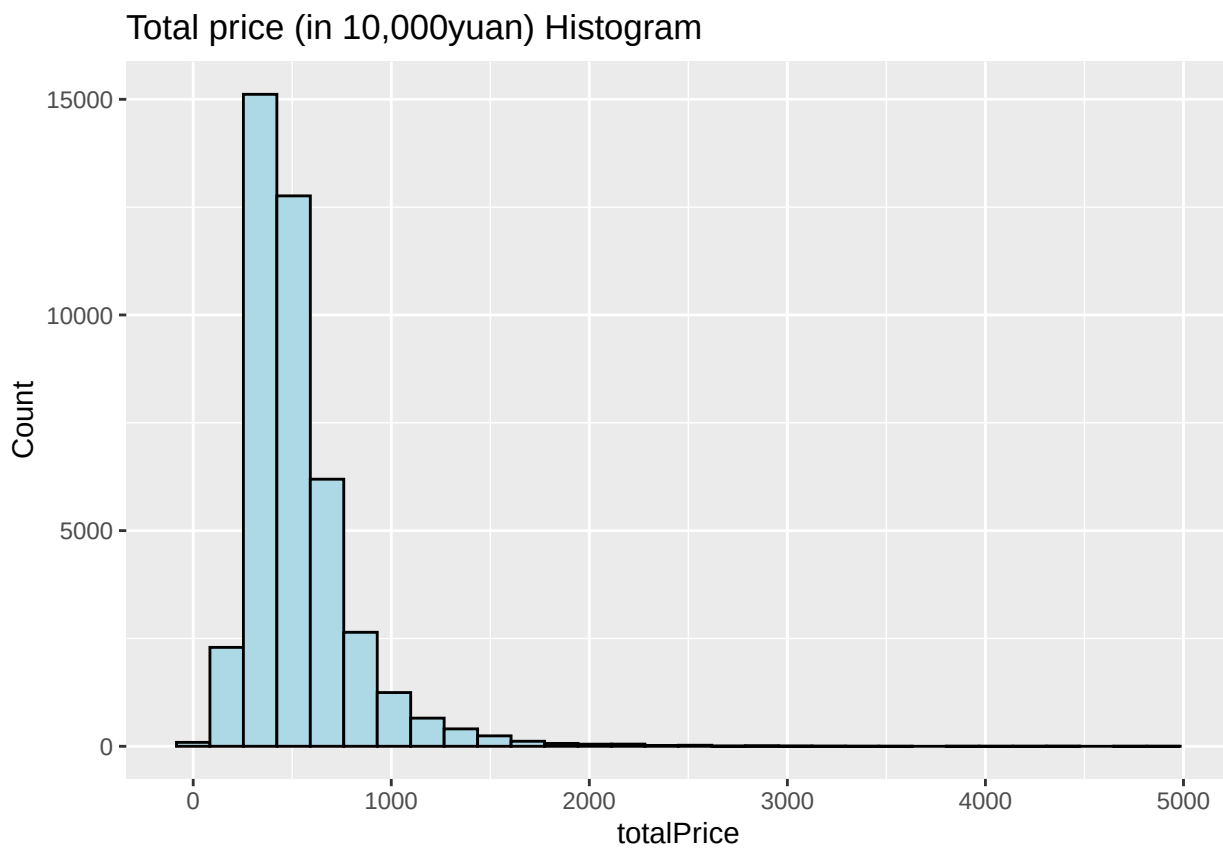
```
##             id DOM followers totalPrice  price square bedRoom livingRoom
## 1 101100732988 132      1085       48.0  26667  18.00       1          1
## 2 101100746033 141      1015      297.5  78393  37.95       1          0
## 3 101101590746 229      1143      380.0 116780  32.54       1          1
## 4 101102303199  25      1045      514.0  87415  58.80       2          1
##   kitchen bathRoom constructionTime renovationCondition buildingStructure
## 1       1        1             2011           hardcover    steel-concrete
## 2       1        1             2007           hardcover    steel-concrete
## 3       1        2             2005          simplicity    steel-concrete
## 4       1        1             1992           hardcover             mixed
##   ladderRatio elevator fiveYearsProperty          subway  district    month day
## 1       0.385      yes                no not_near_subway ChangPing February  24
## 2       0.250      yes               yes     near_subway  ChaoYang    March   9
## 3       0.077      yes               yes     near_subway   XiCheng December  31
## 4       0.500       no               yes     near_subway   HaiDian December   8
##   totalFloor floorRange
## 1         28     medium
## 2         12       high
## 3         10     medium
## 4          6       high
```

3.1.7 totalPrice

The distribution of total price is unimodal and skews right. The average total price is 456x10,000 yuan (around 650~760k USD depend on inflation) and the median is 418x10,000 yuan (around 600~640k USD). Approximately 2.5% of values are below 220x10,000 yuan. Approximately 2.5% of values are above 1250x10,000 yuan.

```r
ggplot(data2017, aes(x=totalPrice))+
  geom_histogram(fill='lightblue', color='black') +
  xlab("totalPrice")+ylab("Count")+ggtitle("Total price (in 10,000yuan) Histogram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
mean(data2017$totalPrice)
```

```
## [1] 526.3465
```

```
quantile(data2017$totalPrice, c(0, 0.025, 0.25, 0.5, 0.75, 0.975, 1))
```

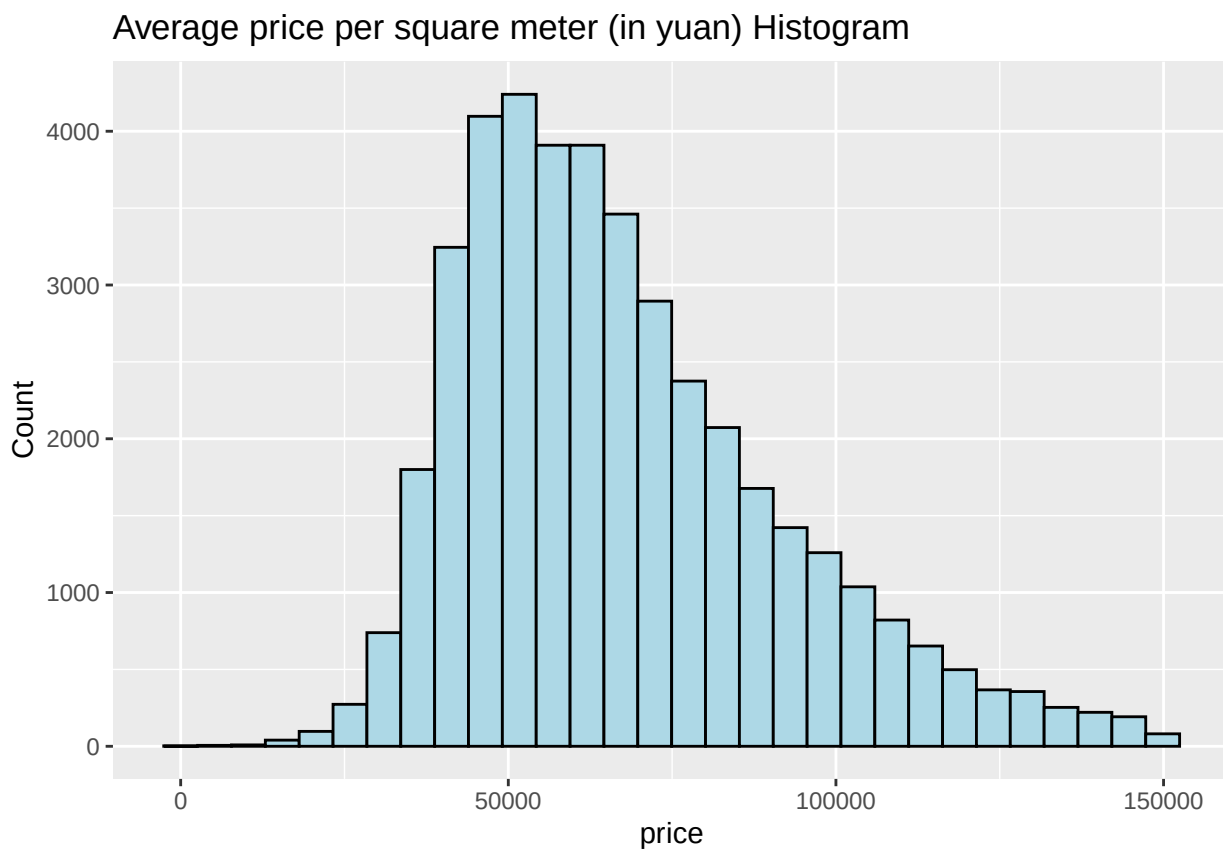
```
##     0%   2.5%    25%    50%    75%  97.5%   100%
##    1.0  215.5  355.0  460.0  616.0 1250.0 4900.0
```

3.1.8 price

The distribution for price is unimodal, and slightly skewed right. The average price per square meter is around 67,430 yuan and median is 62,670 yuan, which are close. Approximately 2.5% of values are above 127,359 yuan. Approximately 2.5% of values are below 33,035 yuan. The house that has the lowest value has 136 yuan per square meter, which is about 20 dollars per square meter. Such low price doesn't make sense in any part of Beijing. Therefore, we decided to remove this outlier.

```
ggplot(data2017, aes(x=price))+
  geom_histogram(fill='lightblue', color='black') +
  xlab("price")+ylab("Count")+ggtitle("Average price per square meter (in yuan) Histogram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
mean(data2017$price)
```

```
## [1] 67428.99
```

```r
quantile(data2017$price, c(0, 0.025, 0.25, 0.5, 0.75, 0.975, 1))
```

```
##        0%      2.5%       25%       50%       75%     97.5%      100%
##    136.00  33034.15  49330.00  62669.00  81075.50 127358.70 150000.00
```

```r
# remove outlier
data2017 <- data2017 %>% filter(price > 1000)
```

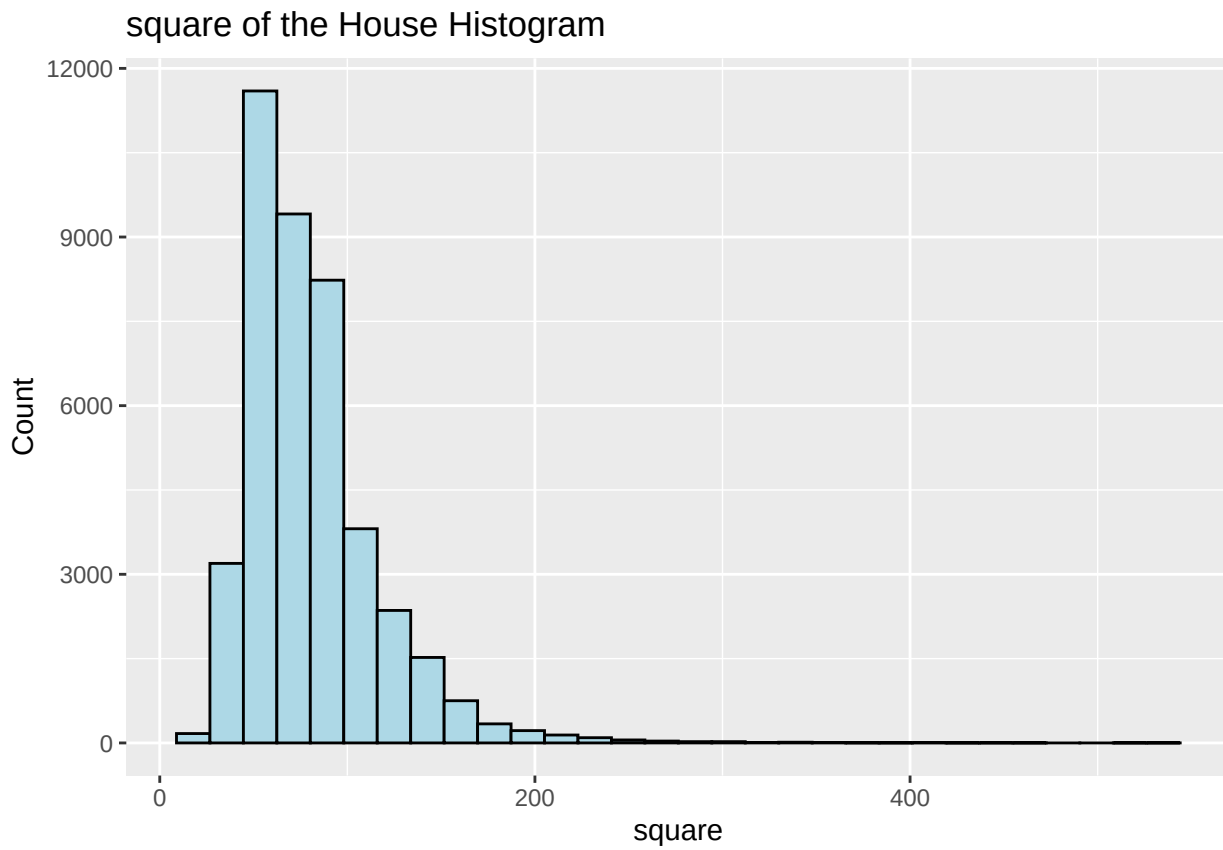
3.1.9 square

The graph of number of squares is unimodal and skewed to the right. The mean is around 80 square meters. The median is 72.4 square meters. Approximately 2.5% of values are below 37.3 square meters. Approximately 2.5% of values are above 166.6 square meters.

We looked at the top 3 houses with the most square meters. Two of them are located in ShunYi, the rural area, which makes it reasonable to have large houses. Another one is located in ChaoYang, one of the core districts, but it has 6 bedrooms. We thought this house is also reasonable.

```r
ggplot(data2017, aes(x=square))+
  geom_histogram(fill='lightblue', color='black') +
  xlab("square")+ylab("Count")+ggtitle("square of the House Histogram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
mean(data2017$square)
```

```
## [1] 80.9039
```

```r
quantile(data2017$square, c(0, 0.025, 0.25, 0.5, 0.75, 0.975, 1))
```

```
##       0%     2.5%      25%      50%      75%    97.5%     100%
##  15.0000  37.2725  57.3800  72.3600  95.0600 166.5900 532.2500
```

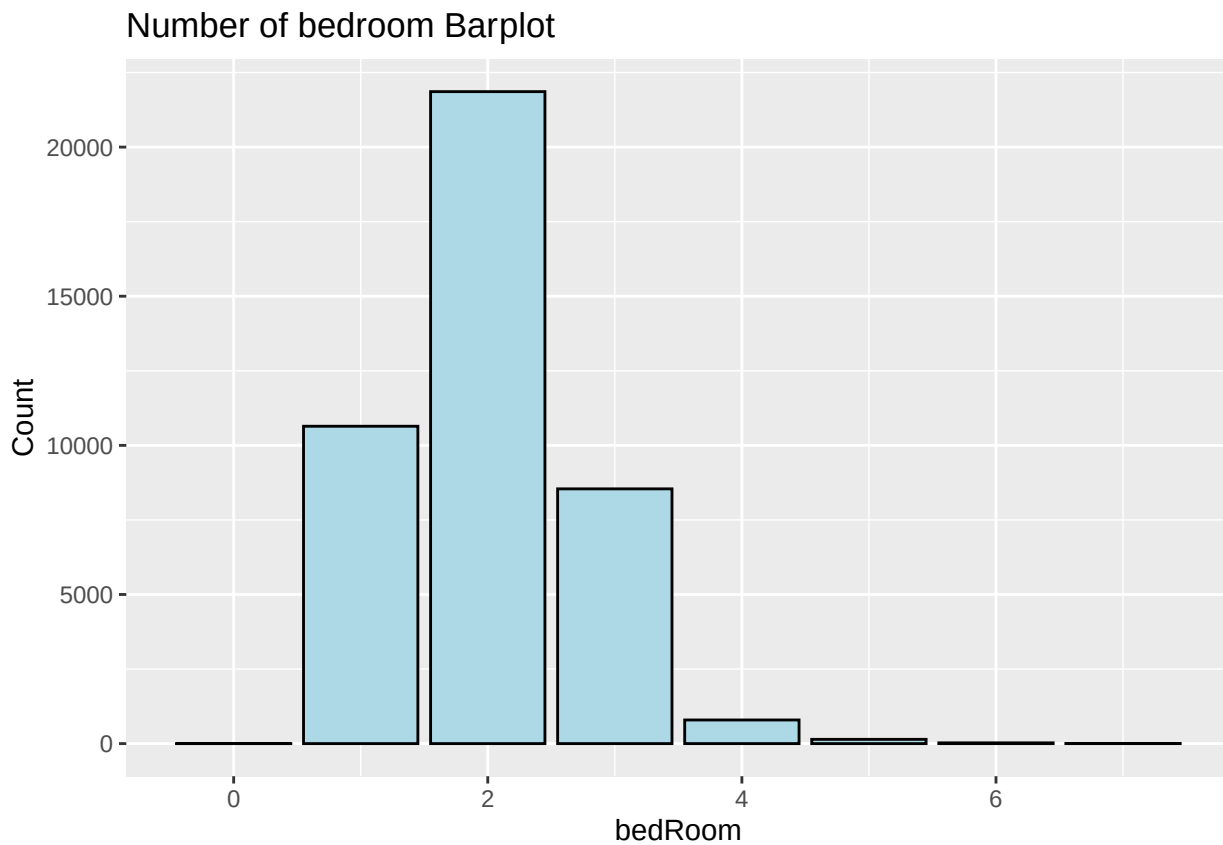
```r
data2017 %>% select(price, square, district, bedRoom) %>%
  arrange(desc(square)) %>% head(3)
```

```
##   price square district bedRoom
## 1 28183 532.25   ShunYi       6
## 2 42613 509.00   ShunYi       5
## 3 72162 458.00 ChaoYang       6
```

3.1.10 bedRoom

The graph of number of bedrooms is unimodal and skews right. Most of the houses have 2 bedrooms, and the mean and median number of bedrooms is also 2. Approximately 25% of values are below or equal to 1. Approximately 2.5% of values are above 3.

```
ggplot(data2017, aes(x=bedRoom))+
  geom_bar(fill='lightblue', color='black') +
  xlab("bedRoom")+ylab("Count")+ggtitle("Number of bedroom Barplot")
```




```
mean(data2017$bedRoom)
```

```
## [1] 2.000881
```

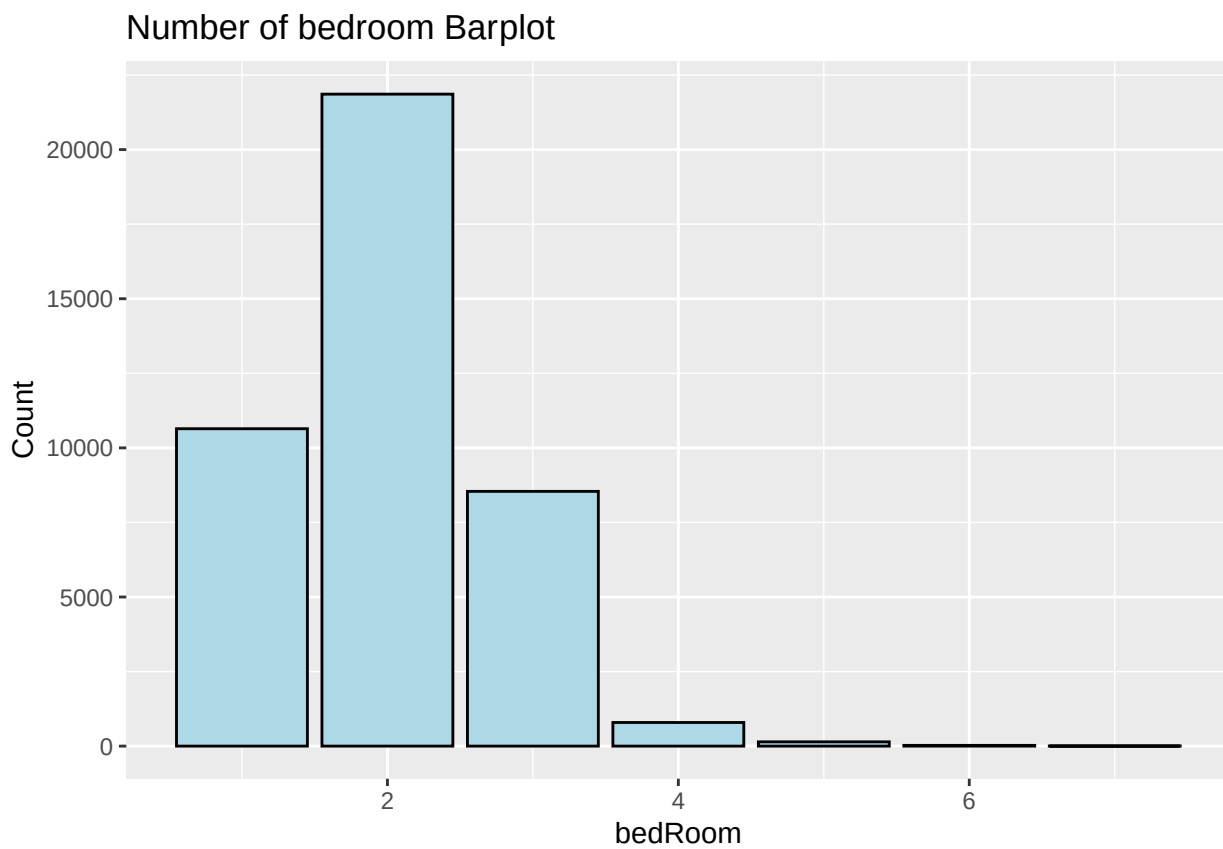
```
quantile(data2017$bedRoom, c(0, 0.025, 0.25, 0.5, 0.75, 0.975, 1))
```

```
##    0%  2.5%   25%   50%   75% 97.5%  100%
##     0     1     1     2     2     3     7
```

There is one house has 0 bedroom but sold in 145,612 yuan per square meter which almost reached the highest price among all observations (150,000 yuan). It is located in XiCheng. According to Lianjia, it belongs to "RongFeng 2008" community. After a research on the community, we found out that this is the average price per square meter for this community. An investigation conducted by The Beijing News indicates that "RongFeng 2008" has a relative 10,000 yuan higher price per square meter compared with other communities

in that area due to management and good school district. But since the properties in this community have very small areas, the total price is still acceptable for most people who wanted to move to XiCheng. Also, this house is labeled 0 bedroom but 1 livingroom, which might be a mistake. This house could be considered as a studio with total of 1 room. Thus, we decide to change it from "1 livingroom, 0 bedroom" to "1 bedroom, 0 livingroom". So this house is not an outlier. (https://www.sohu.com/a/479132582_114988)

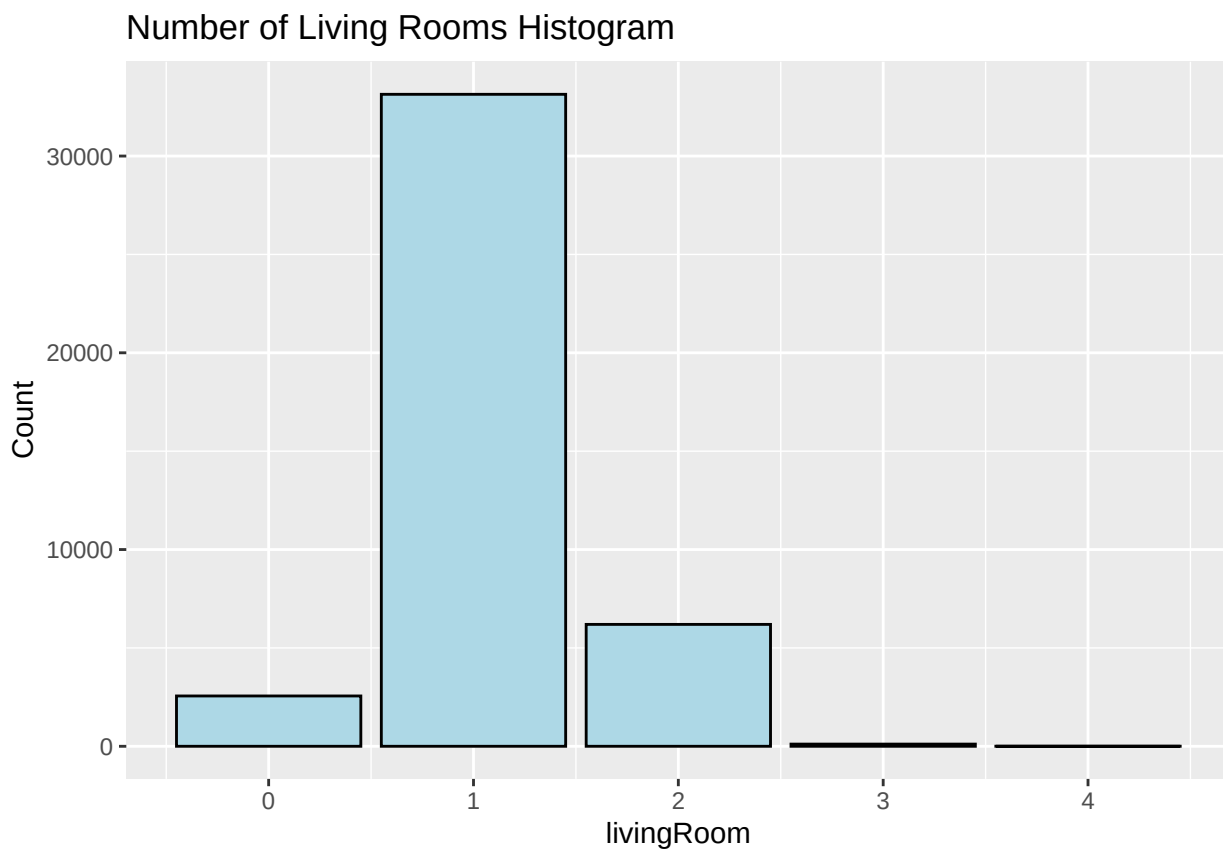
```
data2017[data2017$bedRoom == 0, 8] <- 0
data2017[data2017$bedRoom == 0, 7] <- 1

ggplot(data2017, aes(x=bedRoom))+
  geom_bar(fill='lightblue', color='black') +
  xlab("bedRoom")+ylab("Count")+ggtitle("Number of bedroom Barplot")
```



3.1.11 livingRoom

The distribution of number of livling rooms is unimodal and skews right. The mean and median number of living room is 1. Approximately 2.5% of values are above 2. The number of living rooms for all houses sold in 2017 is between 0 to 4. 79% of properties have 1 living room. Properties with 0 living room is reasonable because these apartments are usually called "studios", which is common in the city. We also consider 4 living rooms as reasonable because sometimes dining rooms are counted as living rooms in China.

```
ggplot(data2017, aes(x=livingRoom))+
  geom_bar(fill='lightblue', color='black') +
  xlab("livingRoom")+ylab("Count")+ggtitle("Number of Living Rooms Histogram")
```




```r
table(data2017$livingRoom)/length(data2017$livingRoom)
```

```
##
##            0            1            2            3            4
## 0.0608960625 0.7888396896 0.1474789316 0.0026424796 0.0001428367
```

```r
mean(data2017$livingRoom)
```

```
## [1] 1.092296
```

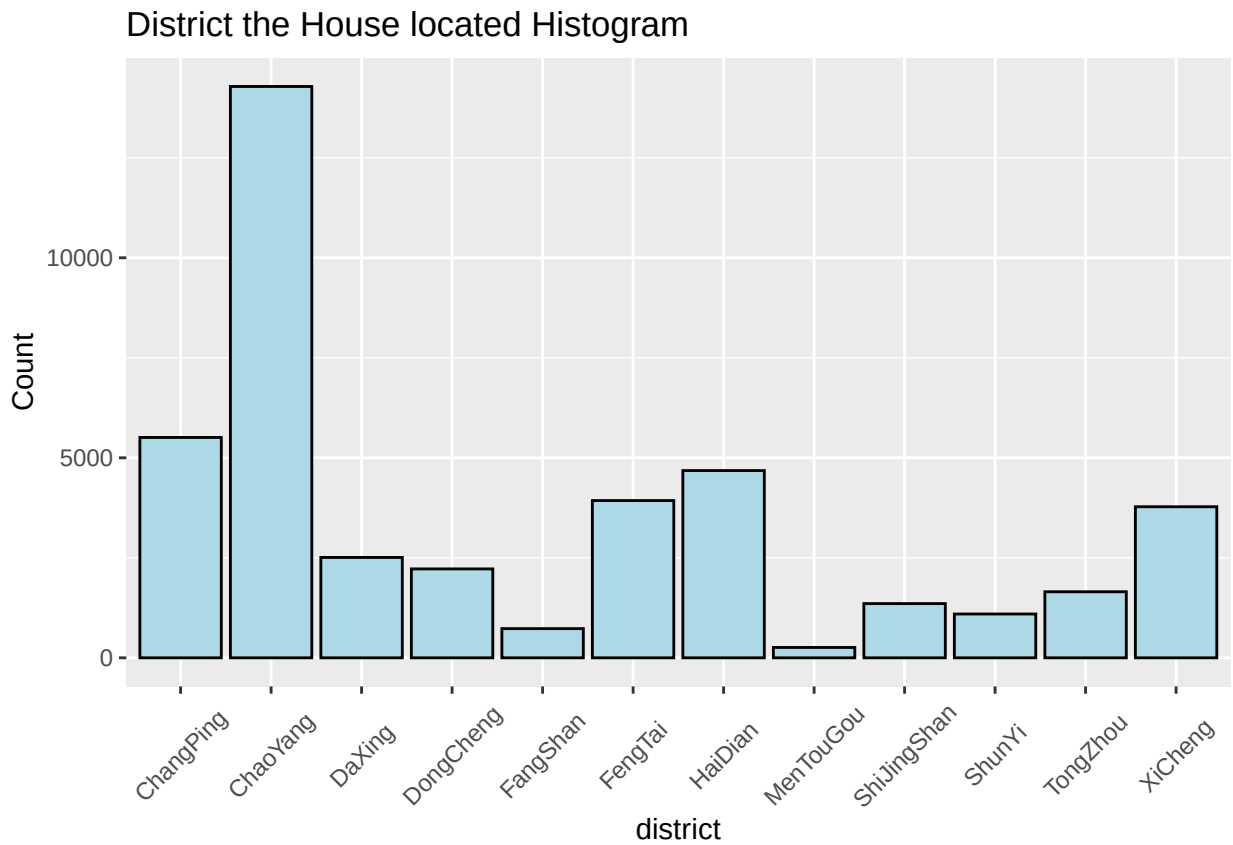
```r
quantile(data2017$livingRoom, c(0,0.025,0.25,0.5,0.75,0.975,1))
```

```
##    0%  2.5%   25%   50%   75% 97.5%  100%
##     0     0     1     1     1     2     4
```

3.1.12 district (Categorical)

ChaoYang as one of the core districts (yellow and red) has the most houses sold in 2017. XiCheng and Haidian have similar amount of houses sold where XiCheng is a smaller district than Haidian.
FangShan and MenTouGou have significant small amounts of houses transactions where these 2 are suburb districts (purple) that have relative large areas shown on the map. In contrast, as another suburb district, ChangPing has the second greatest amounts of houses sold in 2017.

```r
ggplot(data2017, aes(x=district))+
  geom_bar(fill='lightblue', color='black') +
  xlab("district")+ylab("Count")+theme(axis.text.x = element_text(angle = 45, vjust = 0.5))+
  ggtitle("District the House located Histogram")
```



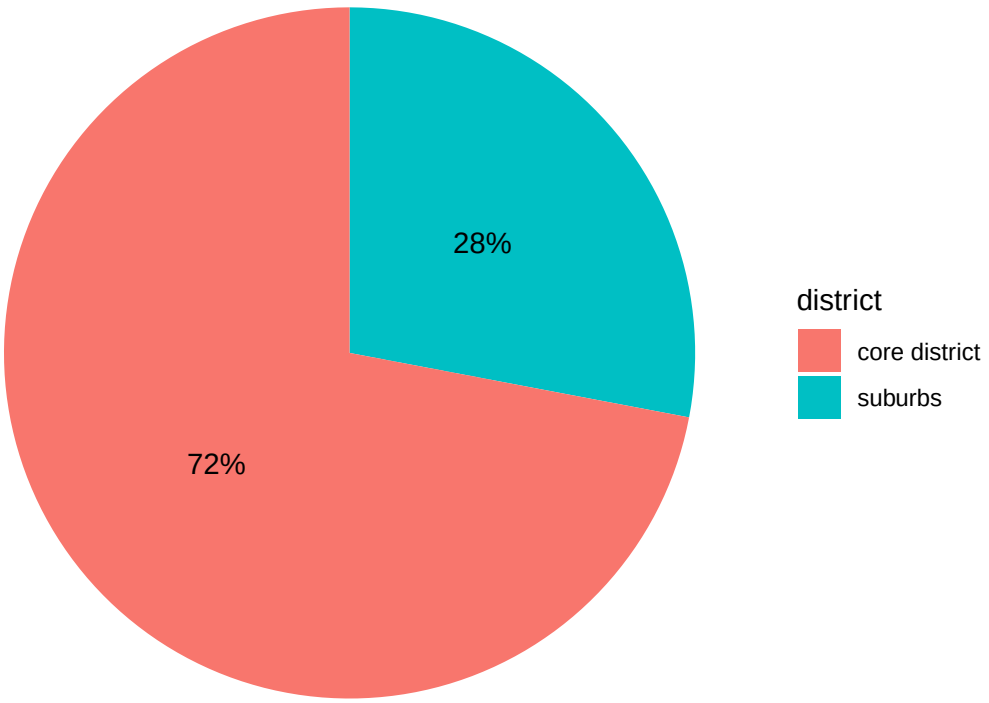

```r
table(data2017$district)/length(data2017$district) # percentage of houses sold in each district
```

```
##
##   ChangPing    ChaoYang      DaXing   DongCheng    FangShan     FengTai
## 0.131147931 0.339999048 0.059753369 0.052921011 0.017378470 0.093581869
##     HaiDian   MenTouGou ShiJingShan      ShunYi    TongZhou     XiCheng
## 0.111436461 0.006165786 0.032281103 0.026091511 0.039327715 0.089915726
```

Percentage of houses sold in core districts or suburb Both suburb and core districts contain 6 districts. Though suburb have much larger areas shown on the map, 28% of houses sold are located in the suburb where core districts have houses sold that made up 72% of the market.

```r
core_district <- data2017 %>%
  filter(district %in% c("XiCheng" , "DongCheng" , "ChaoYang" , "HaiDian" , "ShiJingShan" , "FengTai"))

suburb <- data2017 %>%
  filter(district %in% c("DaXing", "ChangPing", "FangShan", "MenTouGou", "ShunYi", "TongZhou")) %>% nrou

district_table <- data.frame(district = c("suburbs", "core district"), numbers = c(suburb, core_distric
```

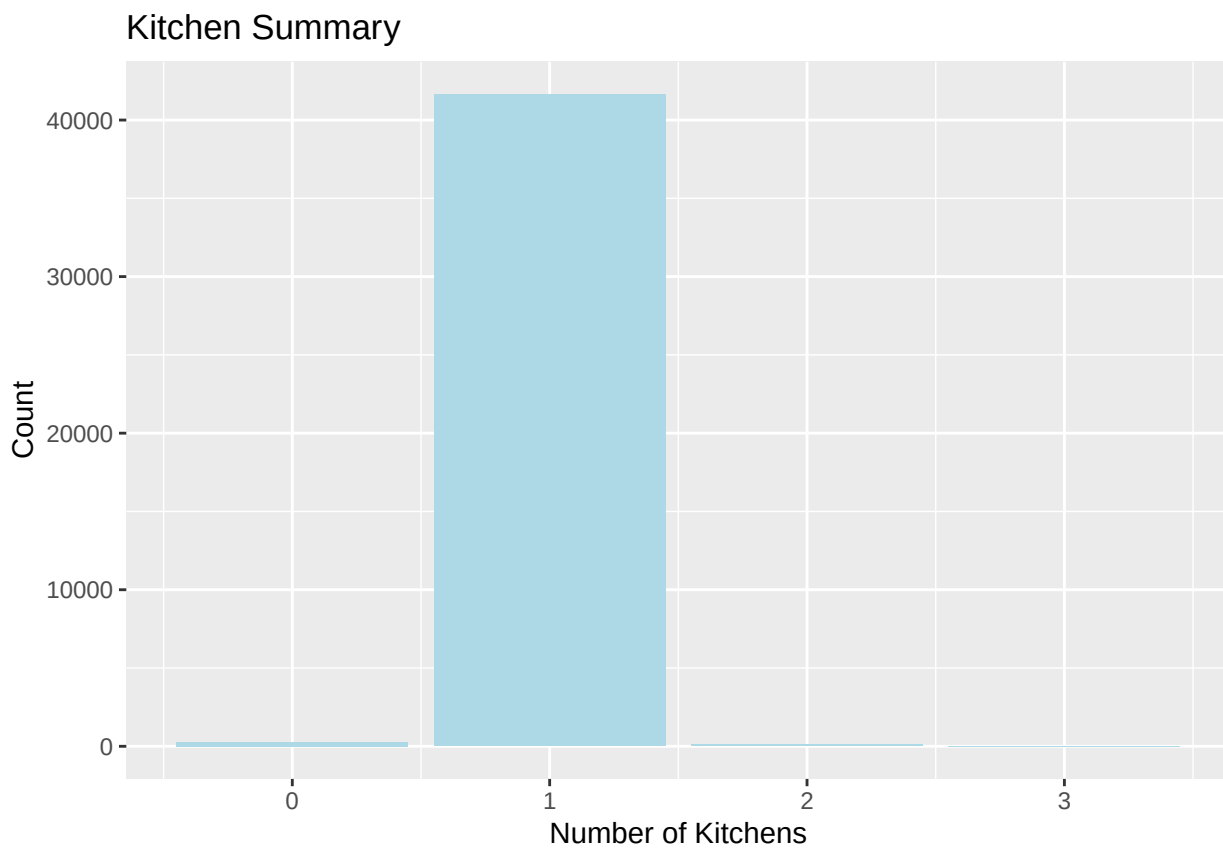
```r
district_table$label <- scales::percent(district_table$numbers/length(data2017$district))


ggplot(district_table, aes(x="", y=numbers, fill=district)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0)+
  theme_void()+
  geom_text(aes(x=1, y = cumsum(numbers) - numbers/2, label=label))
```



3.1.13 kitchen

The distribution for the number of kitchens shows that most listings have only one kitchen in their property.

```r
ggplot(data2017, aes(x = kitchen)) +
  geom_bar(fill = "lightblue") +
  ggtitle("Kitchen Summary") + xlab("Number of Kitchens") + ylab("Count")
```




```
data2017 %>%
  filter(kitchen > 2)
```

```
##             id DOM followers totalPrice price square bedRoom livingRoom kitchen
## 1 101101782929  69        78        735 26750 274.77       4          2       3
##   bathRoom constructionTime renovationCondition buildingStructure ladderRatio
## 1        2             2001           hardcover    steel-concrete         0.5
##   elevator fiveYearsProperty      subway  district     month day totalFloor
## 1      yes               yes near_subway ChangPing September  13         13
##   floorRange
## 1       high
```

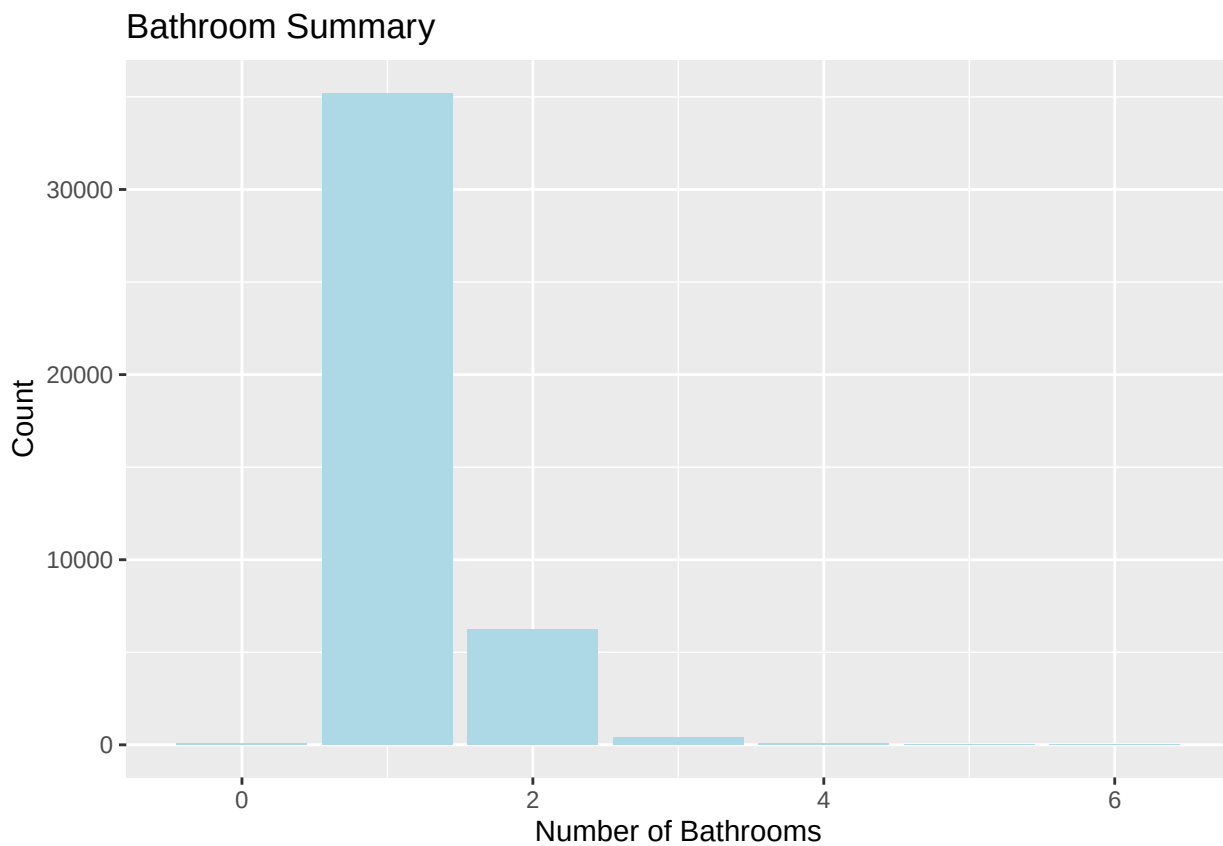
There are two houses with more than 2 kitchens, which seem odd. Taking a closer look, it had less than 300 square meters, which is impossible to have 3 kitchens in fairly medium house size. Therefore, we removed these this outlier.

```
data2017 <- data2017 %>%
  filter(kitchen < 3)
```

3.1.14 bathroom

The distribution for the number of bathrooms shows that most listings have only one bathroom in their property. This distribution skews heavily to the right and the count decreases as number of bathrooms increases.

```r
ggplot(data2017, aes(x = bathRoom)) +
  geom_bar(fill = "lightblue") +
  ggtitle("Bathroom Summary") + xlab("Number of Bathrooms") + ylab("Count")
```




```r
data2017 %>% filter(bathRoom > 4) %>% select(id, price, square, bedRoom, bathRoom)
```

```
##             id  price square bedRoom bathRoom
## 1 101092296449 105654 407.37       4        5
## 2 101100859813 102625 384.90       5        5
## 3 101101119398  28183 532.25       6        6
## 4 101101463740  39033 383.01       5        5
## 5 101101509627  97344 439.68       4        5
## 6 101101667831  81483 349.77       6        5
## 7 101101797841 101611 282.45       3        5
```

We took a closer look at properties that have more than 4 bathrooms. It's odd to have more bathrooms than bedrooms, so we removed the three properties that had more bathrooms than bedrooms.

```r
data2017 <- data2017 %>% filter(!id %in% c("101092296449", "101101797841", "101101509627"))
```

3.1.15 construction time

```r
quantile(data2017$constructionTime, c(0, .025, .25, .5, .75, .975, 1))
```

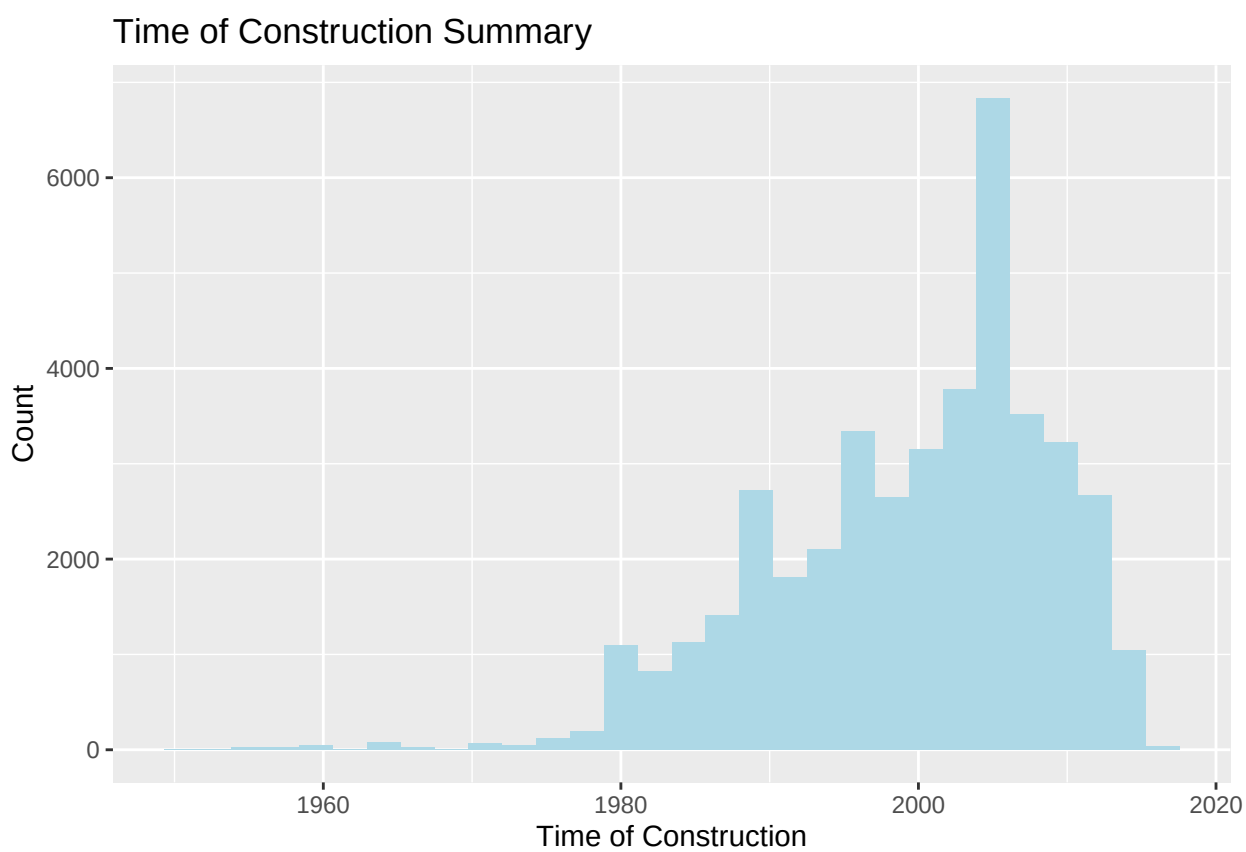
```
##    0%  2.5%   25%   50%   75% 97.5%  100%
##  1950  1980  1993  2002  2006  2014  2016
```

The year of construction of the listings that were transacted in 2017 ranges from the year 1950 to the year 2016.

The median is 2002. Approximately 2.5% of values are above 2014. Approximately 2.5% of values are below 1980.

```r
ggplot(data2017, aes(x = constructionTime)) +
  geom_histogram(fill = "lightblue") +
  ggtitle("Time of Construction Summary") + xlab("Time of Construction") + ylab("Count")
```

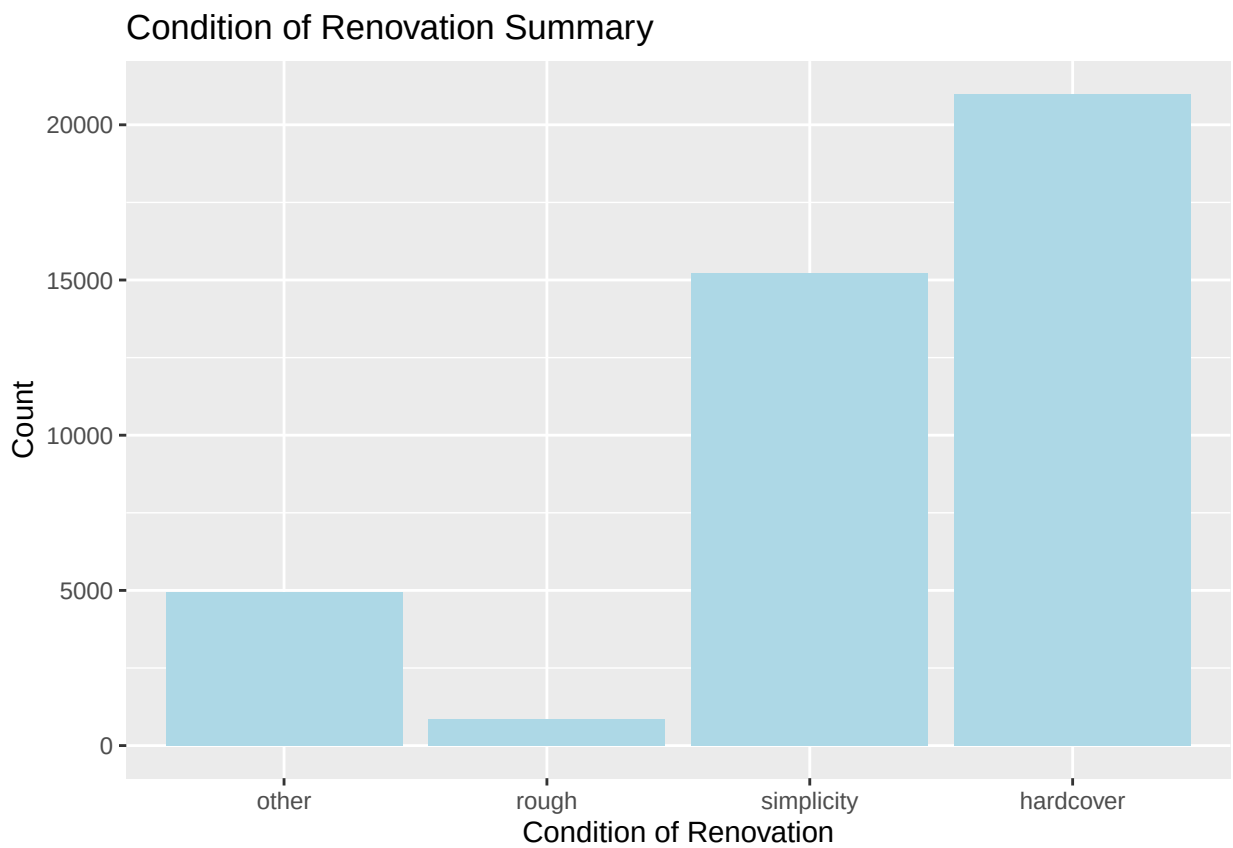
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



3.1.16 renovation condition

The distribution for the condition of renovation shows that most listings have the highest rating of hardcover in terms of the condition of renovation for their listing and simplicity is the second, showing that most of the houses sold in 2017 are in good renovation condition.

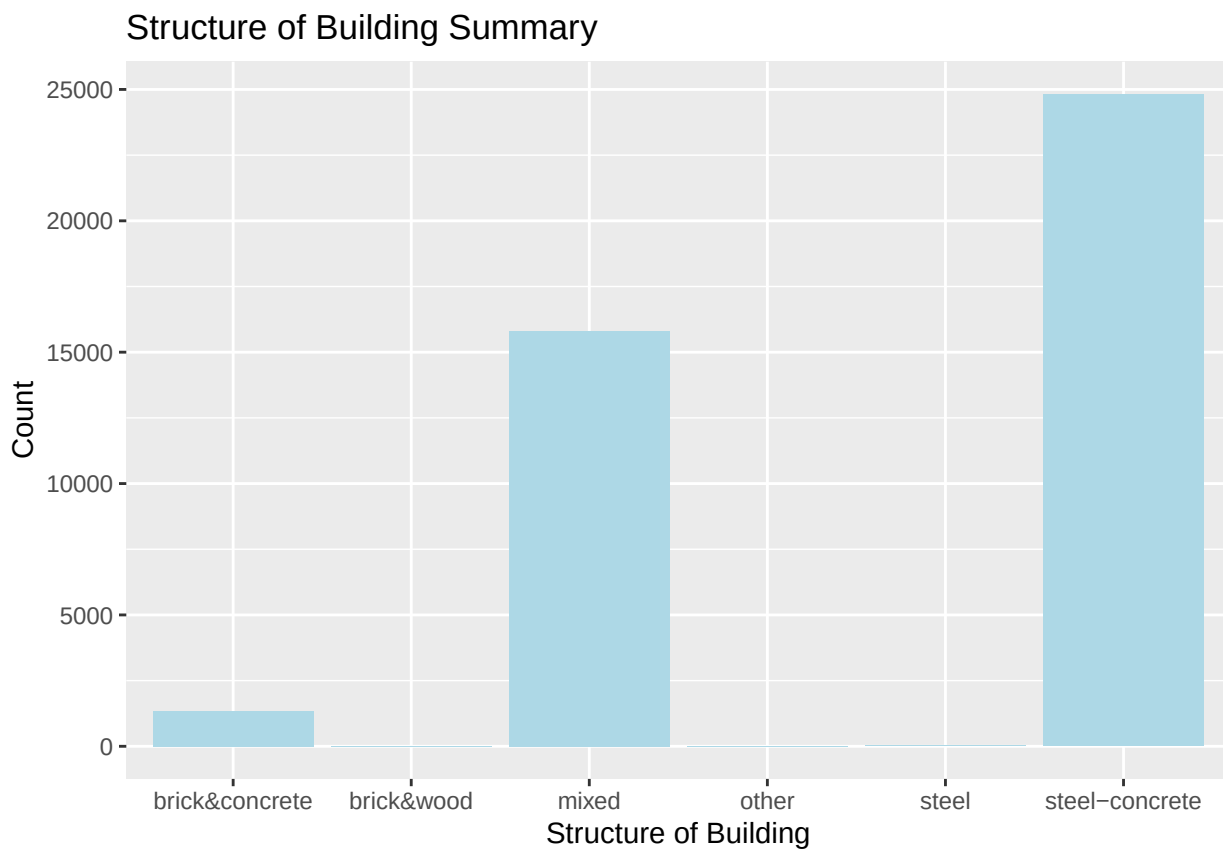
```r
data2017$renovationCondition <- factor(data2017$renovationCondition,
                   levels = c("other", "rough", "simplicity", "hardcover"))
ggplot(data2017, aes(x = renovationCondition)) +
  geom_bar(fill = "lightblue") +
  ggtitle("Condition of Renovation Summary") + xlab("Condition of Renovation") + ylab("Count")
```



3.1.17 building structure

The distribution for the building structure shows that most listings are made of steel-concrete composite, with mixed being second. There are very few properties made of brick&wood, which is uncommon in a big city. There are also very few steel houses.

```r
ggplot(data2017, aes(x = buildingStructure)) +
  geom_bar(fill = "lightblue") +
  ggtitle("Structure of Building Summary") + xlab("Structure of Building") + ylab("Count")
```




```
table(data2017$buildingStructure)
```

```
##
## brick&concrete     brick&wood          mixed          other          steel
##           1346             10          15800              2             30
## steel-concrete
##          24814
```

3.1.18 ladder ratio

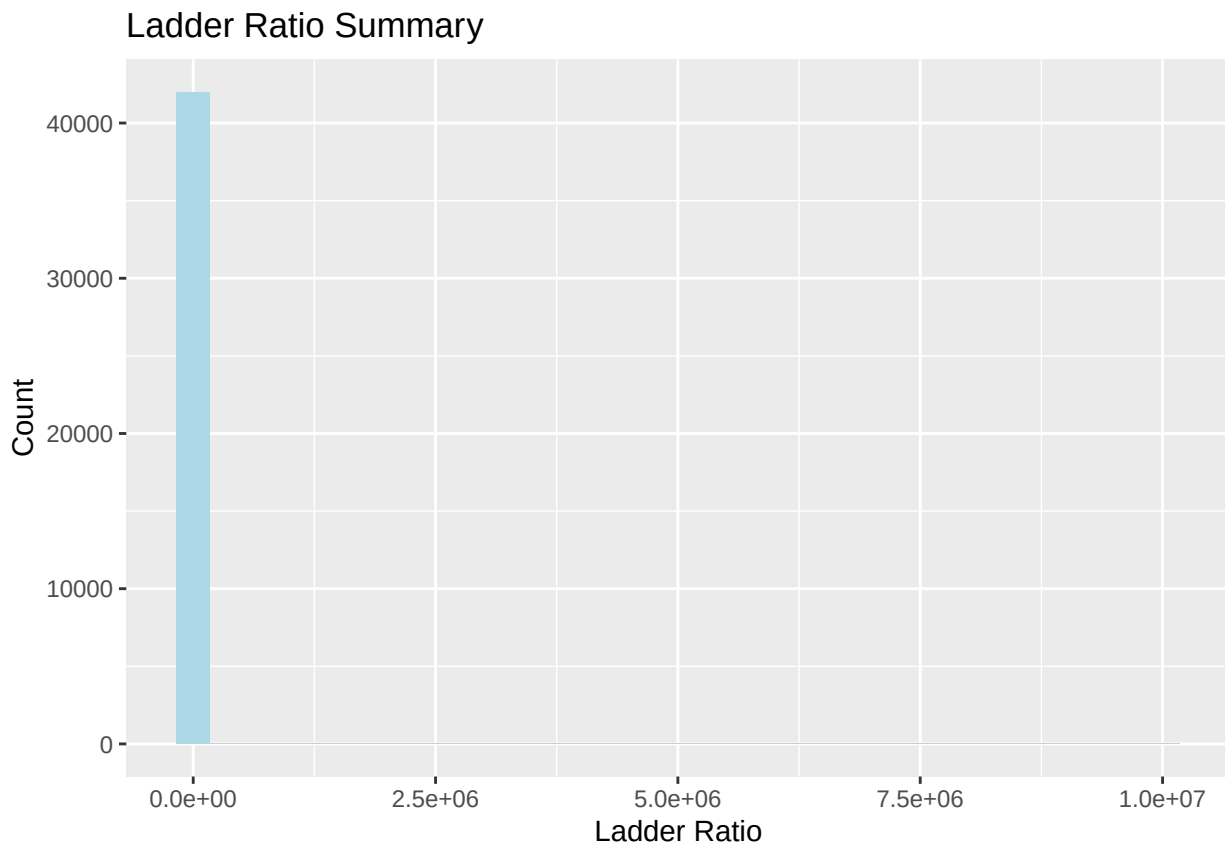
```
quantile(data2017$ladderRatio, c(0, .025, .25, .5, .75, .975, 1))
```

```
##          0%        2.5%         25%         50%         75%       97.5%
## 1.40000e-02 1.21000e-01 2.50000e-01 3.33000e-01 5.00000e-01 6.67000e-01
##        100%
## 1.00094e+07
```

The median is 0.333. Approximately 2.5% of values are above 0.667. Approximately 2.5% of values are below 0.121.

```
ggplot(data2017, aes(x = ladderRatio)) +
  geom_histogram(fill = "lightblue") +
  ggtitle("Ladder Ratio Summary") + xlab("Ladder Ratio") + ylab("Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

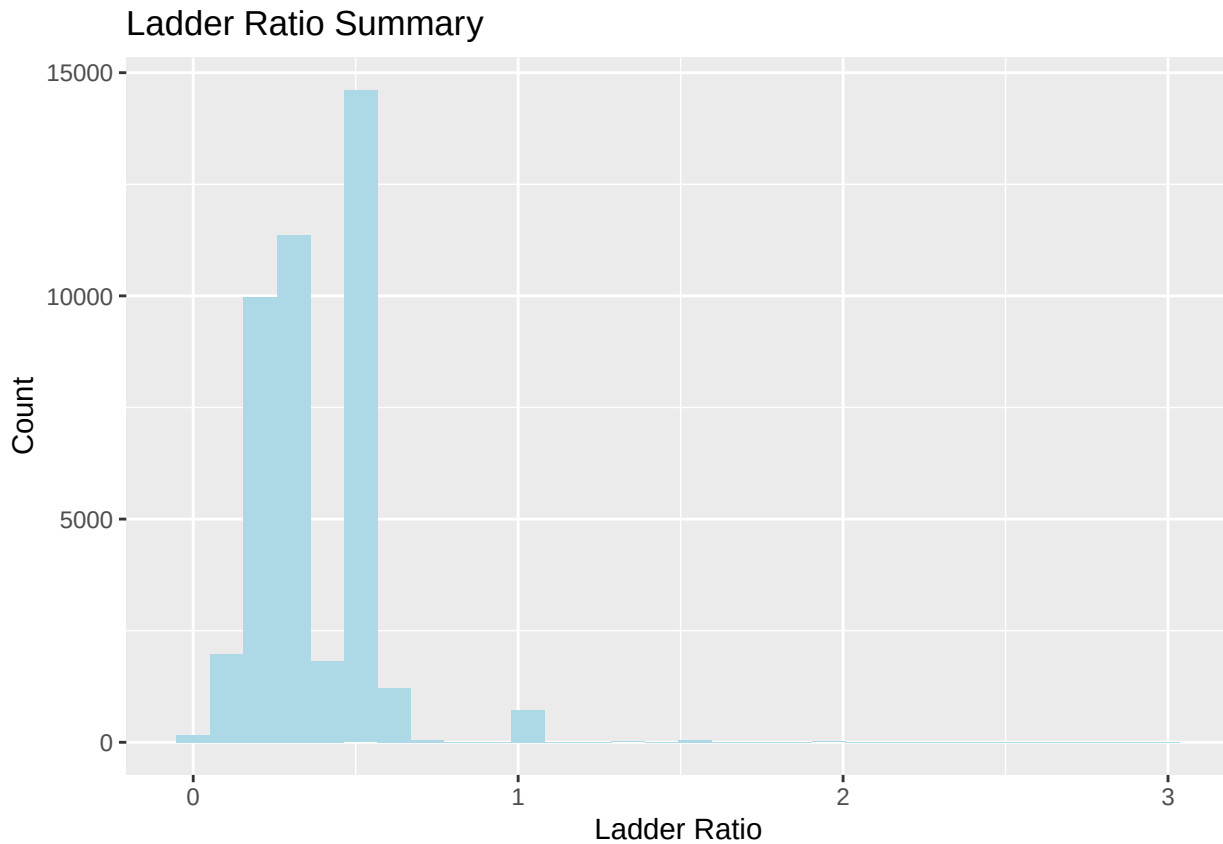


The maximum value for ladderRatio is 10009400, which seems like a typo, so we decided to remove this observation.

```
data2017 <- data2017 %>%
  filter(ladderRatio < 5)
```

```
ggplot(data2017, aes(x = ladderRatio)) +
  geom_histogram(fill = "lightblue") +
  ggtitle("Ladder Ratio Summary") + xlab("Ladder Ratio") + ylab("Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

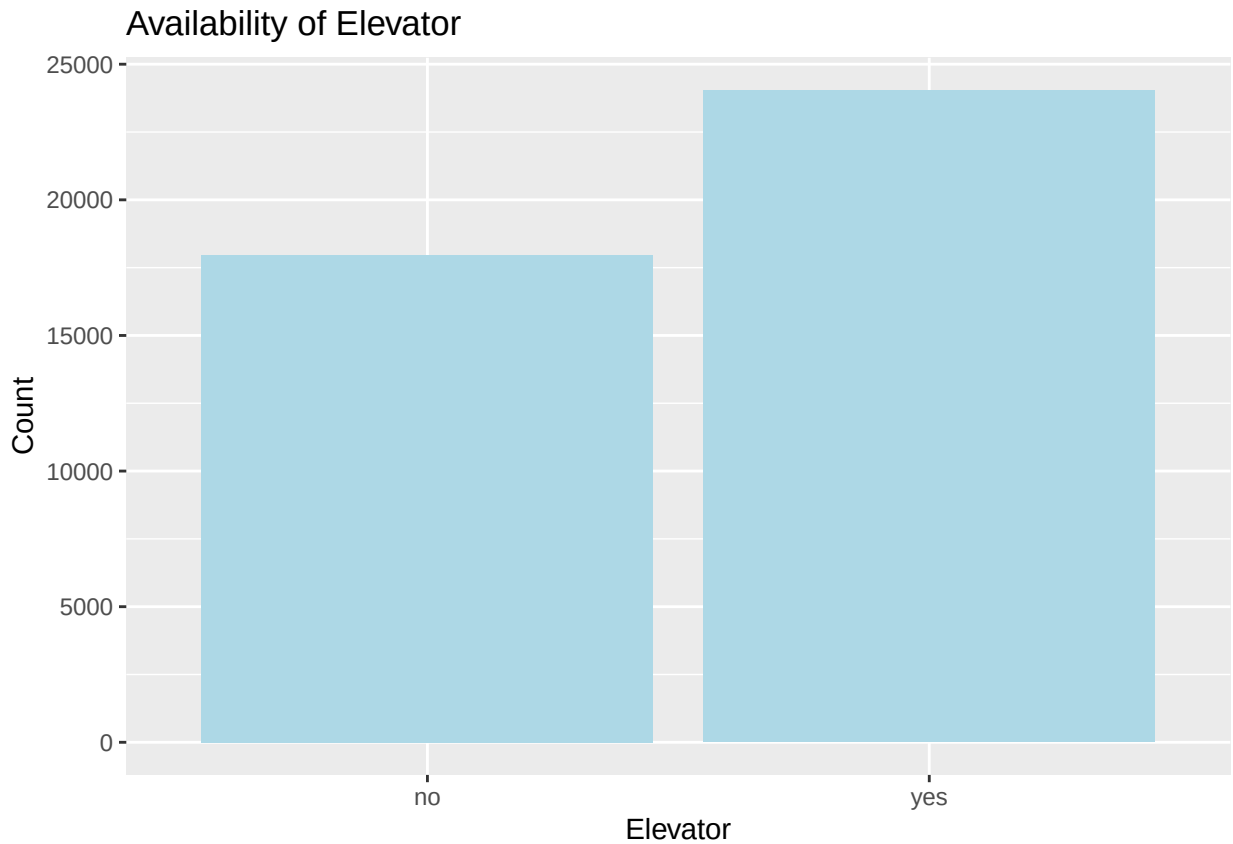


Now, the plot is unimodal and skews to the right, with most values ranging between 0 and 1.

3.1.19 elevator

The distribution for the presence of an elevator shows that it is more common for listings to have at lease one elevator in the building compared to none at all.

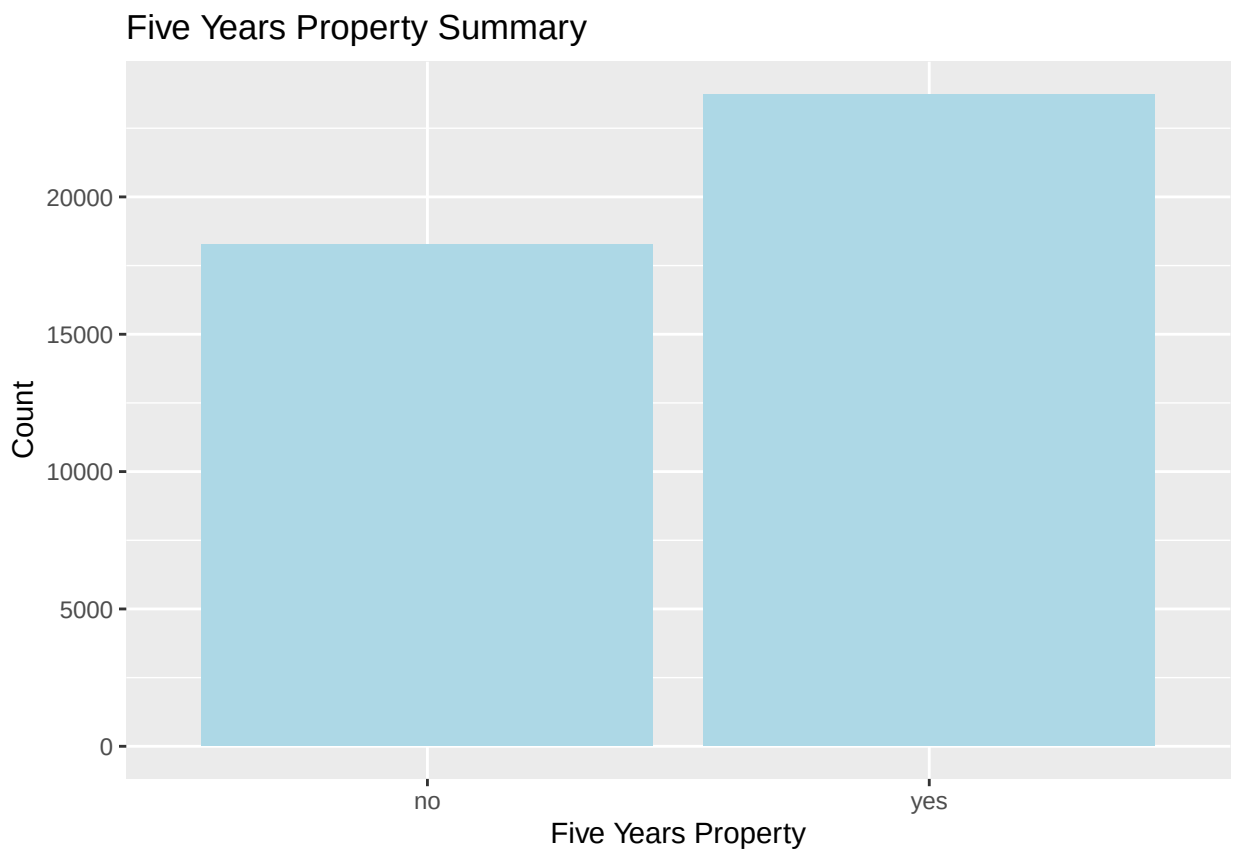
```
ggplot(data2017, aes(x = elevator)) +
  geom_bar(fill = "lightblue") +
  ggtitle("Availability of Elevator") +xlab("Elevator") +ylab("Count")
```



3.1.20 five years property

The distribution shows that among the previous owners of the listings, a higher number of them have stayed in the listing for less than five years and a lower number of them have stayed in the listing for exactly five years or more than five years.

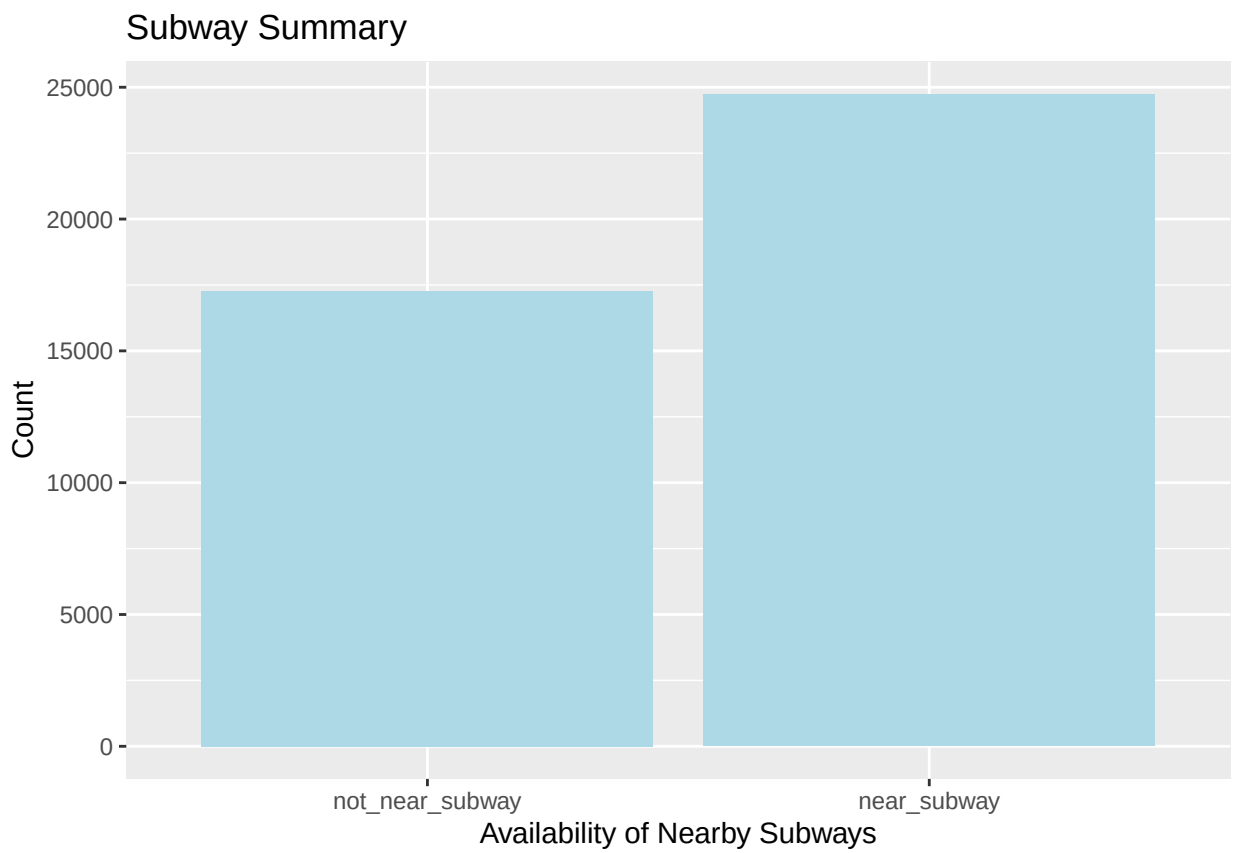
```
ggplot(data2017, aes(x = fiveYearsProperty)) +
  geom_bar(fill = "lightblue") +
  ggtitle("Five Years Property Summary") + xlab("Five Years Property") + ylab("Count")
```



3.1.21 subway

The distribution for the presence of any nearby subways shows that it is more common for listings to have at lease one subway nearby the building compared to none at all.

```
ggplot(data2017, aes(x = subway)) +
  geom_bar(fill = "lightblue") +
  ggtitle("Subway Summary") + xlab("Availability of Nearby Subways") + ylab("Count")
```



3.2 Potential Relationships between Variables

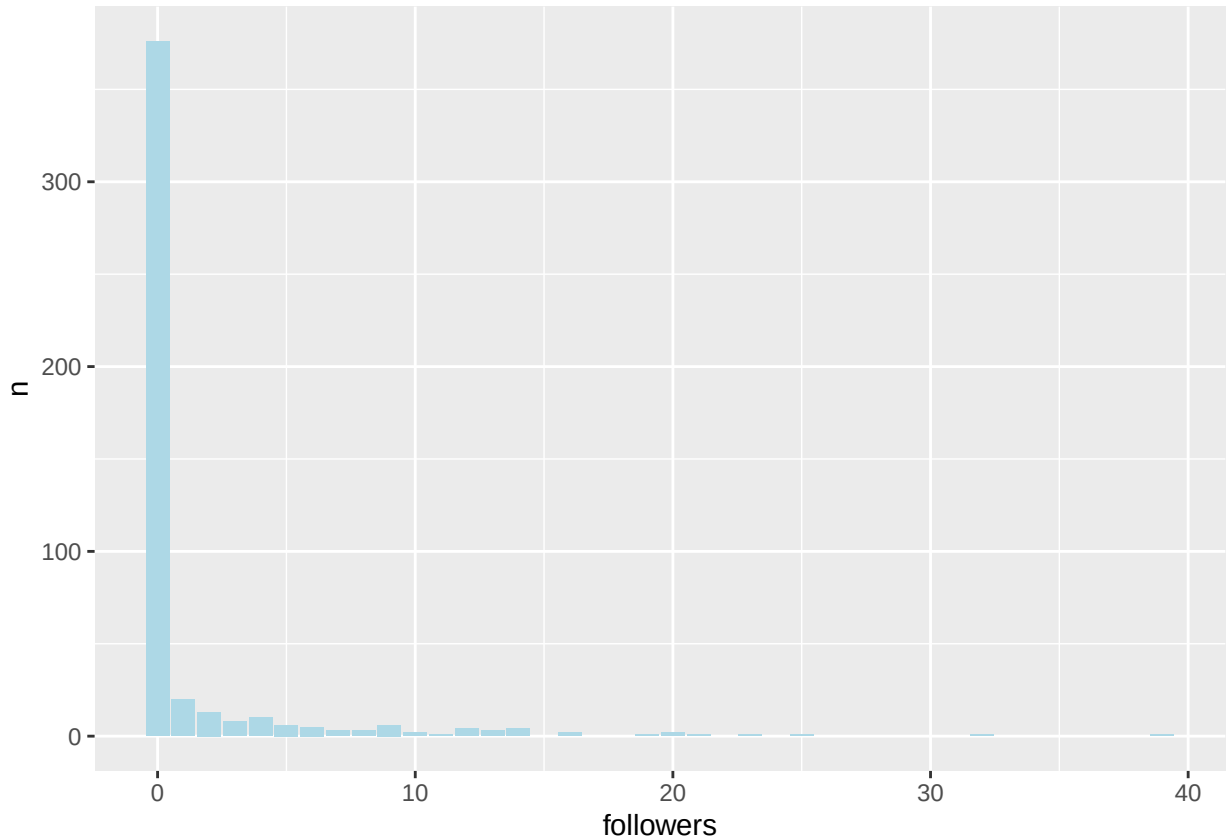
3.2.1 DOM vs. followers

Do houses sold fast have many followers? Among these 474 houses that were sold within 1 day, 376 have no follower and the maximum number of followers is 39, which is not a significant number. Our guessing is that the owner and buyer might have already planned about this deal privately and they just list the property on an official third-party website like LianJia to make the transaction secure and supervised.

```
table(data2017$DOM<=1) # houses are 0 or 1 Days on market
```

```
##
## FALSE  TRUE
## 41527   474
```

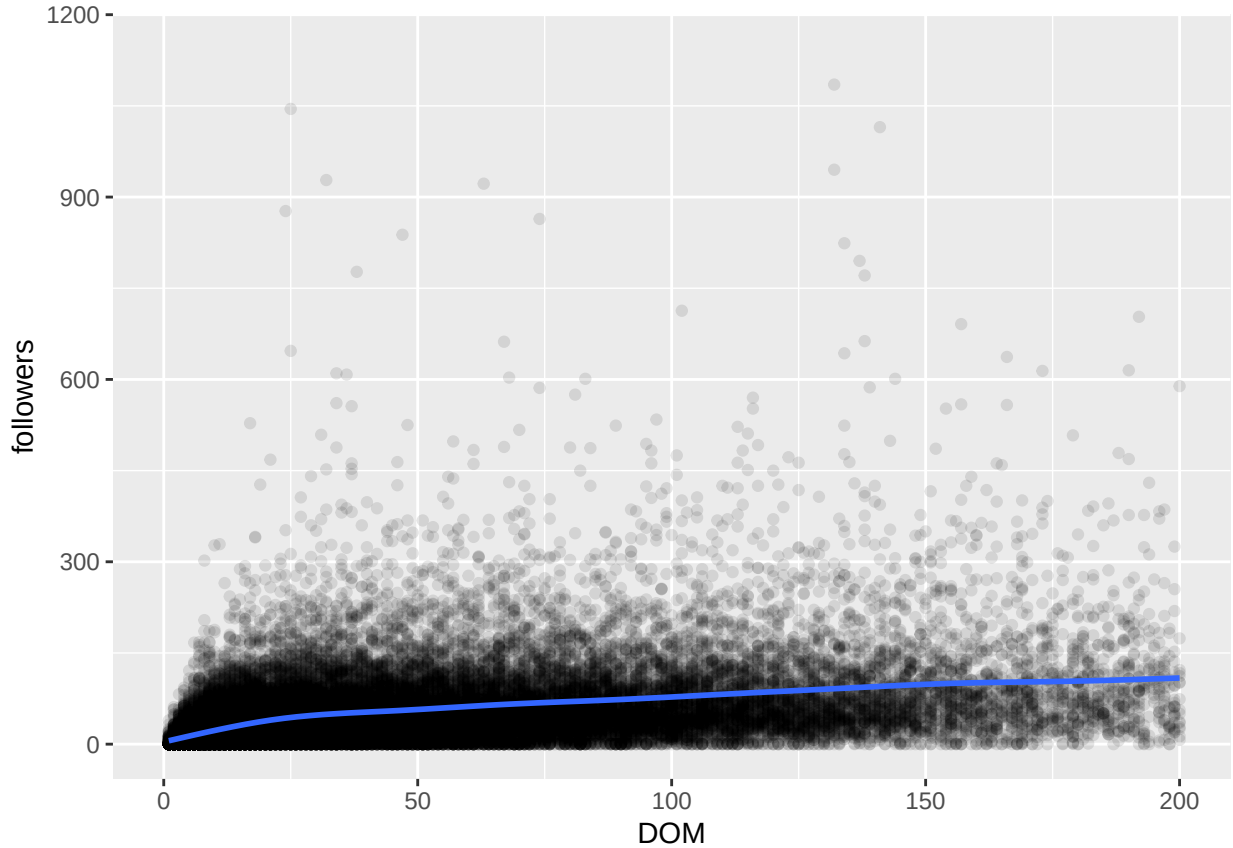
```
follower1DOM <- data2017 %>%
  filter(DOM<=1) %>%
  count(followers) # Do houses sold fast have many followers?
ggplot(follower1DOM, aes(x=followers,y=n))+
  geom_col(fill="lightblue")
```



Let's take a closer look at the houses sold within 200 days on market. There seems to be a slightly increase on the number of followers as it stays more days on market. But it is not as what we expected that the more days a property listed on market the more followers it has.

```
ggplot(data2017, aes(x=DOM,y=followers))+ # overall number of followers vs. DOM
  geom_point(alpha=.1)+
  geom_smooth()+
  xlim(0,200)
```

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

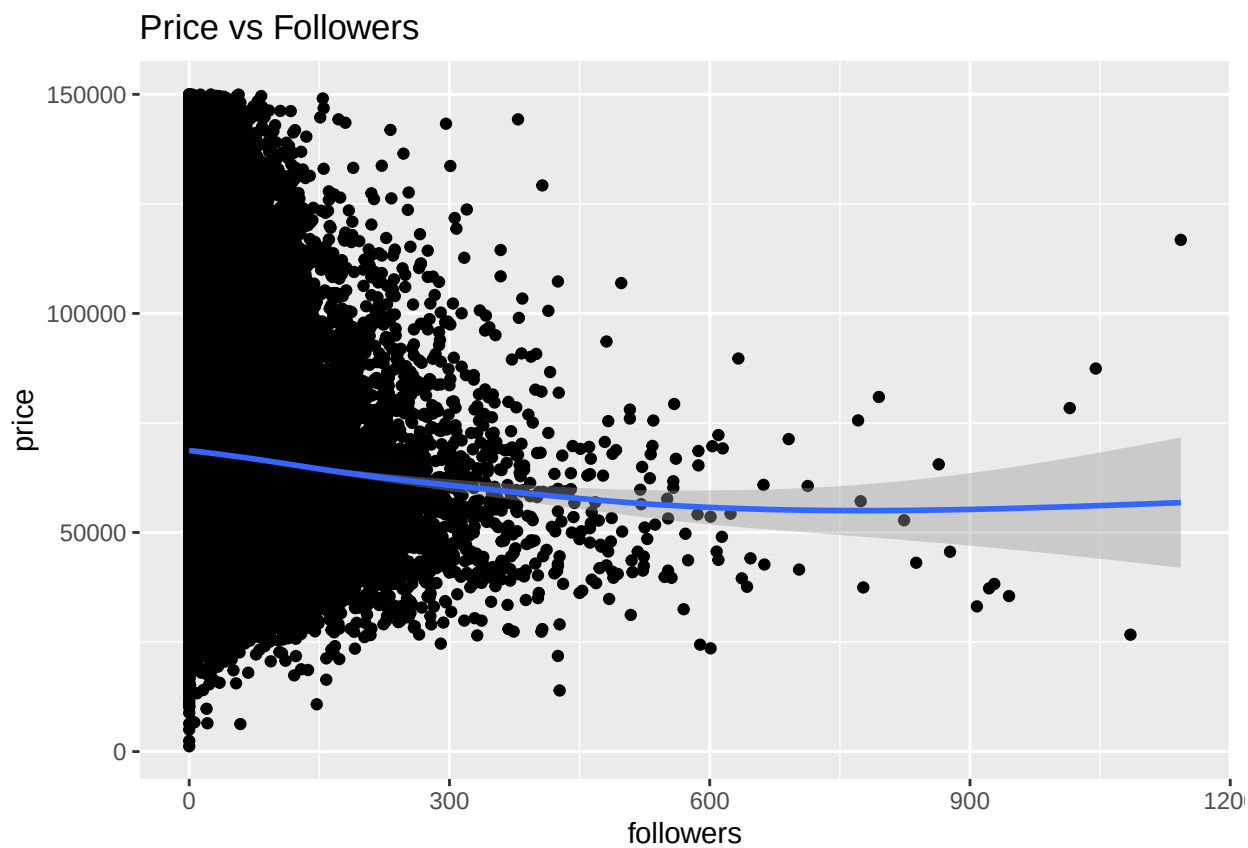


3.2.2 Followers vs Price

We would expect the price to be lower if there were many followers on a property because people like to keep an eye on things that are cheaper.

```
ggplot(data2017, aes(x=followers, y=price))+ geom_point() + geom_smooth() +
  labs(title = "Price vs Followers")
```

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




```
cor(data2017$followers, data2017$price)
```

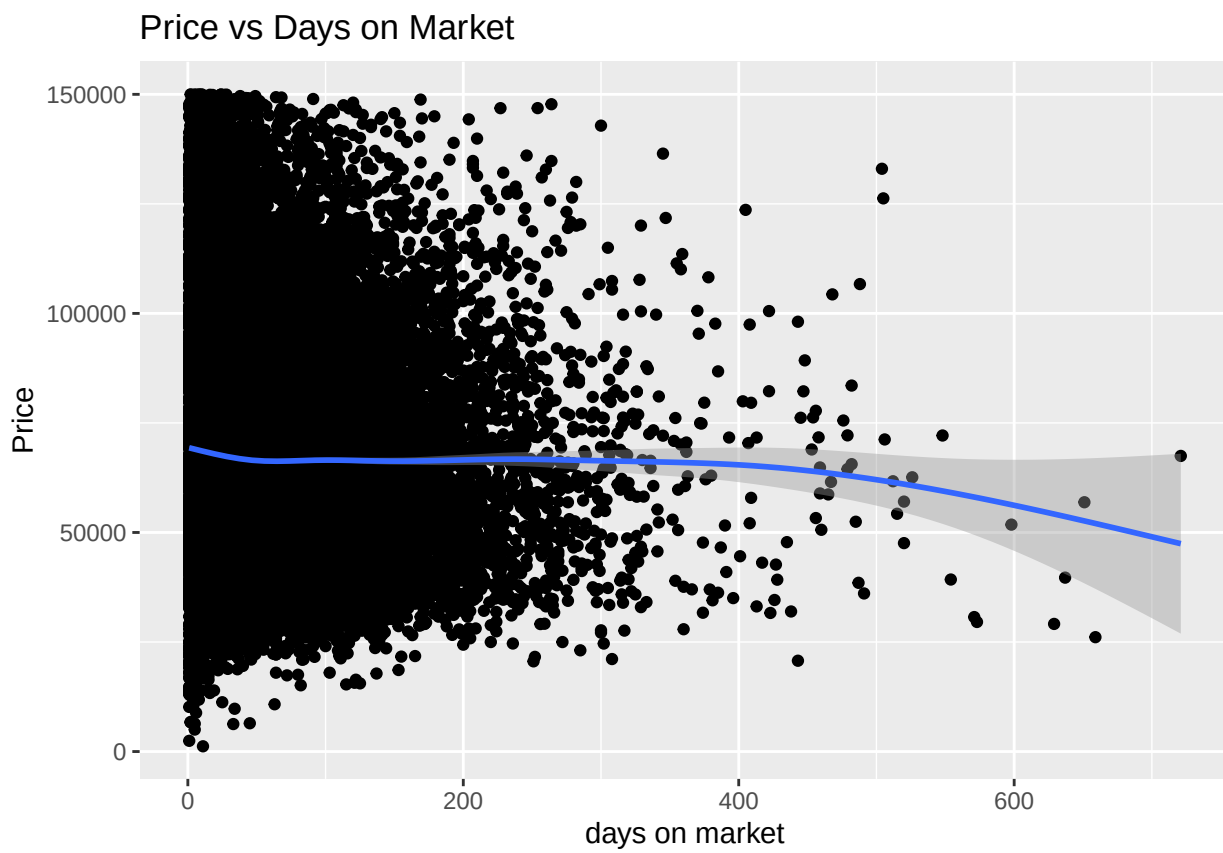
```
## [1] -0.06648905
```

The above graph does show a slight negative trend between price and followers. As followers increase, the price of the perperty is lower, although this trend does not apply to houses with very few followers because many properties ranging from low price to high price had few followers. The relationship is very week with a correlation of -0.066.

3.2.3 Days on Market vs Price

```
ggplot(data2017, aes(x = DOM, y = price)) +
  geom_point() +
  labs(x = "days on market", y = "Price", title = "Price vs Days on Market") +
  geom_smooth()
```

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




```
cor(data2017$price, data2017$DOM)
```

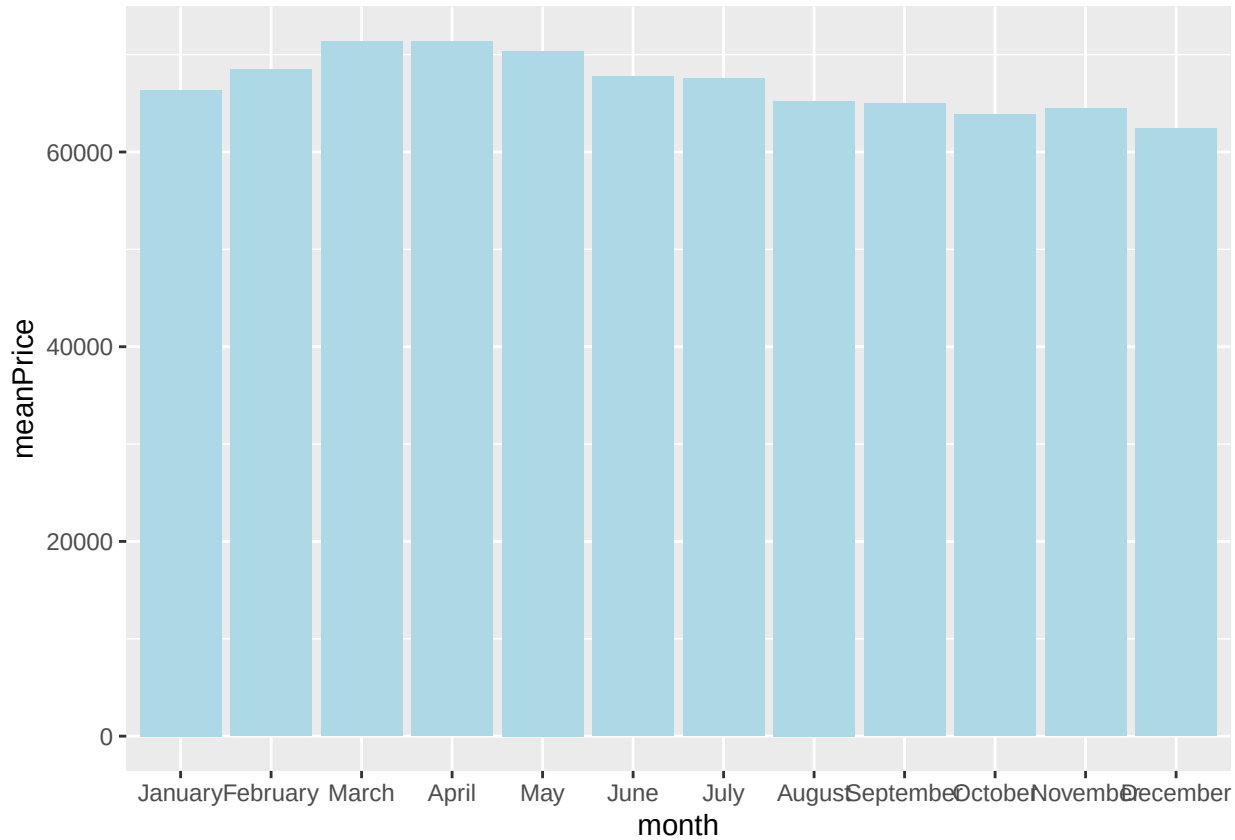
```
## [1] -0.03179496
```

There is a slight decreasing trend for price as days on market increases. The correlation is as small as -0.03 and the fitting line is almost flat, so we predict days on market does not have significant relationship with average house price.

3.2.4 Month vs Price

Overall, there is no huge differences on mean price per square on each month. As shown under the month variable section that March has the most transactions, the price is also at the peak on March. April and May also have relative higher price. After March, the price starts to decrease and reaches the bottom at December.

```
monthPrice <- data2017 %>%
  group_by(month) %>%
  summarise(meanPrice=mean(price))
ggplot(monthPrice,aes(x=month,y=meanPrice))+
  geom_col(fill = "lightblue")+
  scale_x_discrete(limits = month.name)
```




```
march_price <- data2017 %>%
  filter(month== "March") %>%
  select(price) %>% pull()
dec_price <- data2017 %>%
  filter(month== "December") %>%
  select(price) %>% pull()
t.test(march_price, dec_price)
```

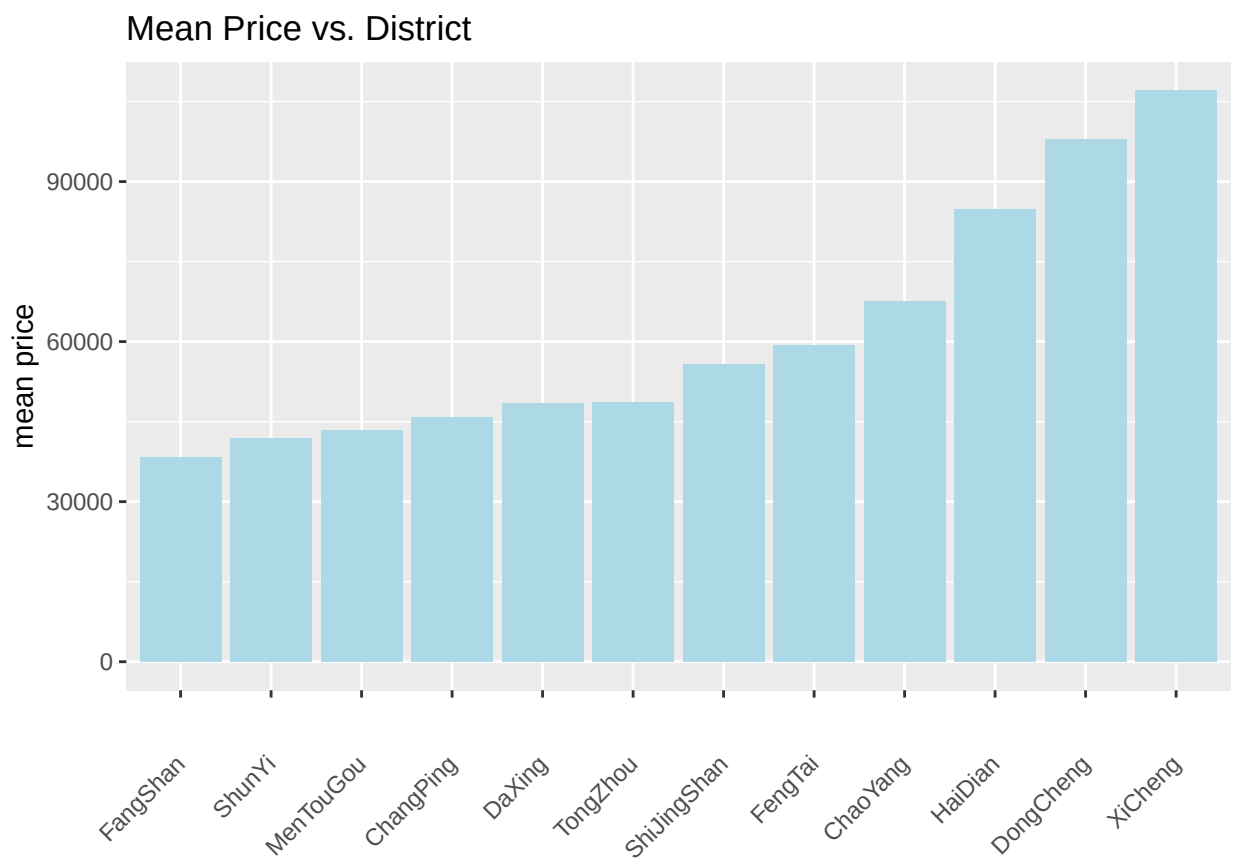
```
##
##  Welch Two Sample t-test
##
## data:  march_price and dec_price
## t = 18.802, df = 6748.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  8032.064 9901.820
## sample estimates:
## mean of x mean of y
##  71364.05  62397.11
```

The result of t-test is consistent with the graph, showing that the average house price sold in March is significantly higher than average house price sold in December.

3.2.5 District vs Price

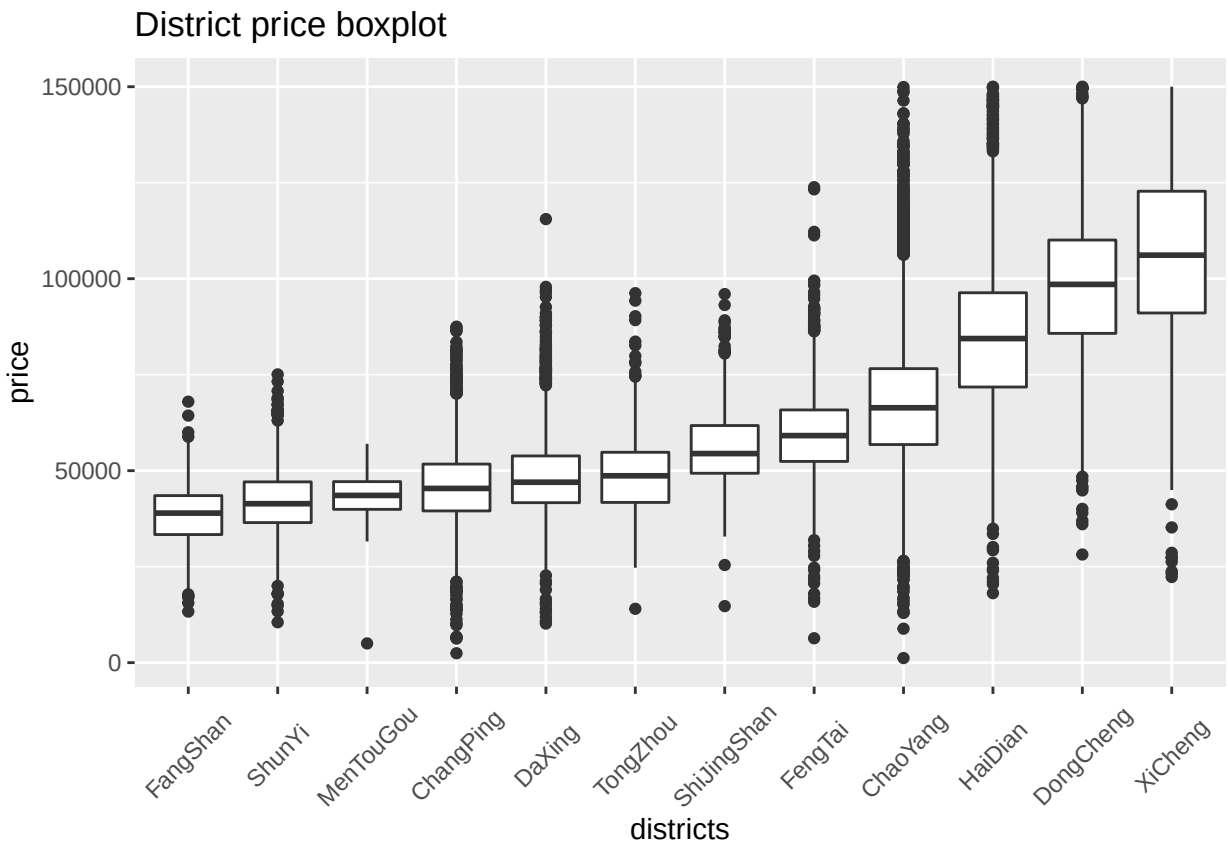
There is no doubt that the most essential factor that affects the price is the location of the property. XiCheng has the highest price which is above 105,000 yuan and DongCheng is the next, they are the most center districts of the city (Yellow). And then followed by other core districts (Red) HaiDian, ChaoYang, FengTai, and ShiJingShan. TongZhou has the highest price among other suburb districts, and then DaXing, ChangPing, MenTouGou, and ShunYi. Lastly, FangShan has the lowest price which is around 30,000 yuan. The highest price is more than 3 times higher than the lowest price. And the distribution of price is strictly following the location of the district that center districts have the highest prices, and then core districts, and lastly the suburb districts.

```
price_district <- data2017 %>%
  group_by(district) %>%
  summarise(meanPriceDistrict=mean(price))
ggplot(price_district, aes(x=reorder(district, meanPriceDistrict), y=meanPriceDistrict))+
  geom_col(fill="lightblue") +
  ggtitle("Mean Price vs. District") +ylab("mean price")+xlab("")+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1))
```



Taking a look at boxplot, XiCheng still has the highest range of price and the second highest range is DongCheng. FangShan still got the lowest price range. The distribution of median prices is the same as the distribution of mean price. It seems like both HaiDian and XiCheng have big range of house price. But XiCheng has overall higher prices than ChaoYang.

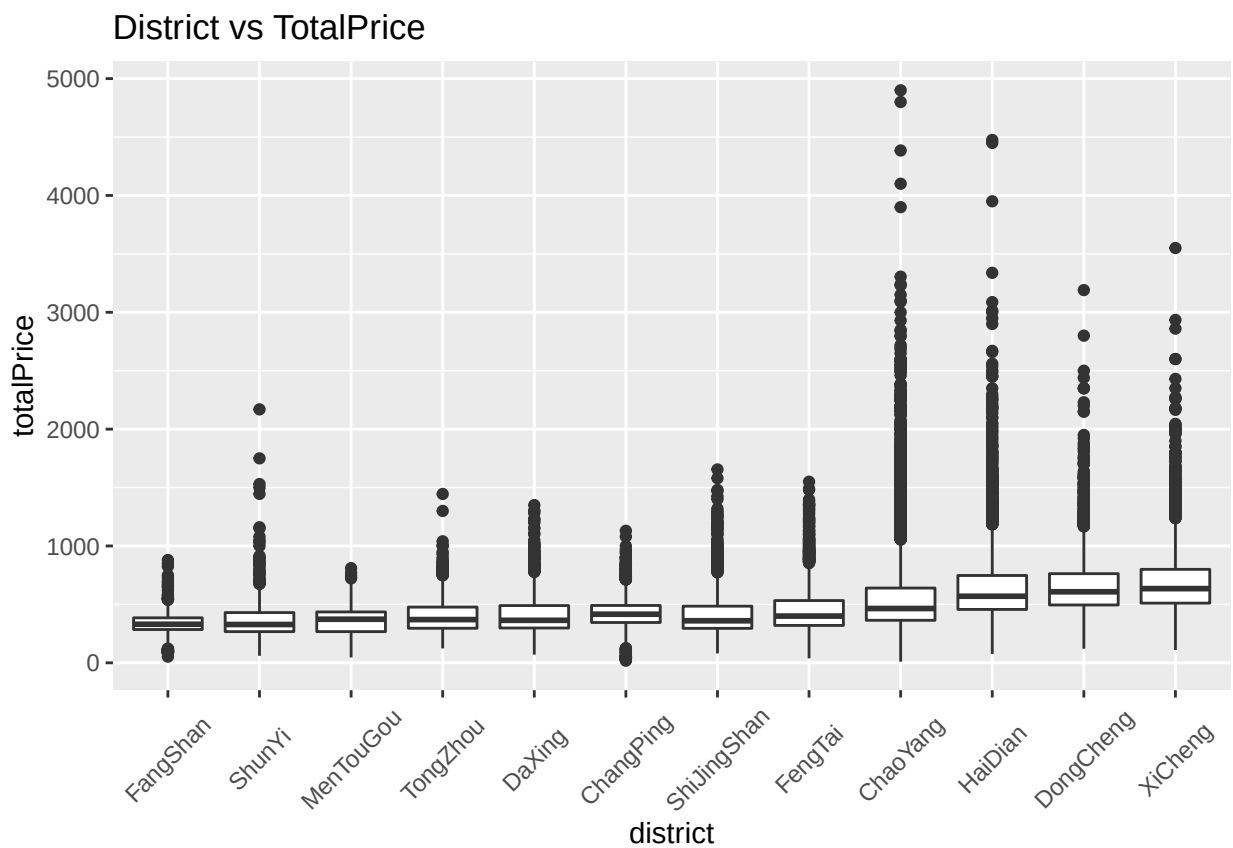
```r
ggplot(data2017, aes(x=reorder(district, price),y=price))+
  geom_boxplot()+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5)) +
  ggtitle("District price boxplot") +ylab("price")+xlab("districts")
```



3.2.6 District vs Total Price

ChaoYang has the top 2 highest total prices. FangShan has the smallest range of total prices. From the boxplot, we can see that DongCheng, HaiDian, and XiCheng are the 3 districts that have relative high total prices.

```r
ggplot(data2017, aes(x=reorder(district, totalPrice),y=totalPrice))+
  geom_boxplot()+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5)) +
  labs(title = "District vs TotalPrice", x = "district")
```

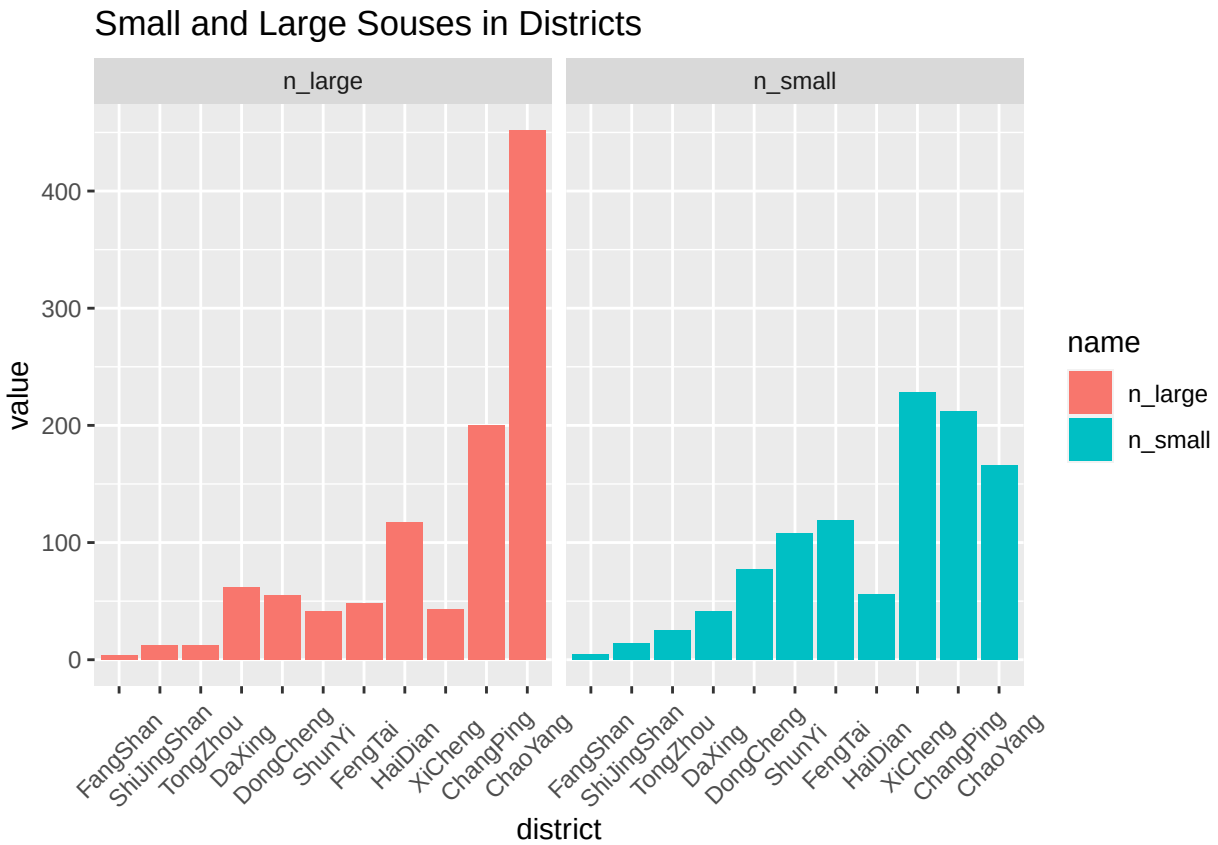


3.2.7 District vs Square

As indicated under the 'square' variable, the top 2.5% of houses sold have area over 166.6 square meters. Let's define them as "large houses". The bottom 2.5% of houses sold have less than 37.27 square meters. Let's define them as "small houses".

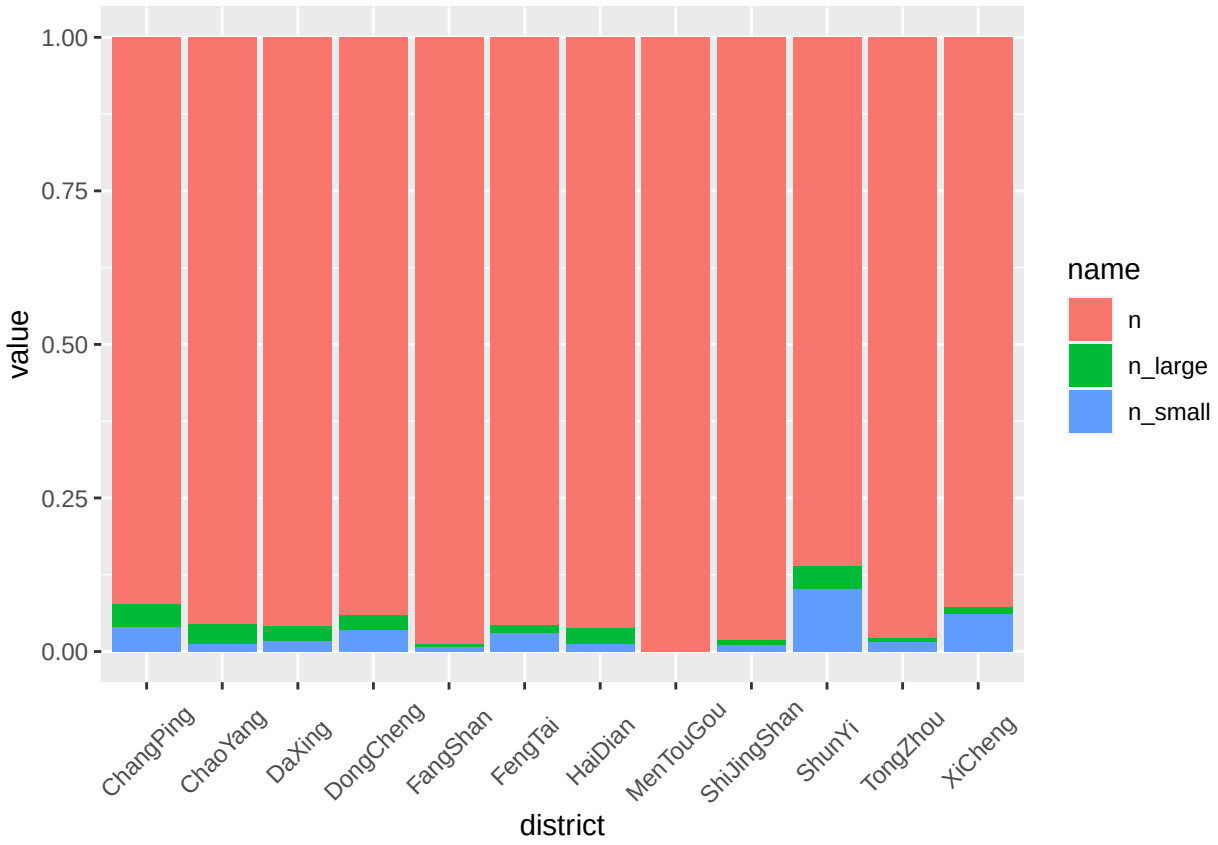
Districts with most large and small houses ChaoYang and ChangPing has the significant highest amounts of large houses sold and XiCheng has the significant highest amounts of small houses sold.

```
large <- data2017 %>%
  filter(square>=166.6) %>% # number of large houses in each district
  count(district)
small <- data2017 %>%
  filter(square<=37.27) %>% # number of small houses in each district
  count(district)
full1 <- small %>%
  full_join(large,by="district",suffix = c("_small","_large"))
pivot <- full1 %>%
  pivot_longer(col=c("n_large","n_small"))

ggplot(pivot,aes(x=reorder(district, value),y=value,fill=name))+
  geom_col()+
  facet_wrap(~name)+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5)) +
  labs(title = "Small and Large Souses in Districts", x = "district")
```



Proportion of large and small houses within each district The majority of houses for all districts are middle sized. ChangPing and ChaoYang has the highest proportion of large houses followed by ShunYi which has the largest proportion of small houses. MenTouGou did not have extreme large and small houses.

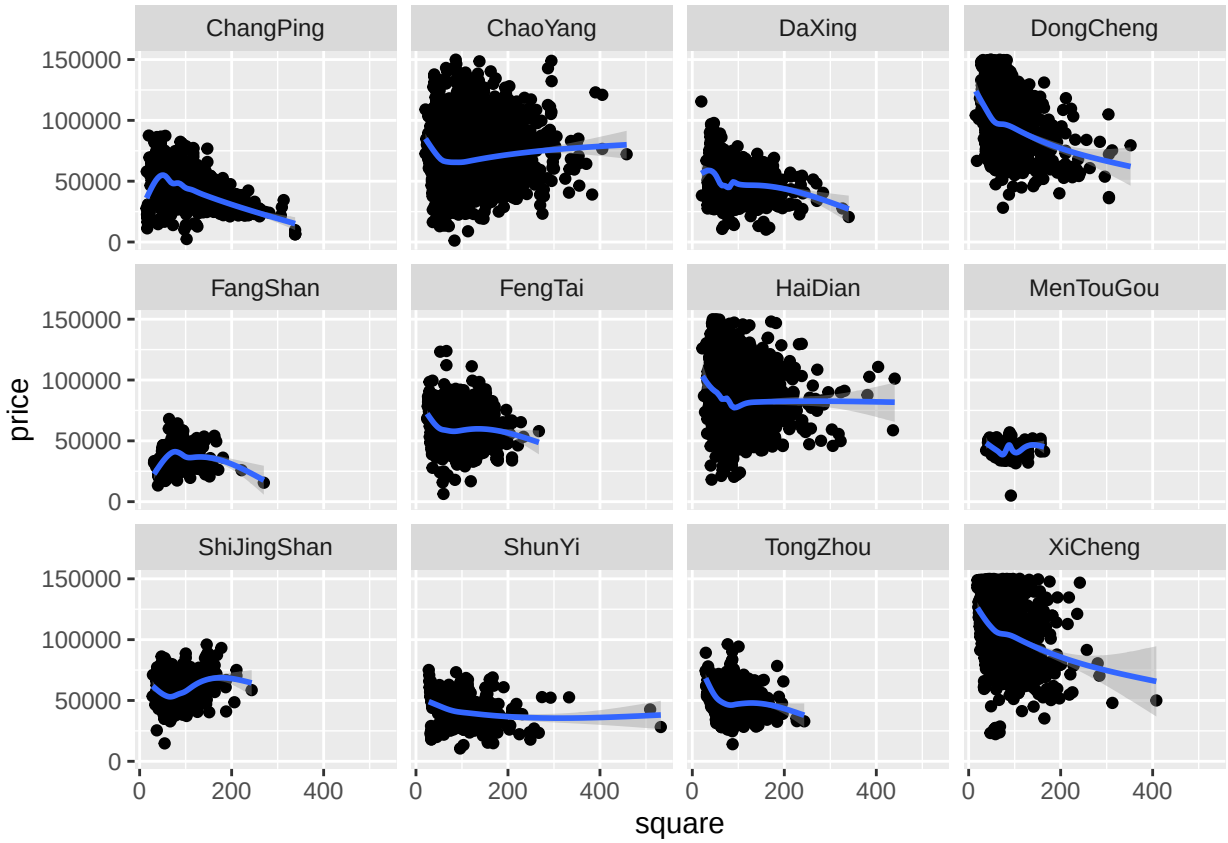
```
middle <- data2017 %>%
  filter(square>=38.63 & square<=149.97) %>%
  count(district)
full2 <- full1 %>%
  full_join(middle,by="district")
pivot2 <- full2 %>%
  pivot_longer(col=c("n_large","n_small","n")) %>%
  replace(is.na(.), 0)
ggplot(pivot2,aes(x=district,y=value,fill=name))+
  geom_col(position="fill")+ # stacked proportion barplot
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5))
```



3.2.8 Square vs Price

```
ggplot(data2017, aes(x=square, y= price)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~district)
```

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

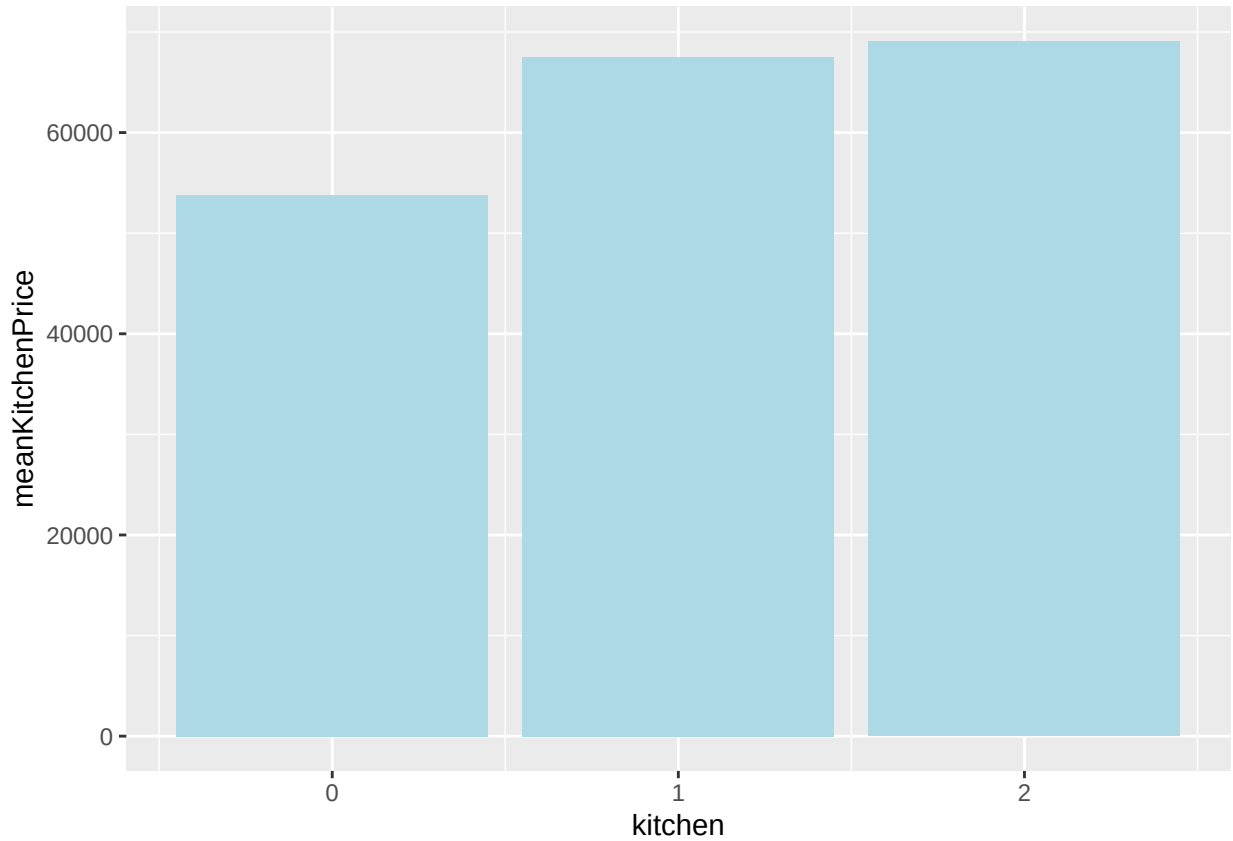


Even though the price is per square meter, it still has a relationship with square when we plot them by district. For most districts, price per square meter decreases as square increases, except for ChaoYang and ShiJingShan. Properties in medium sizes are more expensive probably because the demand is higher.

3.2.9 Kitchen vs Price

Houses with more kitchens have higher mean prices. The difference between 0 and 1 kitchen is about 10,000 yuan, but the difference between 1 and 2 kitchens is not significant. Let's compare how the number of kitchens affect prices within each district.

```
meanKitchen <- data2017 %>%
  group_by(kitchen) %>%
  summarise(meanKitchenPrice=mean(price))
ggplot(meanKitchen, aes(x=kitchen, y=meanKitchenPrice))+
  geom_col(fill="lightblue")
```

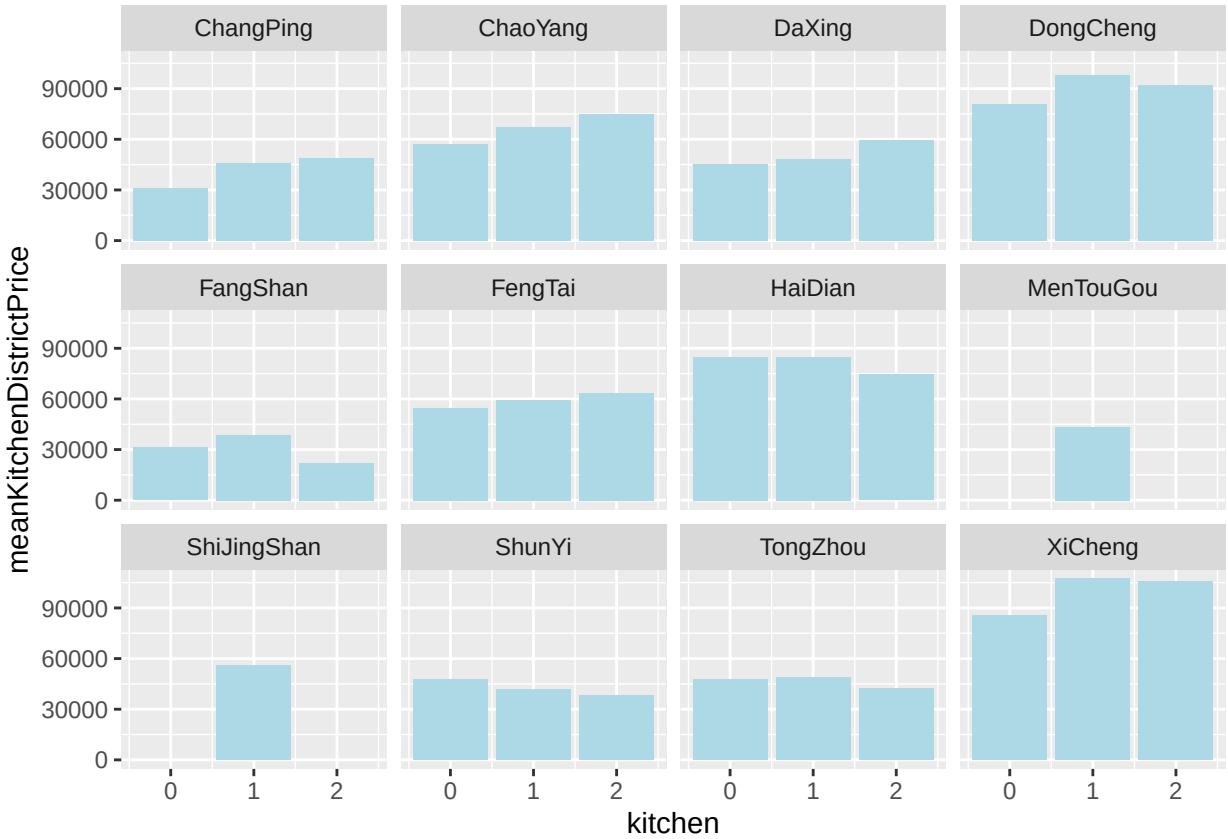


For most districts, houses with 1 kitchen have the highest mean price. Except for ChangPing, ChaoYang, and FengTai, houses with more kitchens have higher mean prices. HaiDian is the special case that the mean house prices keep decreasing as the number of kitchens increase.

```
meanKitchenDistrict <- data2017 %>%
  group_by(kitchen,district) %>%
  summarise(meanKitchenDistrictPrice=mean(price))
```

```
## `summarise()` has grouped output by 'kitchen'. You can override using the
## `.groups` argument.
```

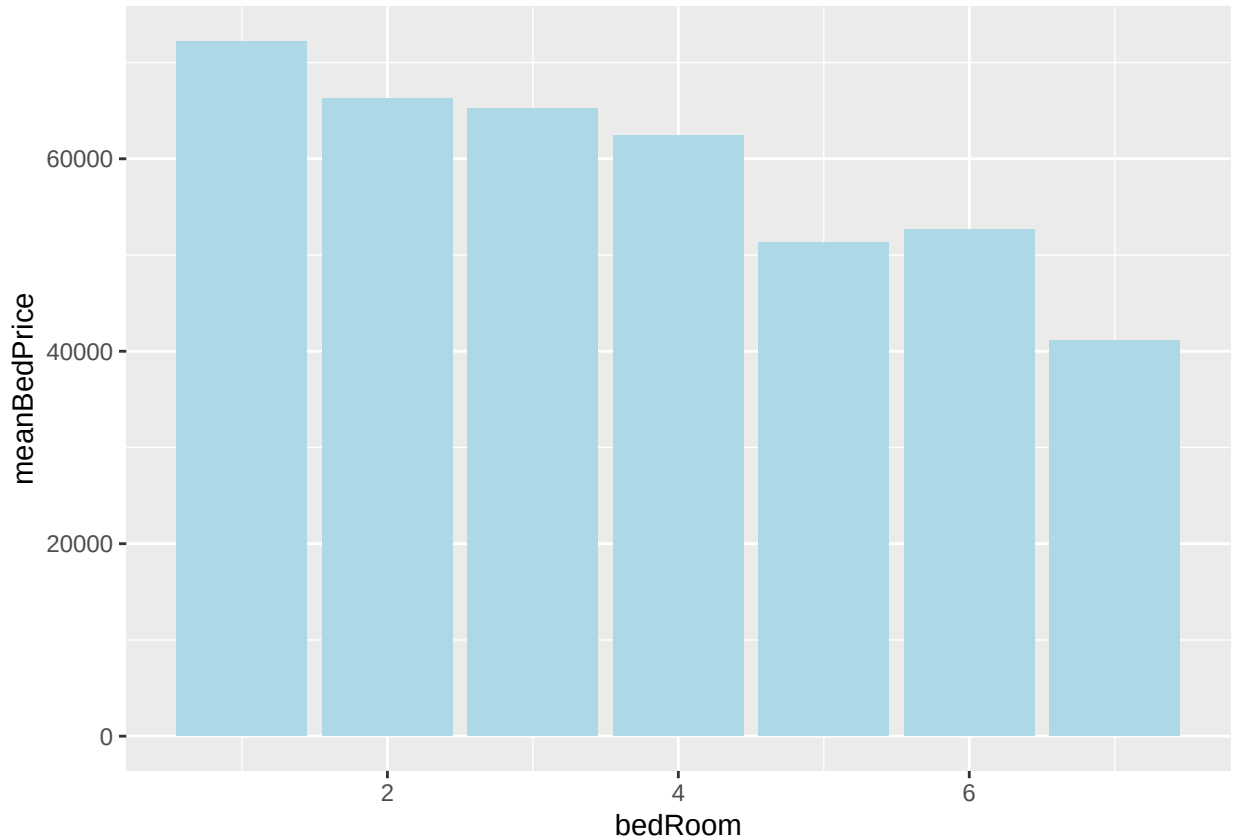
```
ggplot(meanKitchenDistrict, aes(x=kitchen, y=meanKitchenDistrictPrice))+
  facet_wrap(~district)+
  geom_col(fill="lightblue")
```



3.2.10 Bedroom vs Price

The more bedrooms indicates the lower mean house price is. However, simply look at the mean price for all houses is not enough because district plays the key role. Let's compare how the number of bedrooms affect price within each district.

```
meanBed <- data2017 %>%
  group_by(bedRoom) %>%
  summarise(meanBedPrice=mean(price))
ggplot(meanBed, aes(x=bedRoom, y=meanBedPrice))+
  geom_col(fill="lightblue")
```

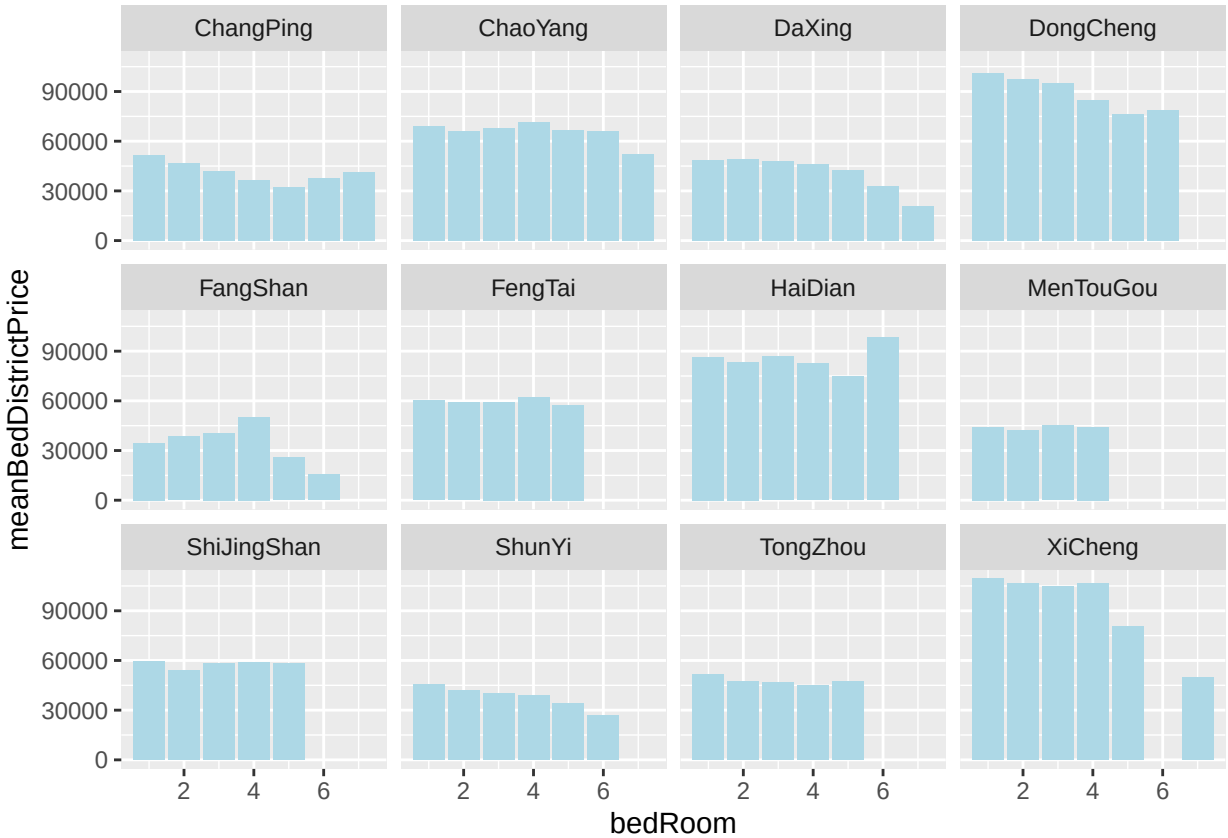


There is also a trend observed for some districts that the more bedrooms the lower price is. ChangPing is an exception, where lots of huge maisons are located. It could be popular in ChangPing to have many bedrooms in a house. HaiDian is another exception where price increases for 6 bedroom, which could be due to that it's rare to see such large houses in HaiDian because it's one of the core districts.

```
meanBedDistrict <- data2017 %>%
  group_by(bedRoom,district) %>%
  summarise(meanBedDistrictPrice=mean(price))
```

```
## `summarise()` has grouped output by 'bedRoom'. You can override using the
## `.groups` argument.
```

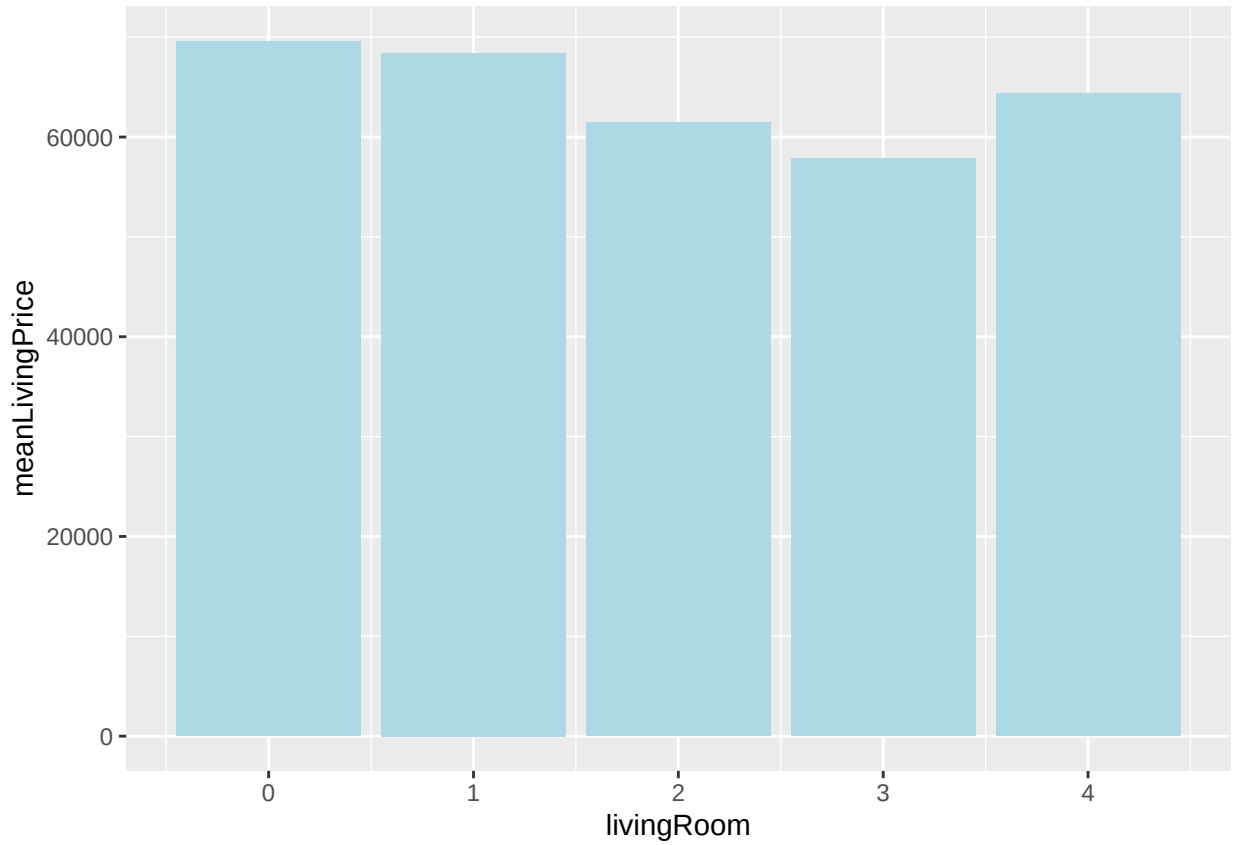
```
ggplot(meanBedDistrict, aes(x=bedRoom, y=meanBedDistrictPrice))+
  facet_wrap(~district)+
  geom_col(fill="lightblue")
```



3.2.11 Living room vs Price

The graph shows that there is a decreasing trend for price as number of bedrooms increase. Houses sold with no living room have the highest mean price. As shown under the living room variable section, most houses have 1 living room. The price of houses with 1 living room is slightly lower. However, the mean price slightly increases for 4-bedrooms. Let's compare how the number of living rooms affect house price within each district.

```
meanLiving <- data2017 %>%
  group_by(livingRoom) %>%
  summarise(meanLivingPrice=mean(price))
ggplot(meanLiving, aes(x=livingRoom, y=meanLivingPrice))+
  geom_col(fill="lightblue")
```

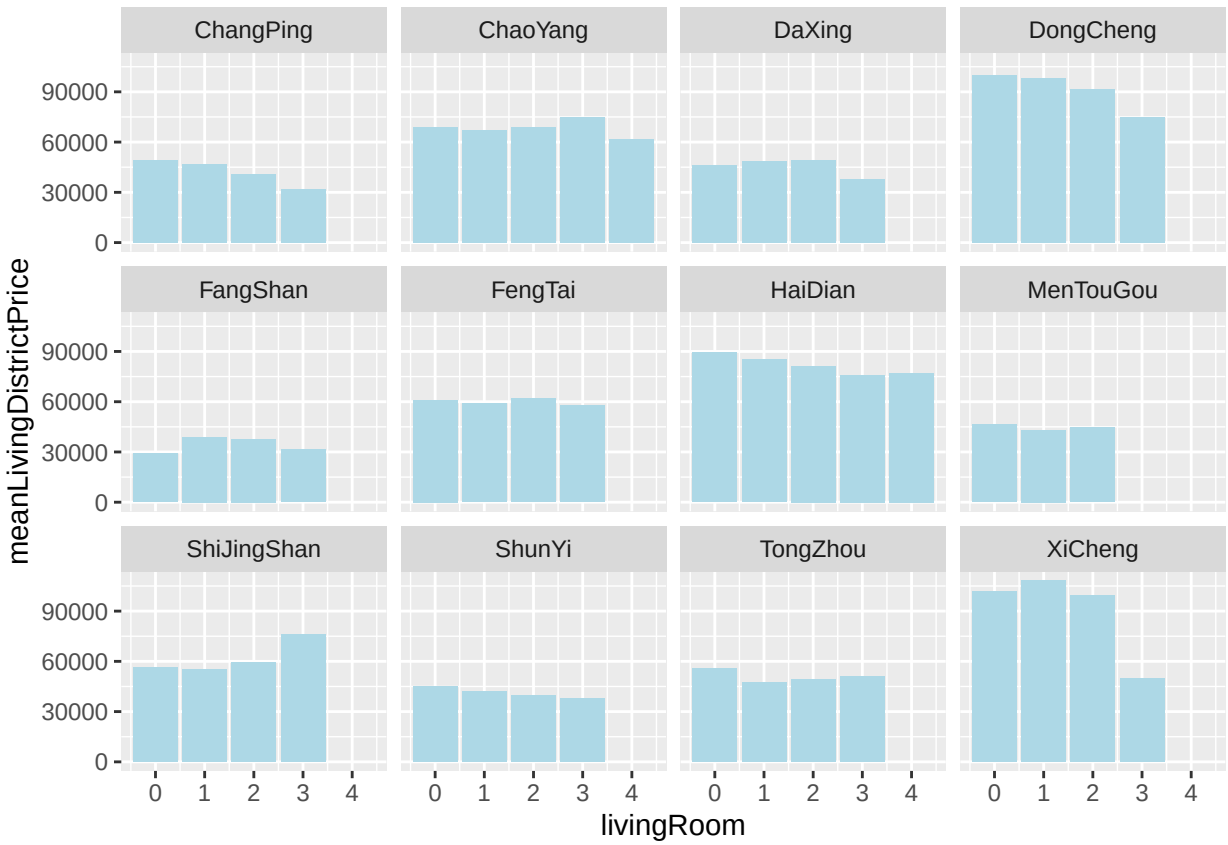


For most district, houses with 0 or 1 living room still have relative higher price. Except for ShiJingShan, which has higher mean price for houses with 4 bedrooms, mainly because ShiJingShan is the smallest district in Beijing. It's more expensive to buy big houses in ShiJingShan.

```
meanLivingDistrict <- data2017 %>%
  group_by(district,livingRoom) %>%
  summarise(meanLivingDistrictPrice=mean(price))
```

```
## `summarise()` has grouped output by 'district'. You can override using the
## `.groups` argument.
```

```
ggplot(meanLivingDistrict, aes(x=livingRoom, y=meanLivingDistrictPrice))+
  facet_wrap(~district)+
  geom_col(fill="lightblue")
```



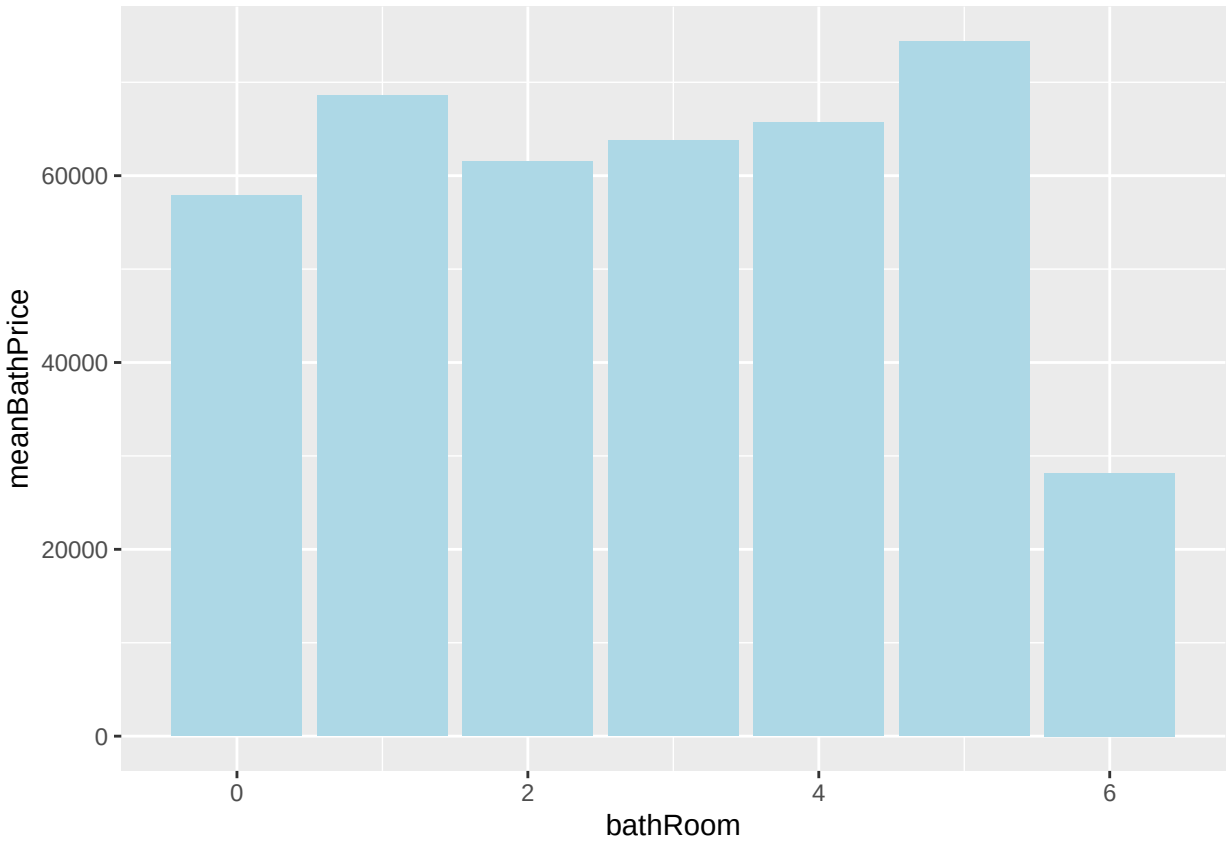
3.2.12 Bathroom vs Price

```
table(data2017$bathRoom==0)
```

```
##
## FALSE  TRUE
## 41924    77
```

Houses with 1 bathroom have the highest mean price and houses with 5 bathrooms have the second highest price. Price slightly changes as the number of bathrooms increase from 2 to 4. There is a sudden drop on price for houses with 6 bathrooms. It has mean price below 30,000 yuan, which is nealy 50% lower than the second lowest price, which is houses with no bathroom.

```
meanBath <- data2017 %>%
  group_by(bathRoom) %>%
  summarise(meanBathPrice=mean(price))
ggplot(meanBath, aes(x=bathRoom, y=meanBathPrice))+
  geom_col(fill="lightblue")
```

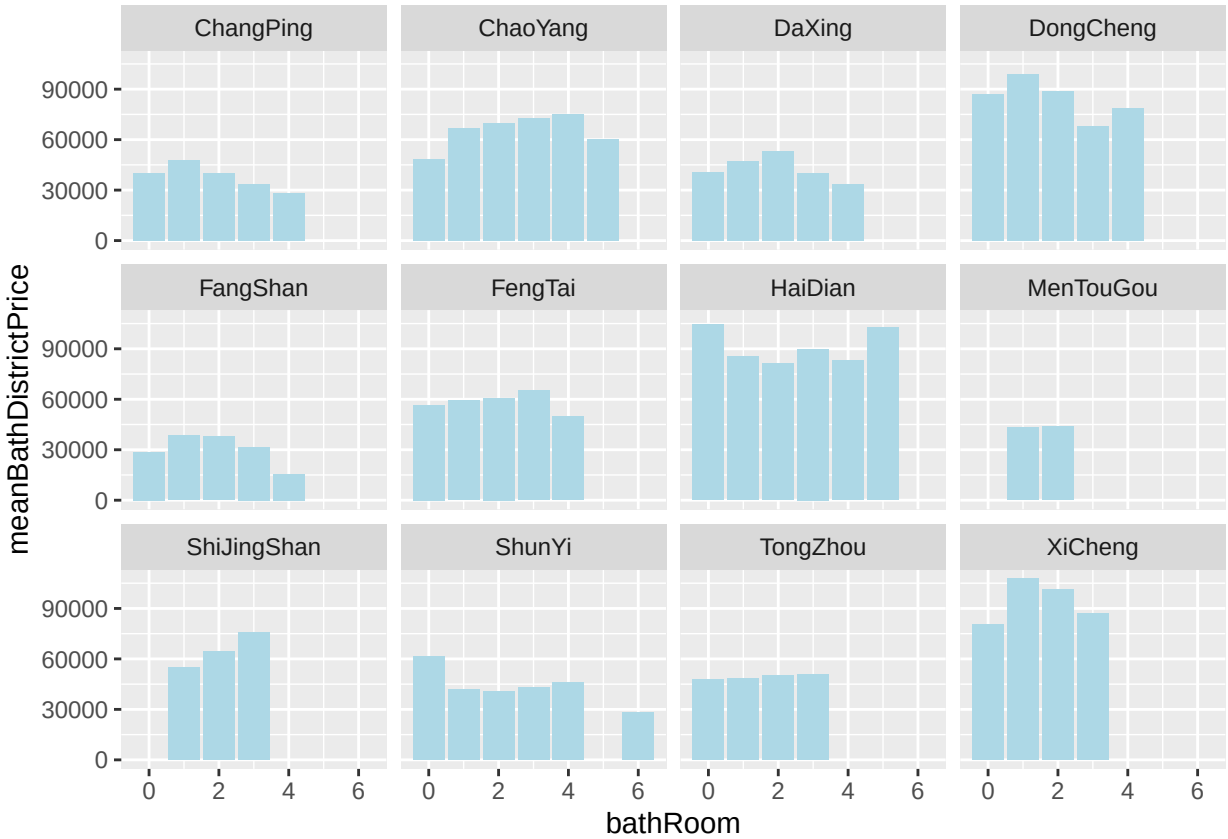


Let's compare how number of bathrooms affect price within each district. For most districts, houses with 1 or 5 bathroom have the highest price. HaiDian is the special case that houses with either no bathroom or the maximum number of bathrooms have the highest prices. For ShiJingShan and TongZhou, more bathrooms indicate higher prices.

```
meanBathDistrict <- data2017 %>%
  group_by(bathRoom, district) %>%
  summarise(meanBathDistrictPrice=mean(price))
```

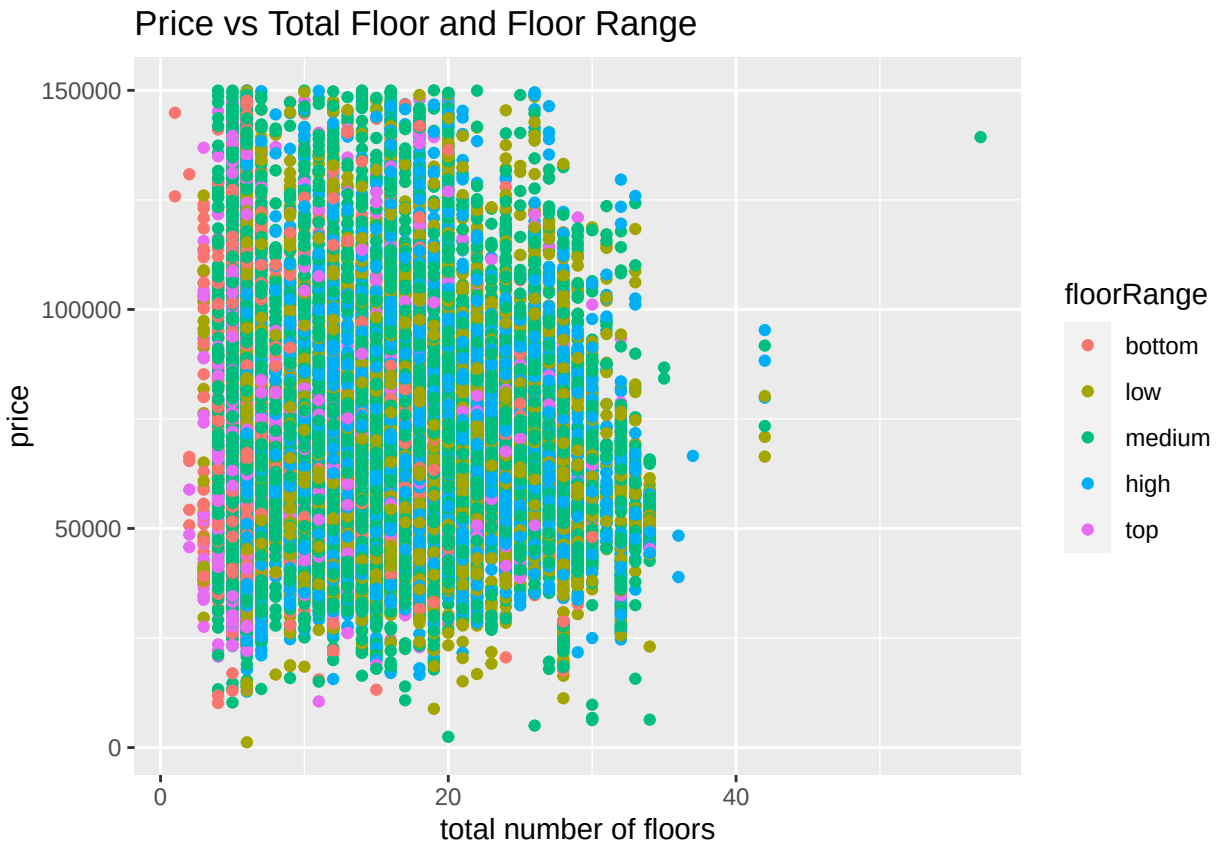
```
## `summarise()` has grouped output by 'bathRoom'. You can override using the
## `.groups` argument.
```

```
ggplot(meanBathDistrict, aes(x=bathRoom, y=meanBathDistrictPrice))+
  geom_col(fill="lightblue")+
  facet_wrap(~district)
```



3.2.13 Ladder Ratio and Floor Range vs Price

```
ggplot(data2017, aes(x = totalFloor, y = price, color = floorRange)) +
  geom_point()+
  labs(x = "total number of floors", title = "Price vs Total Floor and Floor Range")
```

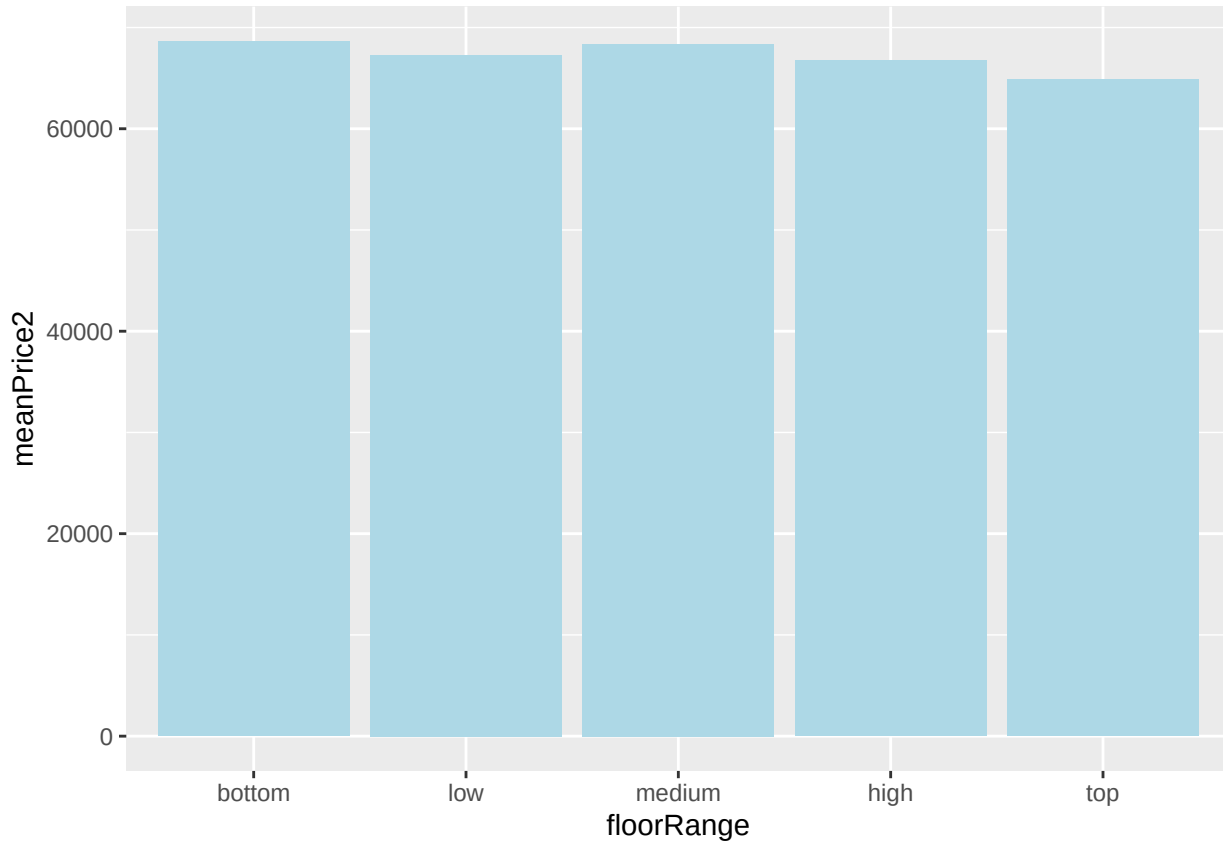


It's hard to see any relationships between these three from this graph. However, the price drops when the total floor is above 30, with one exception, which is the property that has 63 floors. In general, the price per square meter is slightly lower for properties located in buildings higher than 30 floors. Next, lets' take a look at them with price separately.

3.2.14 Floor Range vs Price

There is a slight decreasing trend as properties are located higher. The higher the property is located, the lower the average price is. Medium and bottom floor ranges seem to have relative higher price whereas top floor range has the lowest.

```
data2017 %>%
  group_by(floorRange) %>%
  summarise(meanPrice2=mean(price))%>%
  ggplot(aes(x=floorRange, y=meanPrice2))+
  geom_col(fill="lightblue")
```




```
bottom_price <- data2017 %>%
  filter(floorRange== "bottom") %>%
  select(price) %>% pull()
top_price <- data2017 %>%
  filter(floorRange== "top") %>%
  select(price) %>% pull()
t.test(bottom_price, top_price)
```

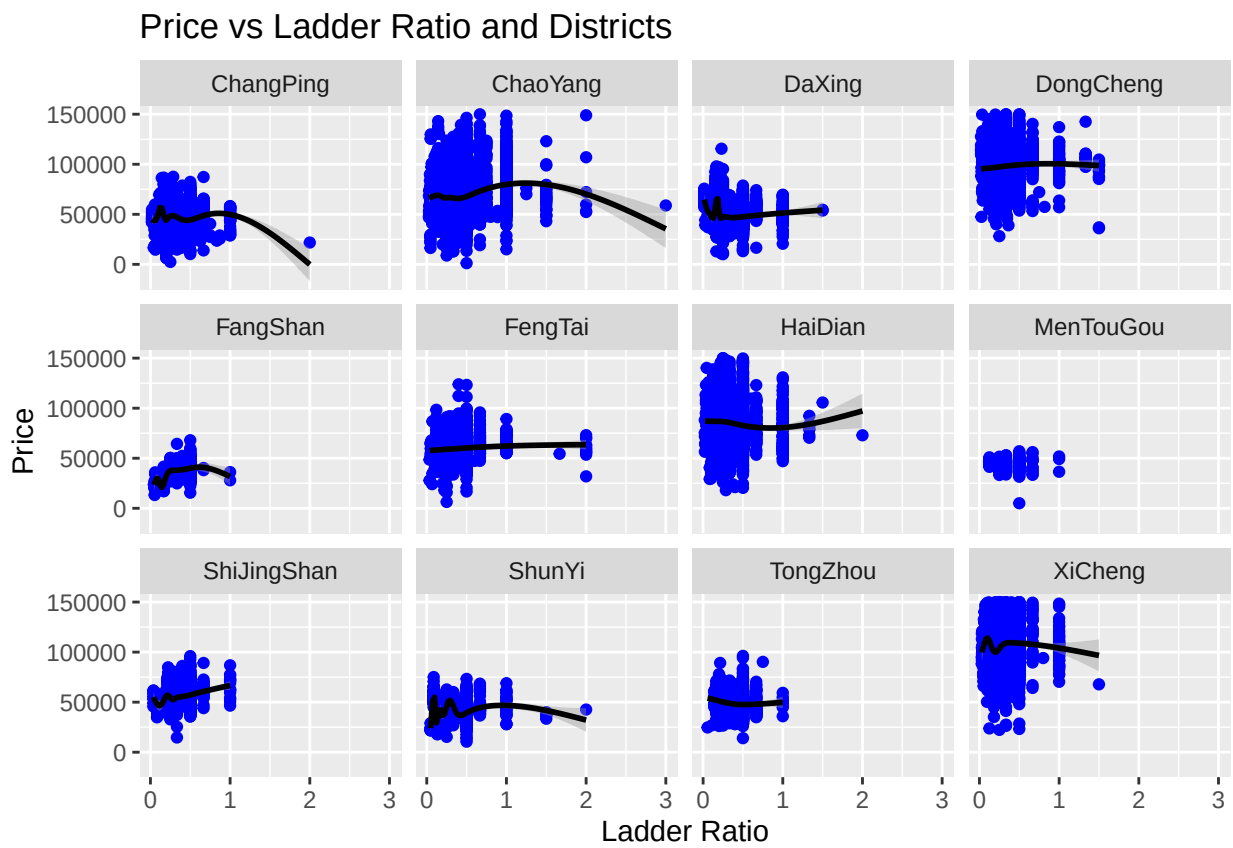
```
##
##  Welch Two Sample t-test
##
## data:  bottom_price and top_price
## t = 6.6676, df = 7131.9, p-value = 2.795e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2677.872 4908.211
## sample estimates:
## mean of x mean of y
##  68632.45  64839.41
```

The result of t-test is consistent with the graph, showing that the average price of bottom properties is significantly different from average price of top houses.

3.2.15 Ladder Ratio vs District vs Price

```
ggplot(data2017, aes(x = ladderRatio, y = price)) +
  geom_point(color = "blue") +
  facet_wrap(~district)+
  labs(x = "Ladder Ratio", y = "Price", title = "Price vs Ladder Ratio and Districts") +
  geom_smooth(color = "black")
```

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

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




```
cor(data2017$price, data2017$ladderRatio)
```

```
## [1] -0.1313789
```

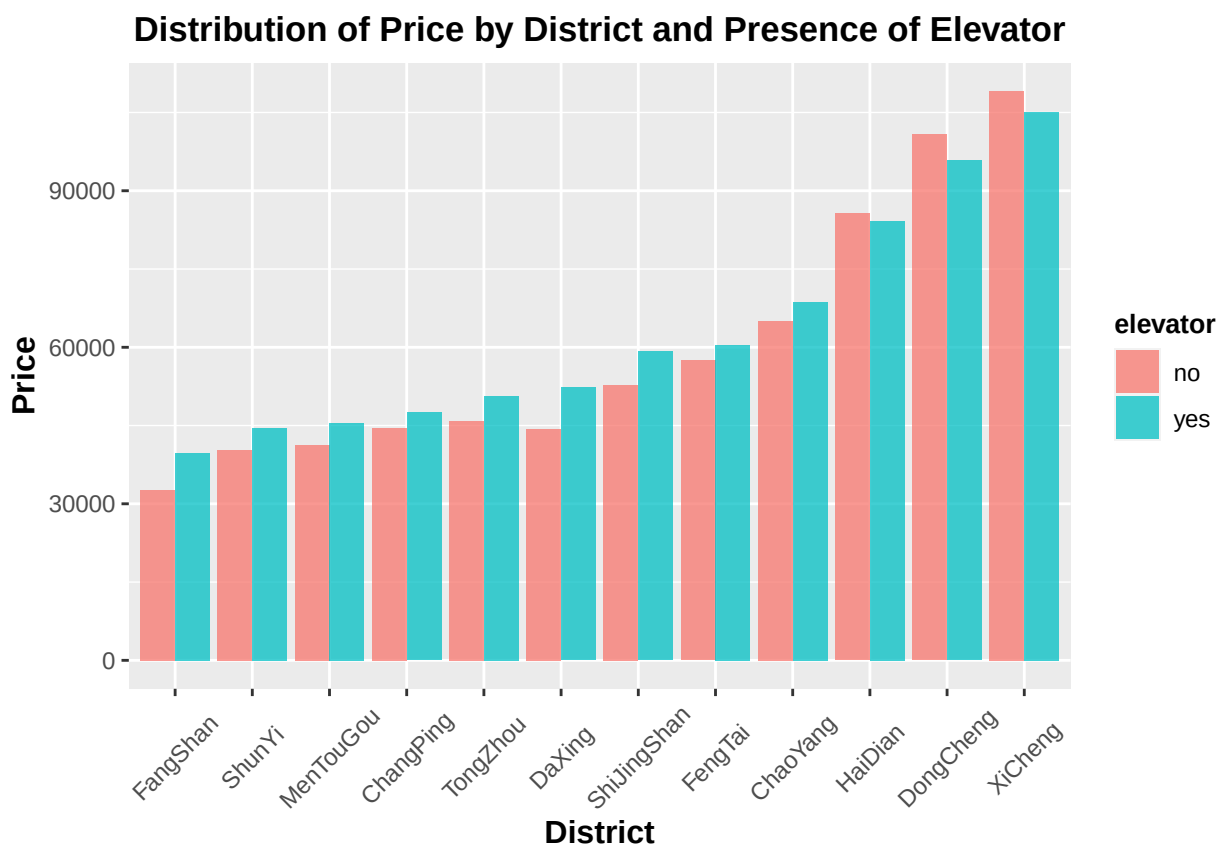
There isn't a significant relationship between ladder ratio and price. The correlation is only -0.13. Therefore, we predict that the ladder ratio will not significantly influence price.
Most districts have ladder ratio between 0 to 1, while ChaoYang has the most properties with higher than 1 ladder ratio.

3.2.16 Elevator vs District vs Price

From the distribution, we can see that most of the listings are more expensive when there is an elevator available in the building with the exception of HaiDian, DongCheng, XiCheng.

```
data2017 %>% group_by(district, elevator) %>%
  summarise(mean_price = mean(price)) %>%
  ggplot(aes(x = reorder(district, mean_price), y = mean_price, fill = elevator)) +
  geom_bar(stat = "identity", position = position_dodge(), alpha = 0.75) +
  ggtitle("Distribution of Price by District and Presence of Elevator") +
  labs(x = "District", y = "Price") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.title.x = element_text(face="bold", size = 12),
        axis.title.y = element_text(face="bold", size = 12),
        legend.title = element_text(face="bold", size = 10))+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5))
```

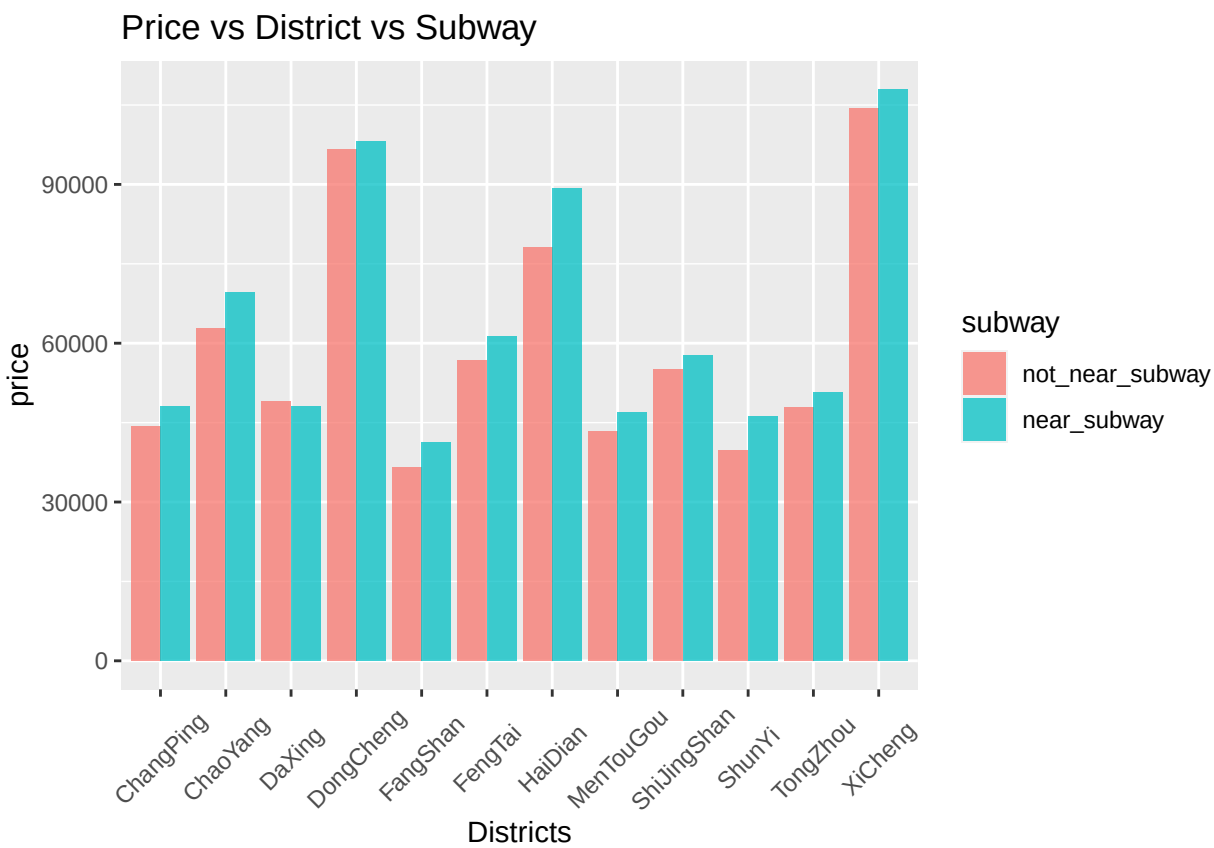
```
## `summarise()` has grouped output by 'district'. You can override using the
## `.groups` argument.
```



3.2.17 Subway vs District vs Price

Subway is one of the most important transportation in Beijing. Near subway or not also shows if the property is located in a busy area.

```
data2017 %>%
  group_by(district, subway) %>%
  summarise(average_price = mean(price)) %>%
  ggplot(aes(x = district, y = average_price, fill = subway)) +
  geom_col(stat = "identity", position = position_dodge(), alpha = 0.75)  +
  labs(x = "Districts", y = "price", title = "Price vs District vs Subway") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5))
```

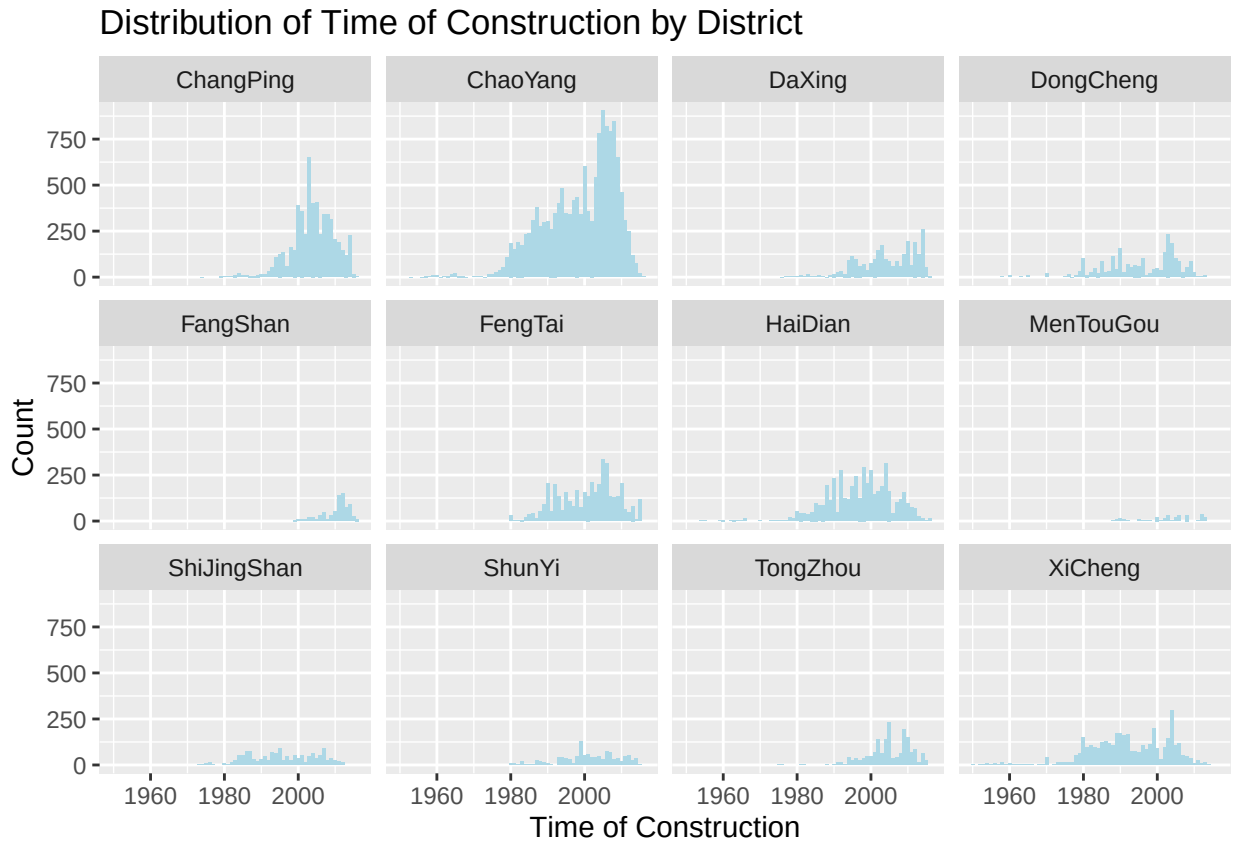


Is this affected by district? Let's see how a house is near subway or not will affect the price within each district. The conclusion is that for most districts, mean price is higher if the house is near subway. However, DaXing is an exception where mean price seems to be lower for properties that are near a subway.

3.2.18 Construction time vs District

From the distribution of time of construction by district, we see that for the district of ChaoYang, there is a gradual increase in the number of listings being constructed since 1950 and then a sharp decrease after it reached its peak.
For the district of ChangPing, there is a sharper increase in the number of listings being constructed since 1950 compared to ChaoYang but a more gradual decrease after it reached its peak.
For the other districts, the number of listings being constructed changes less drastically compared to ChaoYang and ChangPing.

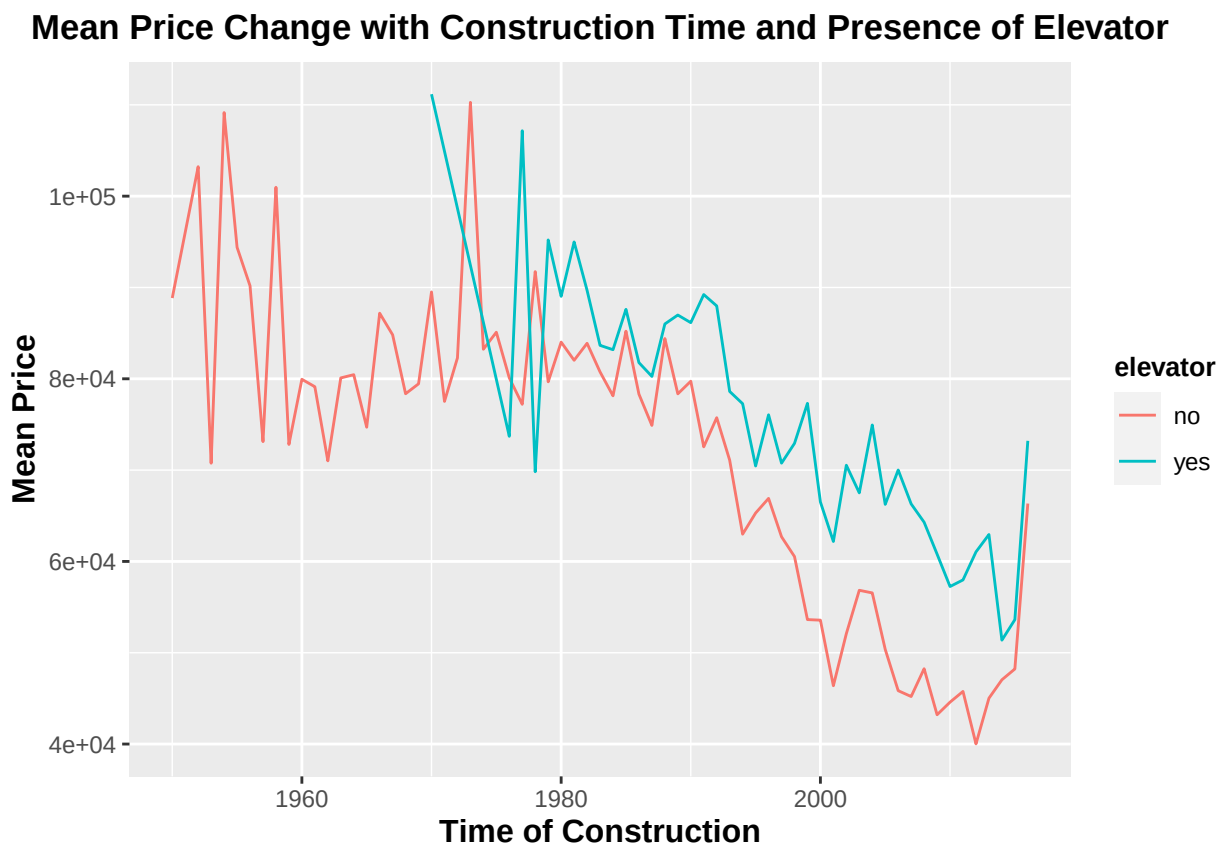
```
ggplot(data2017, aes(x = constructionTime)) +
  geom_bar(fill = "lightblue") +
  ggtitle("Distribution of Time of Construction by District") + xlab("Time of Construction") + ylab("Co
  facet_wrap(~ district)
```



3.2.19 Construction time vs Elevator vs Price

```
data2017 %>% group_by(constructionTime, elevator) %>%
  summarise(mean_price = mean(price)) %>%
  ggplot(aes(x = constructionTime, y = mean_price, color = elevator)) +
  geom_line() +
  ggtitle("Mean Price Change with Construction Time and Presence of Elevator") +
  labs(x = "Time of Construction", y = "Mean Price") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.title.x = element_text(face="bold", size = 12),
        axis.title.y = element_text(face="bold", size = 12),
        legend.title = element_text(face="bold", size = 10))
```

```
## `summarise()` has grouped output by 'constructionTime'. You can override using
## the `.groups` argument.
```

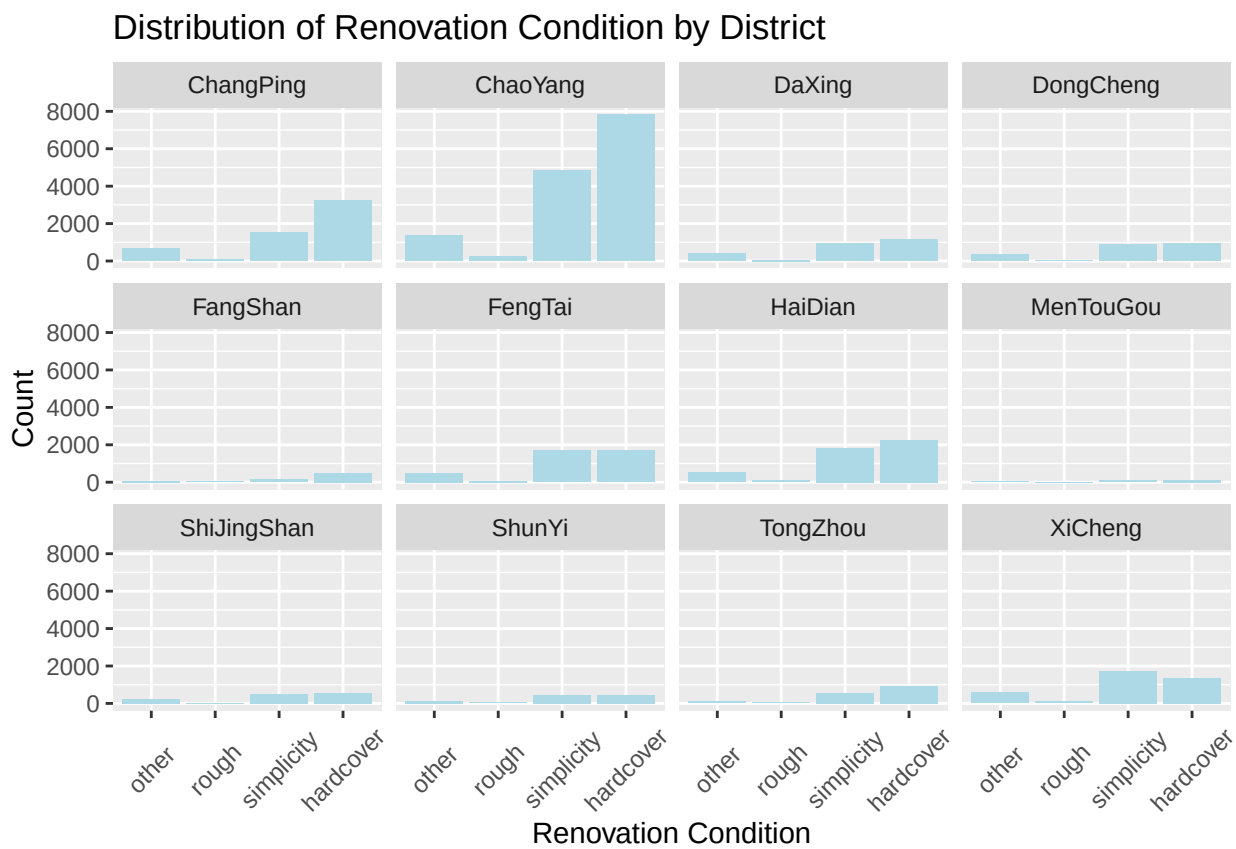


From the graph, we can see that as properties are constructed in later years, the mean price drops. In othe words, the newer the property is, the lower the average price. For some reason, houses construced during 1970-1980 are exceptionally expensive. The presence of elevator emerged for properties built after 1970 in Beijing. Consistent with previous findings, mean price is higher for properties with elevators compared to those without elevator that are constructed in the same year.

3.2.20 Renovation Condition vs District

From the distribution of renovation condition by district, we see that ChaoYang has the highest number of listings with the *hardcover* rating. ChaoYang had a lot of well-renovated properties being sold compared to other districts because ChaoYang is one of the core financial districts where many wealthy poeple live.

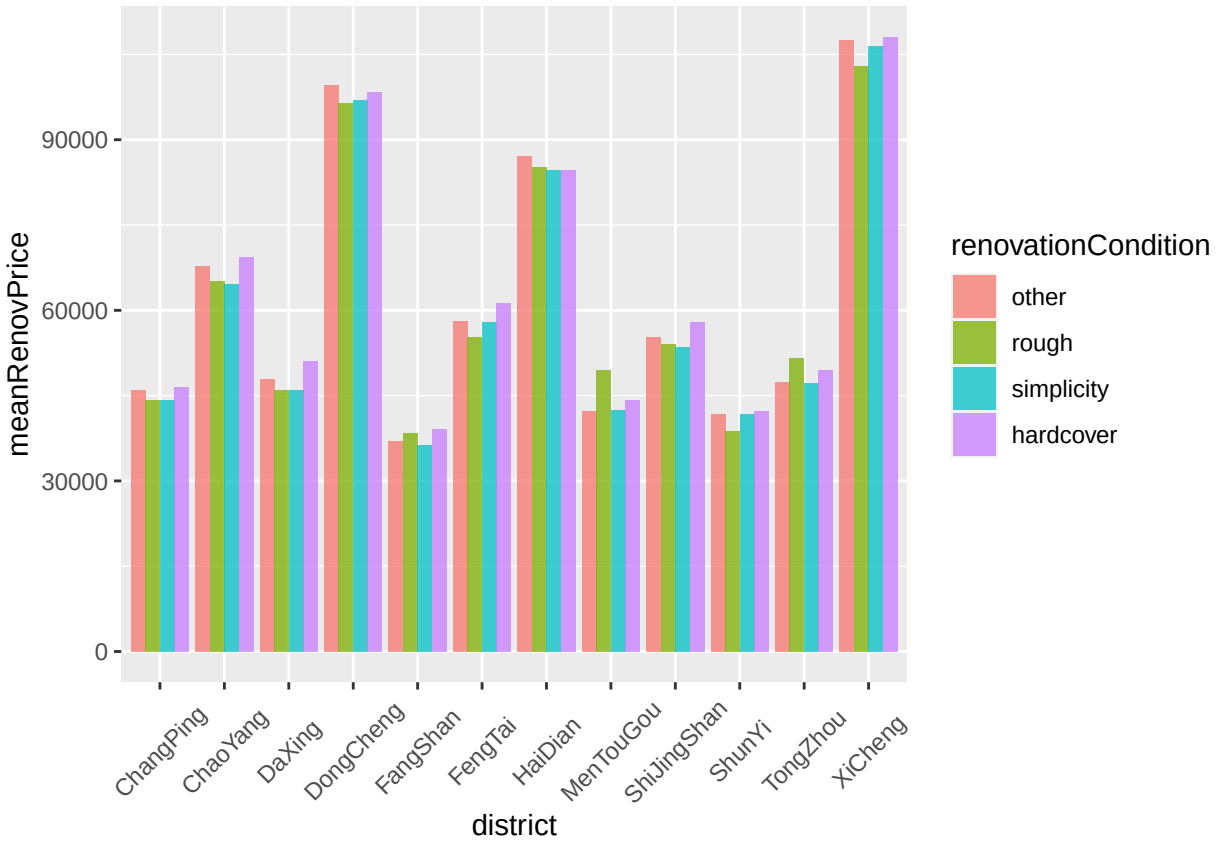
```
ggplot(data2017, aes(x = renovationCondition)) +
  geom_bar(fill = "lightblue") +
  ggtitle("Distribution of Renovation Condition by District") + xlab("Renovation Condition") + ylab("Co
  facet_wrap(~ district) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5))
```



3.2.21 Renovation Condition vs District vs Price

For most districts, hardcover renovation has higher mean price than rough renovation condition with the exception of TongZhou, MenTouGou, and Haidian.

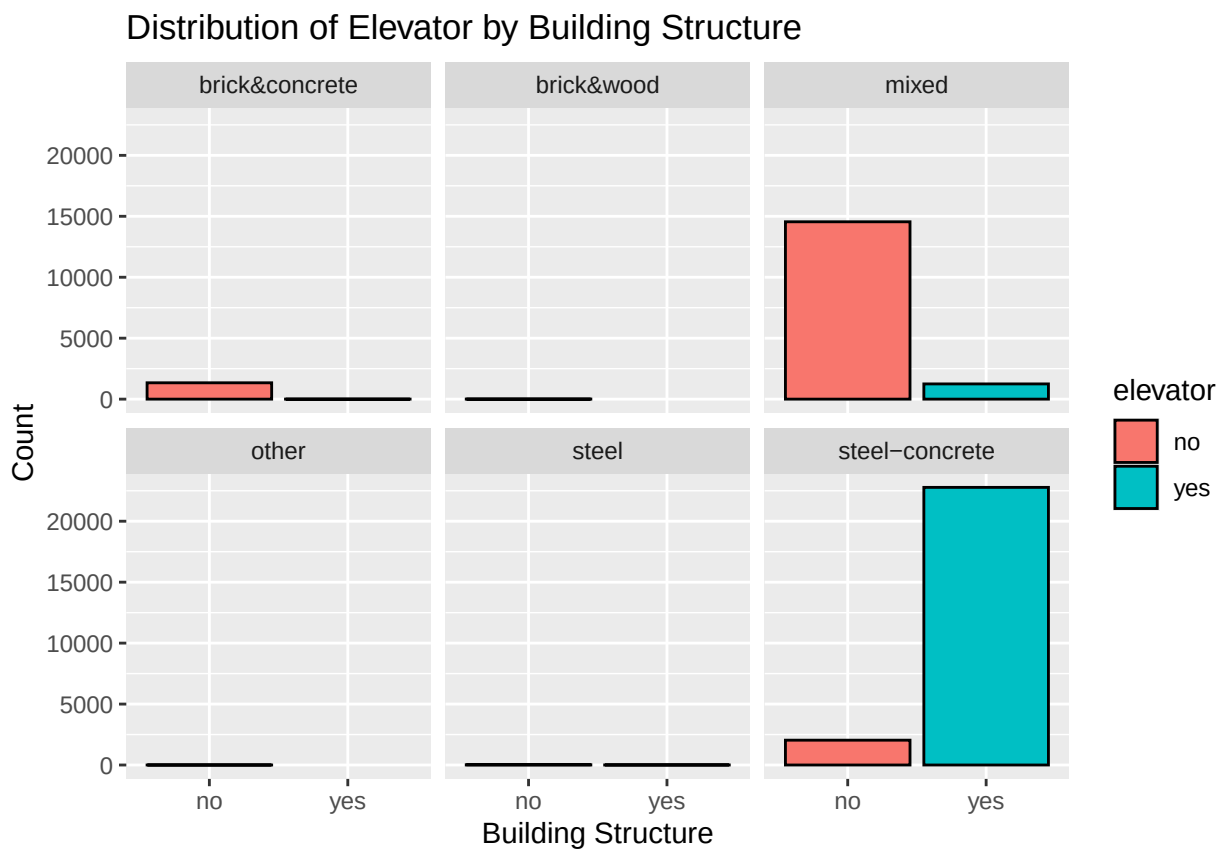
```
meanRenov <- data2017 %>%
  group_by(renovationCondition, district) %>%
  summarise(meanRenovPrice=mean(price))
ggplot(meanRenov, aes(x=district, y=meanRenovPrice, fill=renovationCondition))+
  geom_col(stat = "identity", position = position_dodge(), alpha = 0.75)+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5))
```



3.2.22 Elevator vs. Building structure

From the distribution of presence of elevator by building structure, we see that the greatest number of listings with the availability of an elevator are those that have a building structure of steel-concrete composite. For the greatest number of listings with the absence of an elevator, those would have a building structure of a mix of more than 3 distinguishable materials.

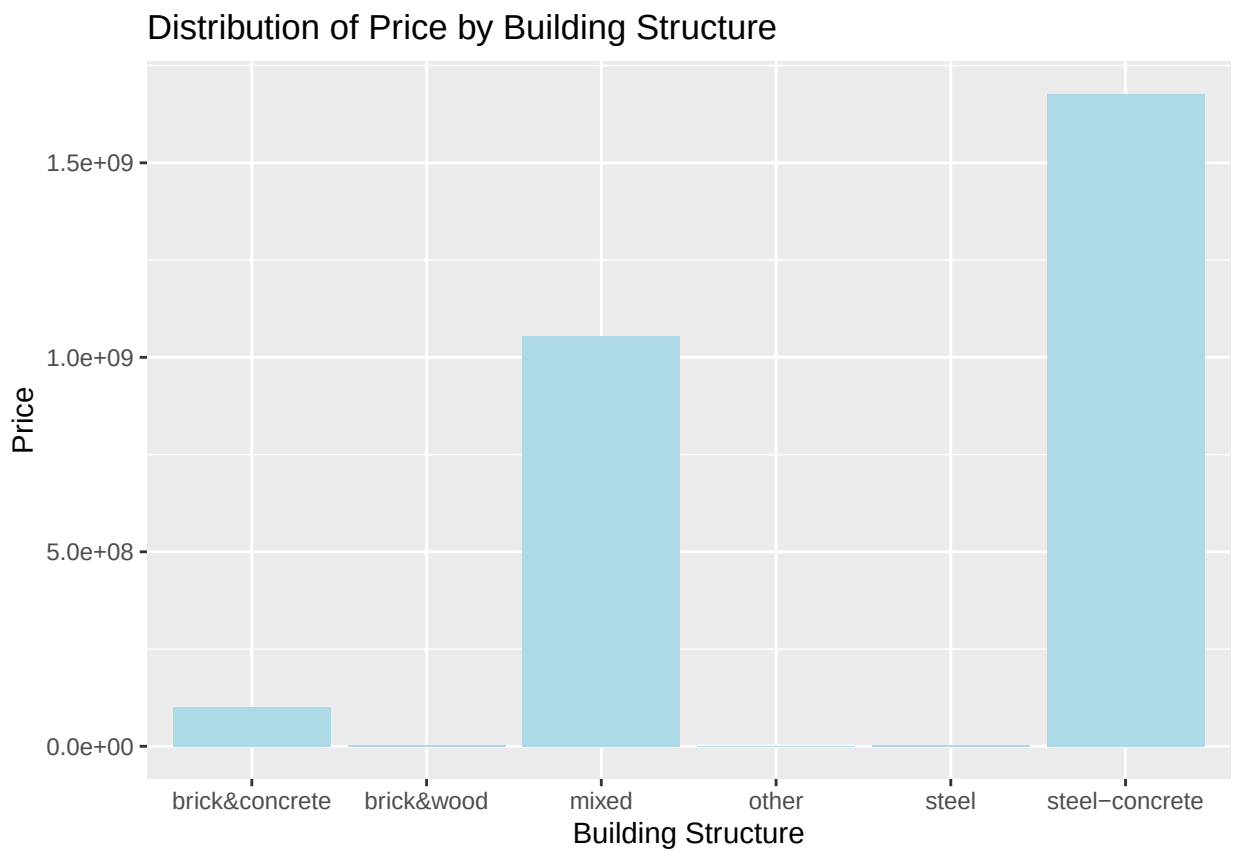
```
ggplot(data2017, aes(x = elevator, fill = elevator)) +
  geom_bar(colour = "black") +
  ggtitle("Distribution of Elevator by Building Structure") +
  xlab("Building Structure") + ylab("Count") +
  facet_wrap(~ buildingStructure)
```



3.2.23 Building structure vs Price

From the distribution of price by building structure, we can see that the most expensive listings are made of a mix of materials and the least expensive listings are made of other materials.

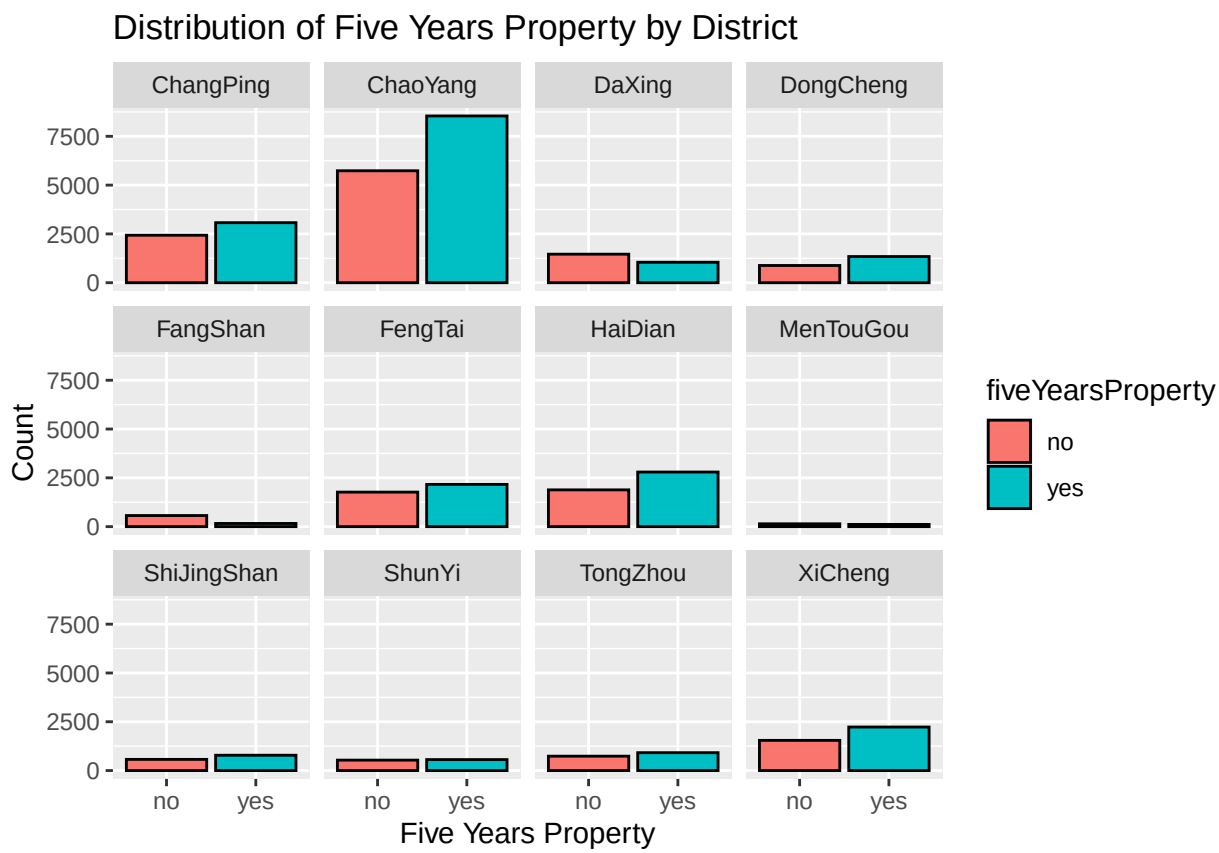
```
ggplot(data2017, aes(x = buildingStructure, y = price)) +
  geom_col(fill="lightblue") +
  labs(x = "Building Structure", y = "Price", title = "Distribution of Price by Building Structure")
```



3.2.24 Five years property vs District

From the distribution of five years property by district, we see that the district of ChaoYang has the highest number of previous owners who stayed in their property for more than 5 years followed by ChangPing. On the other hand, ChaoYang also has the highest number of previous owners who stayed in their property for less than 5 years followed by ChangPing. This is because ChaoYang is the largest district compared to the others and thus in terms of the total number of people staying more than or less than 5 years in that district will be the highest overall.

```
ggplot(data2017, aes(x = fiveYearsProperty, fill = fiveYearsProperty)) +
  geom_bar(colour = "black") +
  ggtitle("Distribution of Five Years Property by District") +
  xlab("Five Years Property") +ylab("Count") +
  facet_wrap(~ district)
```

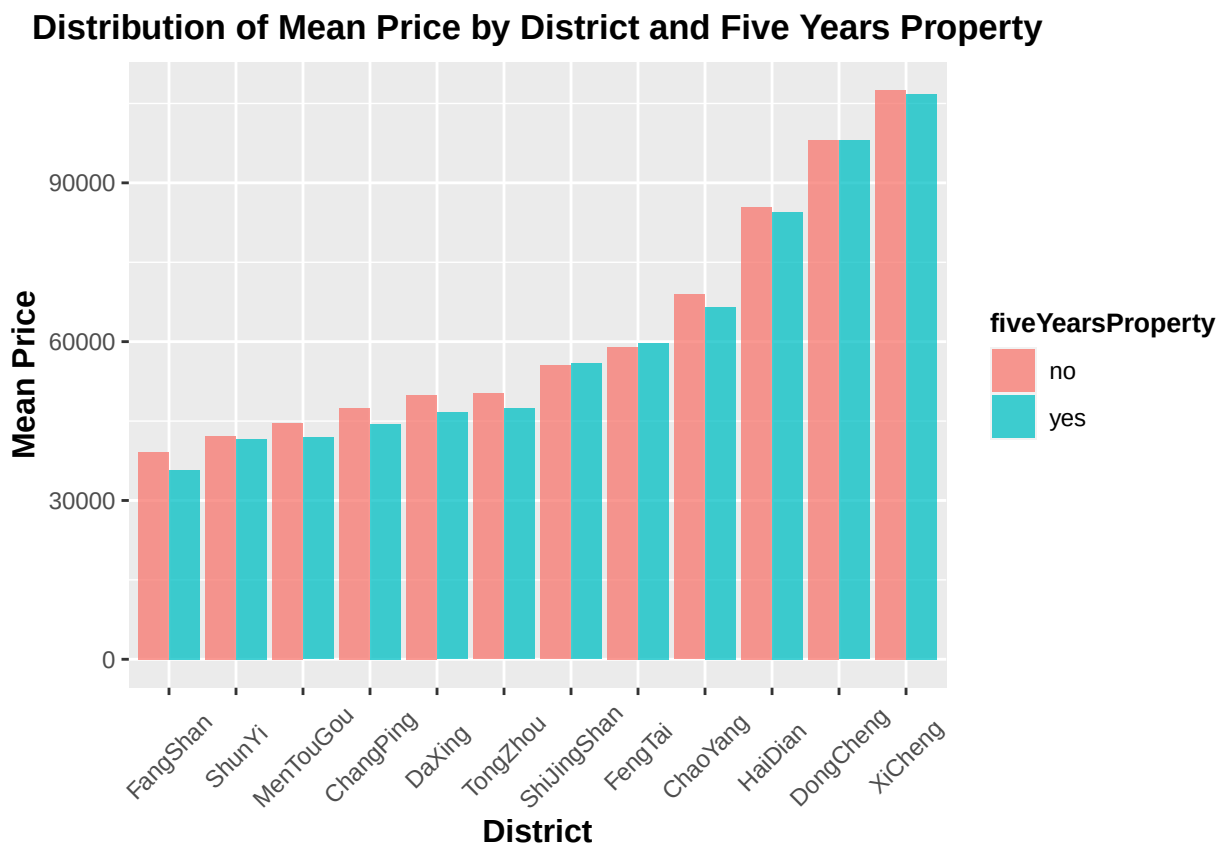


3.2.25 Five years property vs District vs Price

From the distribution, we see that most of the listings are more expensive when the previous owner has stayed in the listing for more than five years except for ShiJingShan, FengTai and DongCheng.

```
data2017 %>% group_by(district, fiveYearsProperty) %>%
  summarise(mean_price = mean(price)) %>%
  ggplot(aes(x = reorder(district, mean_price), y = mean_price, fill = fiveYearsProperty)) +
  geom_bar(stat = "identity", position = position_dodge(), alpha = 0.75) +
  ggtitle("Distribution of Mean Price by District and Five Years Property") +
  labs(x = "District", y = "Mean Price") +
  theme(plot.title = element_text(hjust = 0.5, face="bold"),
        axis.title.x = element_text(face="bold", size = 12),
        axis.title.y = element_text(face="bold", size = 12),
        legend.title = element_text(face="bold", size = 10)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5))
```

```
## `summarise()` has grouped output by 'district'. You can override using the
## `.groups` argument.
```



4 Model

4.1 Split data into train and test

```
set.seed(428)
index <- sample(nrow(data2017),21002,replace = FALSE)
train <- data2017[index,c(2,3,5:22)] # exclude total price
test <- data2017[-index,c(2,3,5:22)] # exclude total price
```

I splited the data in half, into train and test data. Train data is used to train the model, while test data is used to test the accuracy of prediction from the model.

4.2 Linear Model (Validation set approach)

```
linear_model <- lm(price~., data = train)
summary(linear_model)
```

```
##
## Call:
## lm(formula = price ~ ., data = train)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -91445  -7810   -757   6951  85651
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     313577.470  29876.260  10.496  < 2e-16 ***
## DOM                                 12.180      1.879   6.484 9.15e-11 ***
## followers                          -41.104      1.706 -24.097  < 2e-16 ***
## square                            -180.965      5.615 -32.228  < 2e-16 ***
## bedRoom                           1312.150    190.380   6.892 5.65e-12 ***
## livingRoom                        1682.340    248.925   6.758 1.43e-11 ***
## kitchen                          10355.825   1042.752   9.931  < 2e-16 ***
## bathRoom                          5760.562    325.707  17.686  < 2e-16 ***
## constructionTime                  -136.830     14.950  -9.153  < 2e-16 ***
## renovationConditionrough          -595.758    726.535  -0.820 0.412226
## renovationConditionsimplicity    -1153.744    315.831  -3.653 0.000260 ***
## renovationConditionhardcover      2158.937    305.053   7.077 1.52e-12 ***
## buildingStructurebrick&wood       8119.064   5452.328   1.489 0.136476
## buildingStructuremixed           -1083.454    544.279  -1.991 0.046536 *
## buildingStructureother            1941.979  13285.627   0.146 0.883787
## buildingStructuresteel            6963.625   3719.591   1.872 0.061200 .
## buildingStructuresteel-concrete     11.727    637.749   0.018 0.985330
## ladderRatio                      12759.482    644.702  19.791  < 2e-16 ***
## elevatoryes                       5500.379    388.325  14.164  < 2e-16 ***
## fiveYearsPropertyyes               -93.414    194.630  -0.480 0.631262
## subwaynear_subway                 5116.728    203.869  25.098  < 2e-16 ***
## districtChaoYang                 17749.138    329.479  53.870  < 2e-16 ***
## districtDaXing                     560.163    462.462   1.211 0.225808
## districtDongCheng                46669.165    502.792  92.820  < 2e-16 ***
## districtFangShan                 -9662.494    730.626 -13.225  < 2e-16 ***
## districtFengTai                  10588.066    409.529  25.854  < 2e-16 ***
## districtHaiDian                  35776.647    392.741  91.095  < 2e-16 ***
## districtMenTouGou                -2767.768   1143.145  -2.421 0.015478 *
## districtShiJingShan               7841.196    583.780  13.432  < 2e-16 ***
## districtShunYi                   -4952.583    624.703  -7.928 2.34e-15 ***
## districtTongZhou                  1417.899    526.028   2.695 0.007034 **
## districtXiCheng                  56775.642    432.884 131.157  < 2e-16 ***
## monthAugust                      -7862.109    569.096 -13.815  < 2e-16 ***
## monthDecember                   -10779.452    532.873 -20.229  < 2e-16 ***
## monthFebruary                    -5495.791    483.425 -11.368  < 2e-16 ***
## monthJanuary                     -8464.946    517.776 -16.349  < 2e-16 ***
## monthJuly                        -6658.913    577.549 -11.530  < 2e-16 ***
## monthJune                        -4883.629    629.409  -7.759 8.95e-15 ***
## monthMarch                       -1263.568    474.240  -2.664 0.007718 **
## monthMay                         -2316.099    653.369  -3.545 0.000394 ***
## monthNovember                   -10565.754    552.589 -19.120  < 2e-16 ***
## monthOctober                    -10086.283    564.698 -17.861  < 2e-16 ***
## monthSeptember                   -8555.170    544.126 -15.723  < 2e-16 ***
## day                                 15.671     11.617   1.349 0.177368
## totalFloor                        -115.558     19.272  -5.996 2.05e-09 ***
## floorRangelow                     -688.670    391.226  -1.760 0.078373 .
## floorRangemedium                   255.309    360.907   0.707 0.479320
```

```
## floorRangehigh                    -610.500    386.746  -1.579 0.114453
## floorRangetop                    -2353.843    427.997  -5.500 3.85e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13260 on 20953 degrees of freedom
## Multiple R-squared:  0.7014, Adjusted R-squared:  0.7008
## F-statistic:  1026 on 48 and 20953 DF,  p-value: < 2.2e-16
```

In this model, I randomly split data into training set and testing set, which is the validation set approach. Based on the results of linear model, all the variables are important except for five year property, the day in a month being sold, and building structure.

Coefficient interpretation:

- if days on market increase by 1, the average price increase by 12.18
- if followers increase by 1, the average price decreases by 41.1
- if square increases by 1, the average price decreases by 180
- if number of bedroom increase by 1, the average price increases by 1312
- if number of living room increase by 1, the average price increases by 1682
- if number of bkitchen increase by 1, the average price increases by 10356
- if number of bathroom increase by 1, the average price increases by 5761
- if construction year increase by 1, the average price decreases by 136.8
- if ladder ratio increase by 1, the average price increases by 12759
- if total number of floors increase by 1, the average price decreases by 116
- if the house is sold in April, we expect the price to be higher than houses sold in other months.
- if there is a subway near the house, we would expect the price to be 5116 Yuan higher than houses without a subway.
- if there is a elevator near the house, we would expect the price to be 5500 Yuan higher than houses without a elevator
- Being in ChaoYang, DongCheng, FengTai, HaiDian, ShiJingShan, TongZhou, and XiCheng increases the price with a significant amount compared to ChangPing because these are all core districts that are located in the center of Beijing.
- Being in FangShan, MenTouGou and ShunYi significantly decreases the price compared to ChangPing because these are the rural districts.
- The price in DaXing is not significantly different from price in ChangPing since these two are both rural districts.
- If the apartment is at the top, we would expect the price to be 2354 Yuan lower than houses at the bottom.

```r
# training MSE
pred_train_lm <- predict(linear_model, train, type="response")
mean((pred_train_lm-train$price)^2)
```

```
## [1] 175467451
```

```r
# testing MSE
pred_test_lm <- predict(linear_model, test, type="response")
mean((pred_test_lm-test$price)^2)
```

```
## [1] 181399456
```

After putting test data into linear model, the mean squared error for both test data and train data look pretty big. Let's look at if k-fold cross validation could produce a better MSE.

4.3 Linear Model (k-fold cross validation, k=10)

```
library(caret)
set.seed(428)
train.kfold <- trainControl(method = "cv", number = 10)
# Train the model
kfold_model <- train(price ~., data = data2017[,c(2,3,5:22)], method = "lm",
               trControl = train.kfold)
(kfold_model$resample$RMSE)^2
```

```
##  [1] 183887111 177543318 178967560 173453567 171172936 172895095 178314468
##  [8] 182713533 187293992 180828934
```

The above list of number shows the MSE for each fold. Comparing the MSE of k-fold cross validation with MSE of validation set approach, performing k-fold cross validation did not improve much on MSE.

4.4 Backward Regression

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.0.2
```

```
backward <- regsubsets(price~.,data=train,method="backward")
summary(backward)
```

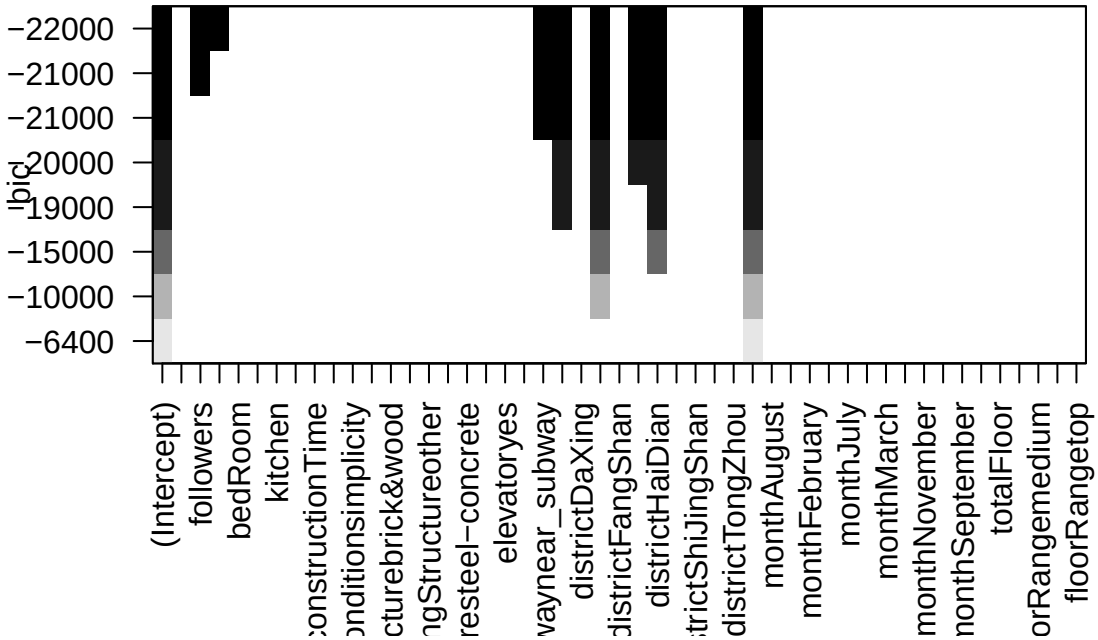
```
## Subset selection object
## Call: regsubsets.formula(price ~ ., data = train, method = "backward")
## 48 Variables  (and intercept)
##                                 Forced in Forced out
## DOM                                 FALSE      FALSE
## followers                           FALSE      FALSE
## square                              FALSE      FALSE
## bedRoom                             FALSE      FALSE
## livingRoom                          FALSE      FALSE
## kitchen                             FALSE      FALSE
## bathRoom                            FALSE      FALSE
## constructionTime                    FALSE      FALSE
## renovationConditionrough            FALSE      FALSE
## renovationConditionsimplicity       FALSE      FALSE
## renovationConditionhardcover        FALSE      FALSE
## buildingStructurebrick&wood         FALSE      FALSE
## buildingStructuremixed              FALSE      FALSE
## buildingStructureother              FALSE      FALSE
## buildingStructuresteel              FALSE      FALSE
## buildingStructuresteel-concrete     FALSE      FALSE
## ladderRatio                         FALSE      FALSE
## elevatoryes                         FALSE      FALSE
## fiveYearsPropertyyes                FALSE      FALSE
## subwaynear_subway                   FALSE      FALSE
```

```
## districtChaoYang                    FALSE      FALSE
## districtDaXing                      FALSE      FALSE
## districtDongCheng                   FALSE      FALSE
## districtFangShan                    FALSE      FALSE
## districtFengTai                     FALSE      FALSE
## districtHaiDian                     FALSE      FALSE
## districtMenTouGou                   FALSE      FALSE
## districtShiJingShan                 FALSE      FALSE
## districtShunYi                      FALSE      FALSE
## districtTongZhou                    FALSE      FALSE
## districtXiCheng                     FALSE      FALSE
## monthAugust                         FALSE      FALSE
## monthDecember                       FALSE      FALSE
## monthFebruary                       FALSE      FALSE
## monthJanuary                        FALSE      FALSE
## monthJuly                           FALSE      FALSE
## monthJune                           FALSE      FALSE
## monthMarch                          FALSE      FALSE
## monthMay                            FALSE      FALSE
## monthNovember                       FALSE      FALSE
## monthOctober                        FALSE      FALSE
## monthSeptember                      FALSE      FALSE
## day                                 FALSE      FALSE
## totalFloor                          FALSE      FALSE
## floorRangelow                       FALSE      FALSE
## floorRangemedium                    FALSE      FALSE
## floorRangehigh                      FALSE      FALSE
## floorRangetop                       FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: backward
##          DOM followers square bedRoom livingRoom kitchen bathRoom
## 1  ( 1 ) " " " "       " "    " "     " "        " "     " "
## 2  ( 1 ) " " " "       " "    " "     " "        " "     " "
## 3  ( 1 ) " " " "       " "    " "     " "        " "     " "
## 4  ( 1 ) " " " "       " "    " "     " "        " "     " "
## 5  ( 1 ) " " " "       " "    " "     " "        " "     " "
## 6  ( 1 ) " " " "       " "    " "     " "        " "     " "
## 7  ( 1 ) " " "*"       " "    " "     " "        " "     " "
## 8  ( 1 ) " " "*"       "*"    " "     " "        " "     " "
##          constructionTime renovationConditionrough
## 1  ( 1 ) " "              " "
## 2  ( 1 ) " "              " "
## 3  ( 1 ) " "              " "
## 4  ( 1 ) " "              " "
## 5  ( 1 ) " "              " "
## 6  ( 1 ) " "              " "
## 7  ( 1 ) " "              " "
## 8  ( 1 ) " "              " "
##          renovationConditionsimplicity renovationConditionhardcover
## 1  ( 1 ) " "                           " "
## 2  ( 1 ) " "                           " "
## 3  ( 1 ) " "                           " "
## 4  ( 1 ) " "                           " "
## 5  ( 1 ) " "                           " "
```

```
## 6  ( 1 ) " "                           " "
## 7  ( 1 ) " "                           " "
## 8  ( 1 ) " "                           " "
##          buildingStructurebrick&wood buildingStructuremixed
## 1  ( 1 ) " "                         " "
## 2  ( 1 ) " "                         " "
## 3  ( 1 ) " "                         " "
## 4  ( 1 ) " "                         " "
## 5  ( 1 ) " "                         " "
## 6  ( 1 ) " "                         " "
## 7  ( 1 ) " "                         " "
## 8  ( 1 ) " "                         " "
##          buildingStructureother buildingStructuresteel
## 1  ( 1 ) " "                    " "
## 2  ( 1 ) " "                    " "
## 3  ( 1 ) " "                    " "
## 4  ( 1 ) " "                    " "
## 5  ( 1 ) " "                    " "
## 6  ( 1 ) " "                    " "
## 7  ( 1 ) " "                    " "
## 8  ( 1 ) " "                    " "
##          buildingStructuresteel-concrete ladderRatio elevatoryes
## 1  ( 1 ) " "                             " "         " "
## 2  ( 1 ) " "                             " "         " "
## 3  ( 1 ) " "                             " "         " "
## 4  ( 1 ) " "                             " "         " "
## 5  ( 1 ) " "                             " "         " "
## 6  ( 1 ) " "                             " "         " "
## 7  ( 1 ) " "                             " "         " "
## 8  ( 1 ) " "                             " "         " "
##          fiveYearsPropertyyes subwaynear_subway districtChaoYang districtDaXing
## 1  ( 1 ) " "                  " "               " "              " "
## 2  ( 1 ) " "                  " "               " "              " "
## 3  ( 1 ) " "                  " "               " "              " "
## 4  ( 1 ) " "                  " "               "*"              " "
## 5  ( 1 ) " "                  " "               "*"              " "
## 6  ( 1 ) " "                  "*"               "*"              " "
## 7  ( 1 ) " "                  "*"               "*"              " "
## 8  ( 1 ) " "                  "*"               "*"              " "
##          districtDongCheng districtFangShan districtFengTai districtHaiDian
## 1  ( 1 ) " "               " "              " "             " "
## 2  ( 1 ) "*"               " "              " "             " "
## 3  ( 1 ) "*"               " "              " "             "*"
## 4  ( 1 ) "*"               " "              " "             "*"
## 5  ( 1 ) "*"               " "              "*"             "*"
## 6  ( 1 ) "*"               " "              "*"             "*"
## 7  ( 1 ) "*"               " "              "*"             "*"
## 8  ( 1 ) "*"               " "              "*"             "*"
##          districtMenTouGou districtShiJingShan districtShunYi districtTongZhou
## 1  ( 1 ) " "               " "                 " "            " "
## 2  ( 1 ) " "               " "                 " "            " "
## 3  ( 1 ) " "               " "                 " "            " "
## 4  ( 1 ) " "               " "                 " "            " "
## 5  ( 1 ) " "               " "                 " "            " "
```

```
## 6  ( 1 ) " "                " "                  " "             " "
## 7  ( 1 ) " "                " "                  " "             " "
## 8  ( 1 ) " "                " "                  " "             " "
##          districtXiCheng monthAugust monthDecember monthFebruary monthJanuary
## 1  ( 1 ) "*"             " "         " "           " "           " "
## 2  ( 1 ) "*"             " "         " "           " "           " "
## 3  ( 1 ) "*"             " "         " "           " "           " "
## 4  ( 1 ) "*"             " "         " "           " "           " "
## 5  ( 1 ) "*"             " "         " "           " "           " "
## 6  ( 1 ) "*"             " "         " "           " "           " "
## 7  ( 1 ) "*"             " "         " "           " "           " "
## 8  ( 1 ) "*"             " "         " "           " "           " "
##          monthJuly monthJune monthMarch monthMay monthNovember monthOctober
## 1  ( 1 ) " "       " "       " "        " "      " "           " "
## 2  ( 1 ) " "       " "       " "        " "      " "           " "
## 3  ( 1 ) " "       " "       " "        " "      " "           " "
## 4  ( 1 ) " "       " "       " "        " "      " "           " "
## 5  ( 1 ) " "       " "       " "        " "      " "           " "
## 6  ( 1 ) " "       " "       " "        " "      " "           " "
## 7  ( 1 ) " "       " "       " "        " "      " "           " "
## 8  ( 1 ) " "       " "       " "        " "      " "           " "
##          monthSeptember day totalFloor floorRangelow floorRangemedium
## 1  ( 1 ) " "            " " " "        " "           " "
## 2  ( 1 ) " "            " " " "        " "           " "
## 3  ( 1 ) " "            " " " "        " "           " "
## 4  ( 1 ) " "            " " " "        " "           " "
## 5  ( 1 ) " "            " " " "        " "           " "
## 6  ( 1 ) " "            " " " "        " "           " "
## 7  ( 1 ) " "            " " " "        " "           " "
## 8  ( 1 ) " "            " " " "        " "           " "
##          floorRangehigh floorRangetop
## 1  ( 1 ) " "            " "
## 2  ( 1 ) " "            " "
## 3  ( 1 ) " "            " "
## 4  ( 1 ) " "            " "
## 5  ( 1 ) " "            " "
## 6  ( 1 ) " "            " "
## 7  ( 1 ) " "            " "
## 8  ( 1 ) " "            " "
```

```r
plot(backward,scale="bic", cex.names= 0.5)
```



Based on the plot and model summary, the best backward regression model includes followers, square, subway, district ChaoYang, district DongCheng, district FengTai, district HaiDian, district Xicheng (all important districts are the core districts). Lets put these important variables in linear model.

```
backward_model <- lm(price~followers+square+subway+district, data = train)
summary(backward_model)
```

```
##
## Call:
## lm(formula = price ~ followers + square + subway + district,
##     data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -86417  -8460   -954   7496  78891
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         51471.291    401.967 128.049  < 2e-16 ***
## followers             -42.645      1.584 -26.919  < 2e-16 ***
## square                -59.658      2.861 -20.850  < 2e-16 ***
## subwaynear_subway    5378.838    213.642  25.177  < 2e-16 ***
## districtChaoYang    19500.838    326.854  59.662  < 2e-16 ***
## districtDaXing       1756.591    487.842   3.601 0.000318 ***
## districtDongCheng   48363.965    522.106  92.633  < 2e-16 ***
## districtFangShan    -8282.352    768.129 -10.783  < 2e-16 ***
```

```
## districtFengTai     11245.388    421.803  26.660  < 2e-16 ***
## districtHaiDian     37057.794    403.827  91.767  < 2e-16 ***
## districtMenTouGou   -2471.013   1220.960  -2.024 0.043001 *
## districtShiJingShan  9277.349    606.502  15.296  < 2e-16 ***
## districtShunYi      -4447.041    665.097  -6.686 2.35e-11 ***
## districtTongZhou     2227.253    557.709   3.994 6.53e-05 ***
## districtXiCheng     57356.940    437.214 131.187  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14190 on 20987 degrees of freedom
## Multiple R-squared:  0.6574, Adjusted R-squared:  0.6572
## F-statistic:  2877 on 14 and 20987 DF,  p-value: < 2.2e-16
```

```r
# training MSE
pred_train_lm <- predict(backward_model, train, type="response")
mean((pred_train_lm-train$price)^2)
```

```
## [1] 201322711
```

```r
# testing MSE
pred_test_lm <- predict(backward_model, test, type="response")
mean((pred_test_lm-test$price)^2)
```

```
## [1] 206735849
```

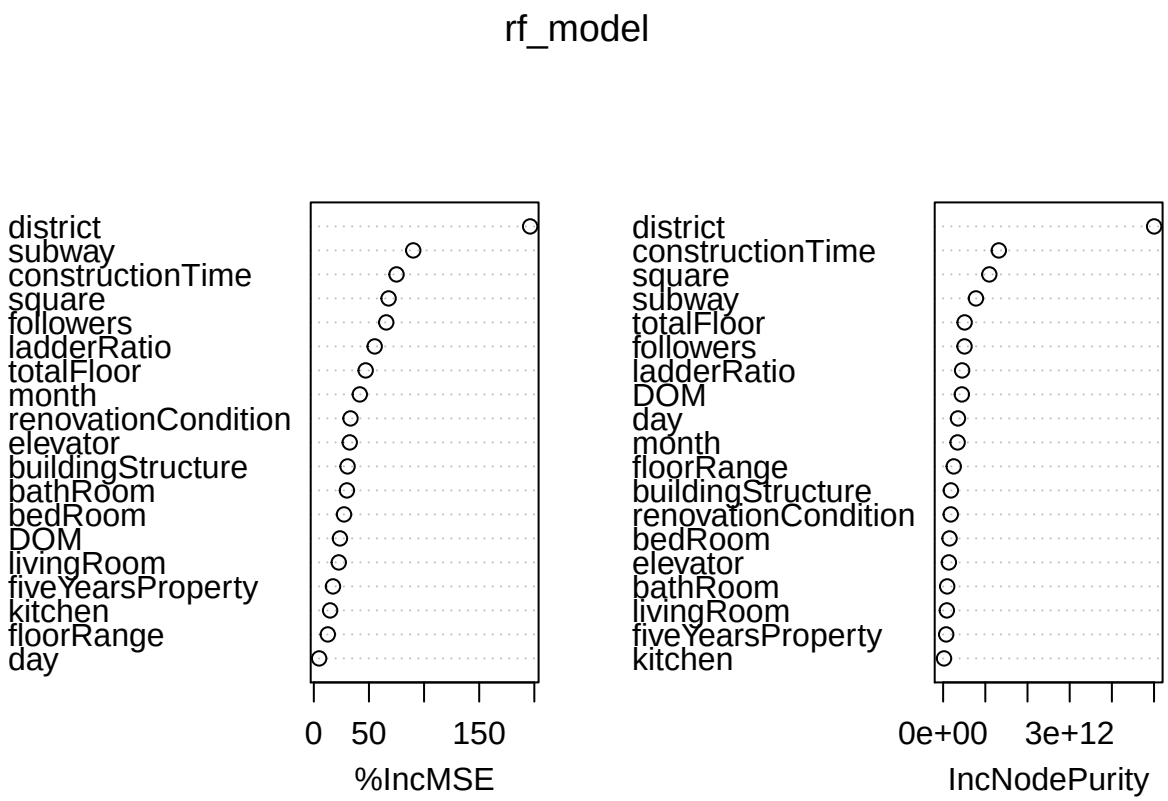
The resulting MSE from backward regression is larger than the MSE from simple linear regression that uses all the predictors, so we prefer the one with all predictors.

4.5 Random Forest

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.2

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

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

```r
set.seed(428)
rf_model <- randomForest(price~.,data=train,mtry=6,importance=TRUE)
varImpPlot(rf_model)
```



Since we are using MSE to evaluate the models, we will look at the plot on the left. The plot shows that district contributes most in reducing MSE in the model, with subway, construction time, square, followers, etc in decreasing order.

```r
# training MSE
pred_train_rf <- predict(rf_model, train, type="response")
mean((pred_train_rf-train$price)^2)
```

```
## [1] 29270712
```

```r
# testing MSE
pred_test_rf <- predict(rf_model, test, type="response")
mean((pred_test_rf-test$price)^2)
```

```
## [1] 148751244
```

Random Forest significantly reduced the MSE for both prediction of training and testing dataset. The MSE for testing dataset is much larger than training dataset because of overfitting effect, but it's still better than linear models. Because we have a smaller MSE, random forest is so far the best model in predicting house price per square meter, with district, subway, contruction time, square as some of the most important predictor.

4.6 Fitting Model For Core Districts

```
core_district <- data2017 %>%
  filter(district %in% c("XiCheng" , "DongCheng" , "ChaoYang" , "HaiDian" , "ShiJingShan" , "FengTai"))

set.seed(428)
index <- sample(nrow(core_district),15123,replace = FALSE)
train <- core_district[index,c(2,3,5:22)] # exclude total price
test <- core_district[-index,c(2,3,5:22)] # exclude total price

# training MSE for linear model
pred_train_lm <- predict(linear_model, train, type="response")
mean((pred_train_lm-train$price)^2)
```

```
## [1] 221628490
```

```
# testing MSE for linear model
pred_test_lm <- predict(linear_model, test, type="response")
mean((pred_test_lm-test$price)^2)
```

```
## [1] 219024786
```

```
# training MSE for random forest
pred_train_rf <- predict(rf_model, train, type="response")
mean((pred_train_rf-train$price)^2)
```

```
## [1] 560138188
```

```
# testing MSE for random forest
pred_test_rf <- predict(rf_model, test, type="response")
mean((pred_test_rf-test$price)^2)
```

```
## [1] 548322832
```

Only fitting core districts in the linear model and random forest model does not improve MSE, and even generates larger MSE.

4.7 Fitting Model for Rural Districts

```
suburb <- data2017 %>%
  filter(district %in% c("DaXing", "ChangPing", "FangShan", "MenTouGou", "ShunYi", "TongZhou"))

set.seed(428)
index <- sample(nrow(suburb),5878,replace = FALSE)
train <- suburb[index,c(2,3,5:22)] # exclude total price
test <- suburb[-index,c(2,3,5:22)] # exclude total price

# training MSE for linear model
pred_train_lm <- predict(linear_model, train, type="response")
mean((pred_train_lm-train$price)^2)
```

```
## [1] 72136058
```

```r
# testing MSE for linear model
pred_test_lm <- predict(linear_model, test, type="response")
mean((pred_test_lm-test$price)^2)
```

```
## [1] 69144011
```

```r
# training MSE for random forest
pred_train_rf <- predict(rf_model, train, type="response")
mean((pred_train_rf-train$price)^2)
```

```
## [1] 137567641
```

```r
# testing MSE for random forest
pred_test_rf <- predict(rf_model, test, type="response")
mean((pred_test_rf-test$price)^2)
```

```
## [1] 141215903
```

Fitting rural districts in the linear model significantly improves MSE, but not for random forest. Therefore, the linear model better predicts house prices in rural districts, while random forest better predicts house prices in all districts together.

5 Conclusion

We found that people like to trade houses in the beginning of the year for 2017. Apartments with 2 bedrooms, 1 living room, 1 kitchen, 1 bathroom, and well renovated with elevator and subway near by was the most popular type of housing for transaction.
The most important predictors for price based on the Random Forest model and Backward Regression model are district, subway, square, and construction time, which is consistent with the relationships we investigated in section 3.
As we predicted, the core districts generally have higher price per square meter compared to rural districts. There is no denying that district is the most essential factor that affects the price. Thus, we need to consider districts while comparing other variables. Among those, ChaoYang is "special". It is a core district but has great amounts of transactions and also significant high amounts of houses with big square meters. The total prices have a wide range as well. That is, you could find almost every kind of house in ChaoYang.
The availability of a subway near by is associated with higher average price. The price also decreases as square meters goes up. As for construction time, the later the property was constructed, the lower the average price is. We were expecting "sold fast" and "have many followers" to indicate the popularity of the property. But the fact turns out to be there are actually many houses sold within a very short period of time that have no followers and there is actually not much trend observed between these 2 variables.
Despite *hardcover* representing the best renovation condition, *other* represents the worst renovation condition yet people are willing to pay a similar price to *hardcover*. Listings that have been previously occupied for more than five years are associated with higher prices.

Work Cited

some online resources that we used 1. It mentions the price drop on beijing houses in 2017 because of some regulations http://www.xinhuanet.com/english/2018-01/29/c_136933533.htm