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 foot 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. This means that any property that was listed or sold in 2017 will be included and the others will be excluded. 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 318,815 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.

Beijing Districts: There are a total of 13 districts within Beijing that we will be focusing on. These 13 districts include DongCheng, FengTai, YiZhuang, DaXing, FangShan, ChangPing, ChaoYang, HaiDian, ShiJingShan, XiCheng, TongZhou, MenTouGou and ShunYi. Each listing in our data set lies in one of these 13 districts.

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.
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 number of residents who live on the same floor to the number of elevators or ladders that particular floor has. It relates to the frequency of usage of the elevator per floor. In this case, 1.00 refers to the highest frequency of usage and 0.00 refers to lowest frequency of usage.
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.
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.

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, 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.

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.

Districts: 1-DongCheng 2-FengTai 3-YiZhuang 4-DaXing 5-FangShan 6-ChangPing 7-ChaoYang 8-HaiDian 9-ShiJingShan 10-XiCheng 11-TongZhou 12-MenTouGou 13-ShunYi

Section 2: Data Cleaning

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

```
# 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.

```
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.

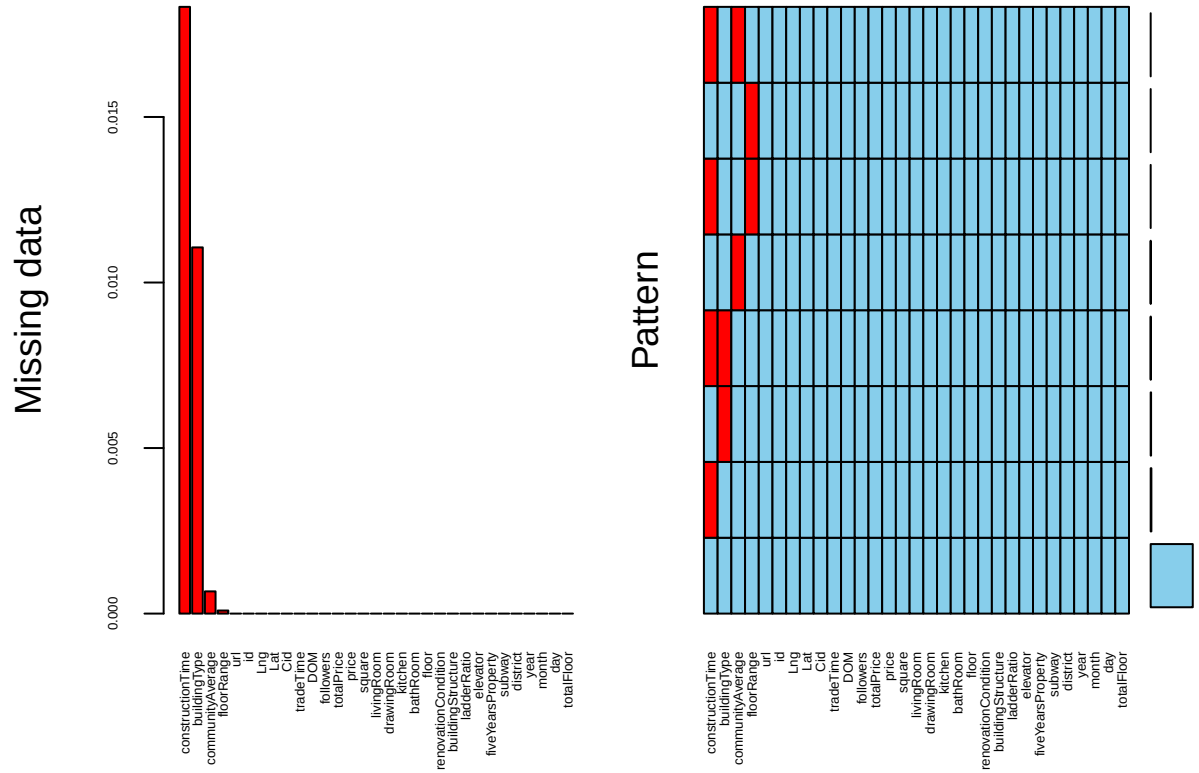
2.3 Recoding District Variable

2.3 Missing Values

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

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

```
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
##           livingRoom 0.000000e+00
##          drawingRoom 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.

2.4 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))
```

Section 3: Exploratory Analysis

3.1 Distribution of variables and outliers

3.1.1 Price

```
summary(data2017$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     136   49332   62670   67430   81078  150000
```

```
data2017 <- data2017 %>%
  filter(price>1000)
```

3.1.2 Square

A house with more 1745.5 square meter seems very unreasonable, so we decided houses that are less than 1000 square meters are plausible.

```
data2017 <- data2017 %>%
  filter(square < 1000)
```

3.1.3 Kitchen

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

```
##             id DOM followers totalPrice price square livingRoom drawingRoom
## 1 101101782929  69        78        735 26750 274.77          4           2
##   kitchen bathRoom constructionTime renovationCondition buildingStructure
## 1       3        2             2001                   4                 6
##   ladderRatio elevator fiveYearsProperty subway district year     month day
## 1         0.5        1                 1      1        6 2017 September  13
##   totalFloor floorRange
## 1         13       high
```

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

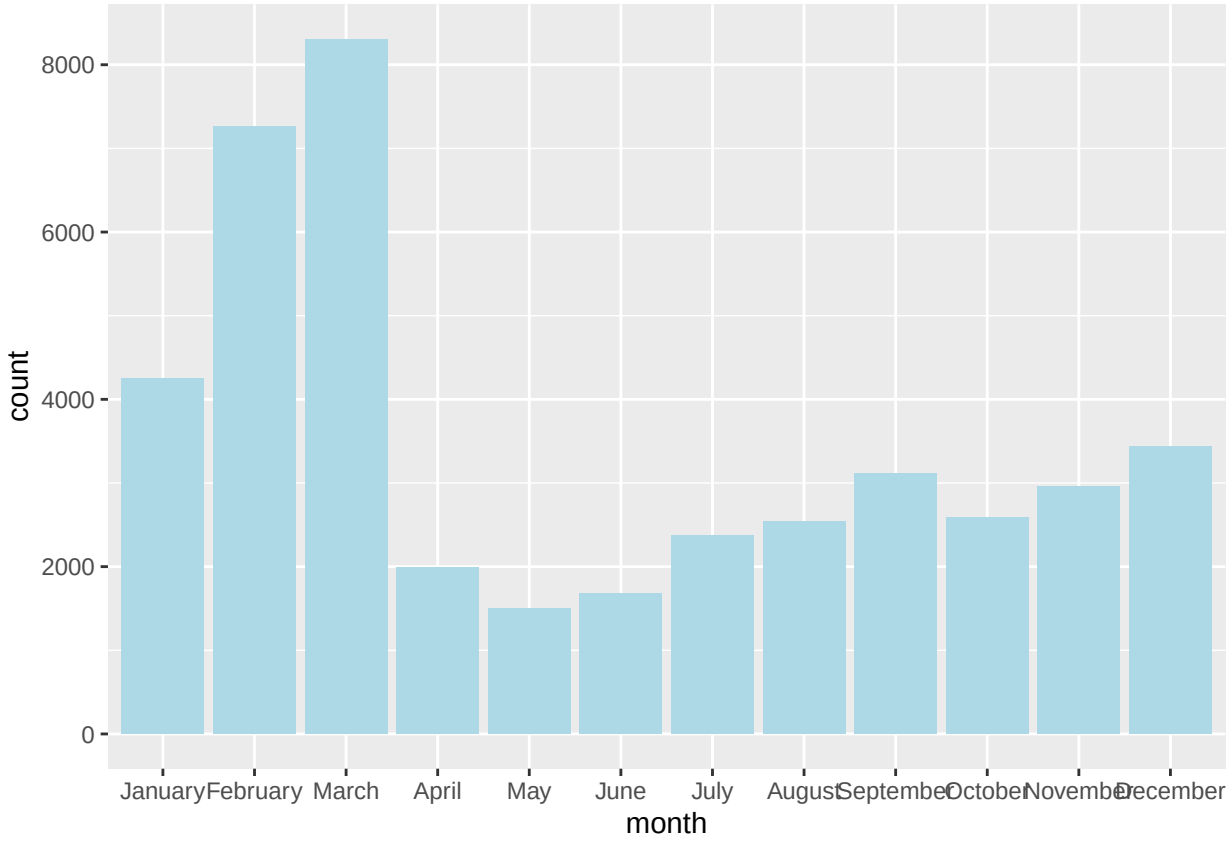
```
data2017 <- data2017 %>%
  filter(kitchen < 3)
```

Ladder Ratio

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

Month

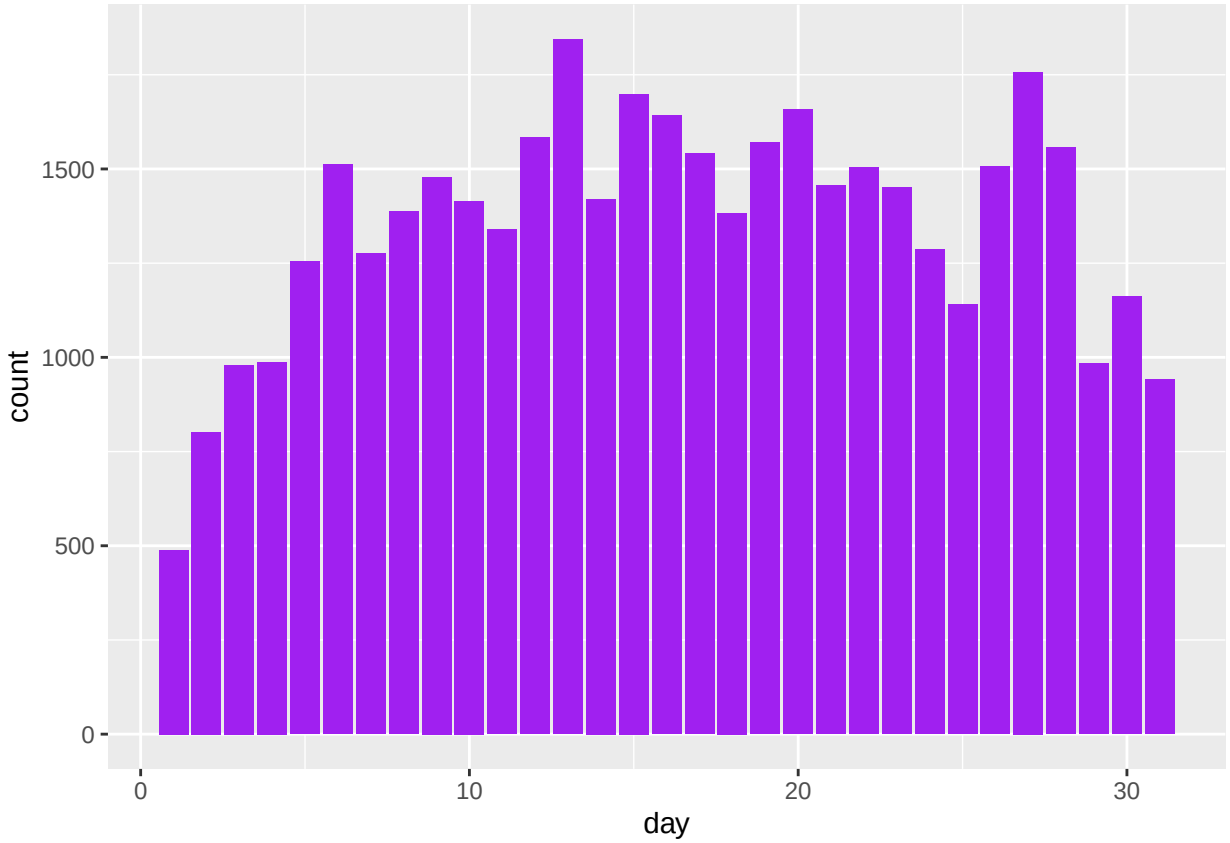
```
ggplot(data2017, aes(x = month)) +
  geom_bar(fill = "light blue") +
  scale_x_discrete(limits = month.name)
```



The first three months of year 2017 had the most trading of houses, with March being the highest number of trading.

Day

```
ggplot(data2017, aes(x = day)) +
  geom_bar(fill = "purple")
```

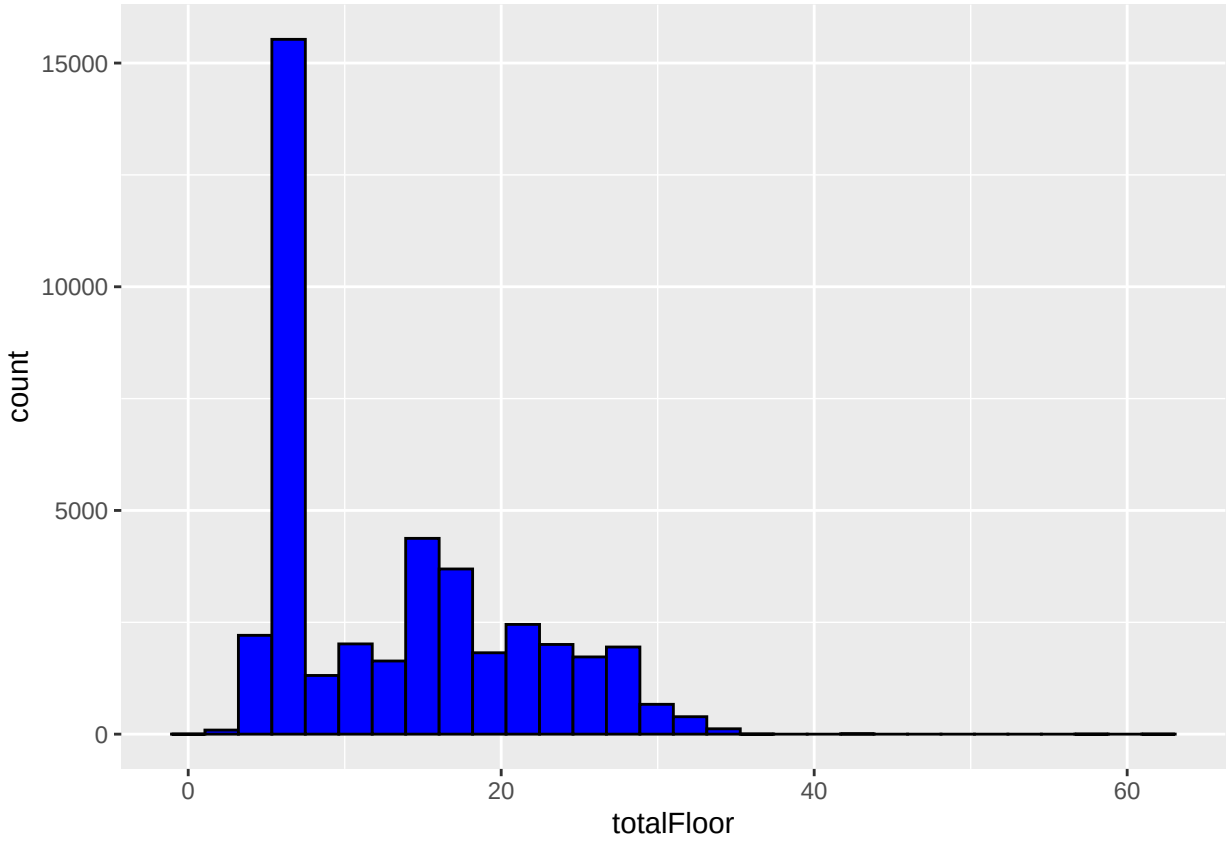


There isn't a unique day of the month that people liked to buy houses, but people relatively like to buy houses in the middle of the month. Few trades happened in the beginning of the month.

Total Floor

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

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




```
summary(data2017$totalFloor)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    6.00   11.00   13.38   20.00   63.00
```

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

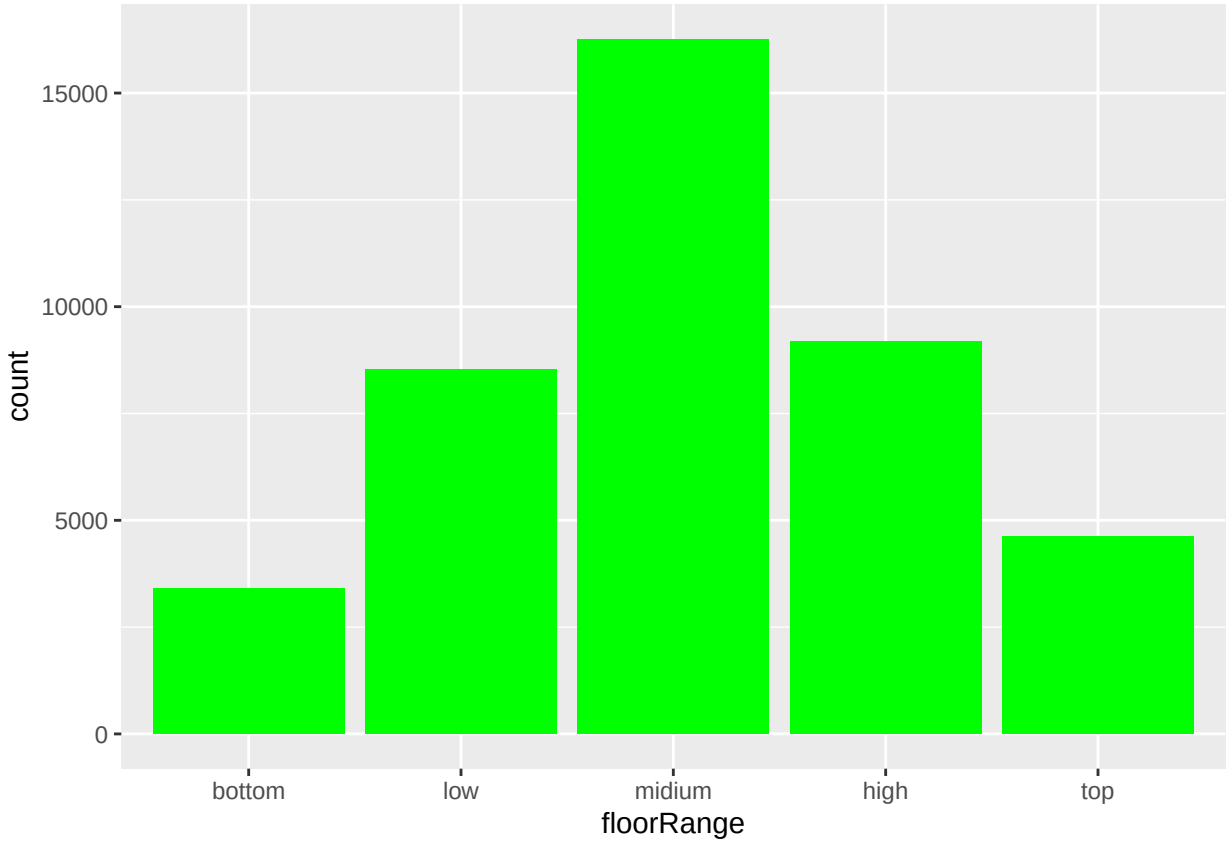
```
##             id totalFloor floorRange
## 1 101101127418         63        low
## 2 101101690389         57     midium
## 3 101091696453         42        low
## 4 101100641538         42        low
## 5 101100768834         42       high
## 6 101100791635         42     midium
```

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

The distribution of total floor is skewed to the right. There are quite a few very tall buildings, but such 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 trade with total floor of 63.

Floor Range

```
ggplot(data2017, aes(x = floorRange)) +
  geom_bar(fill = "green") +
  scale_x_discrete(limits = c("bottom", "low", "midium", "high", "top"))
```



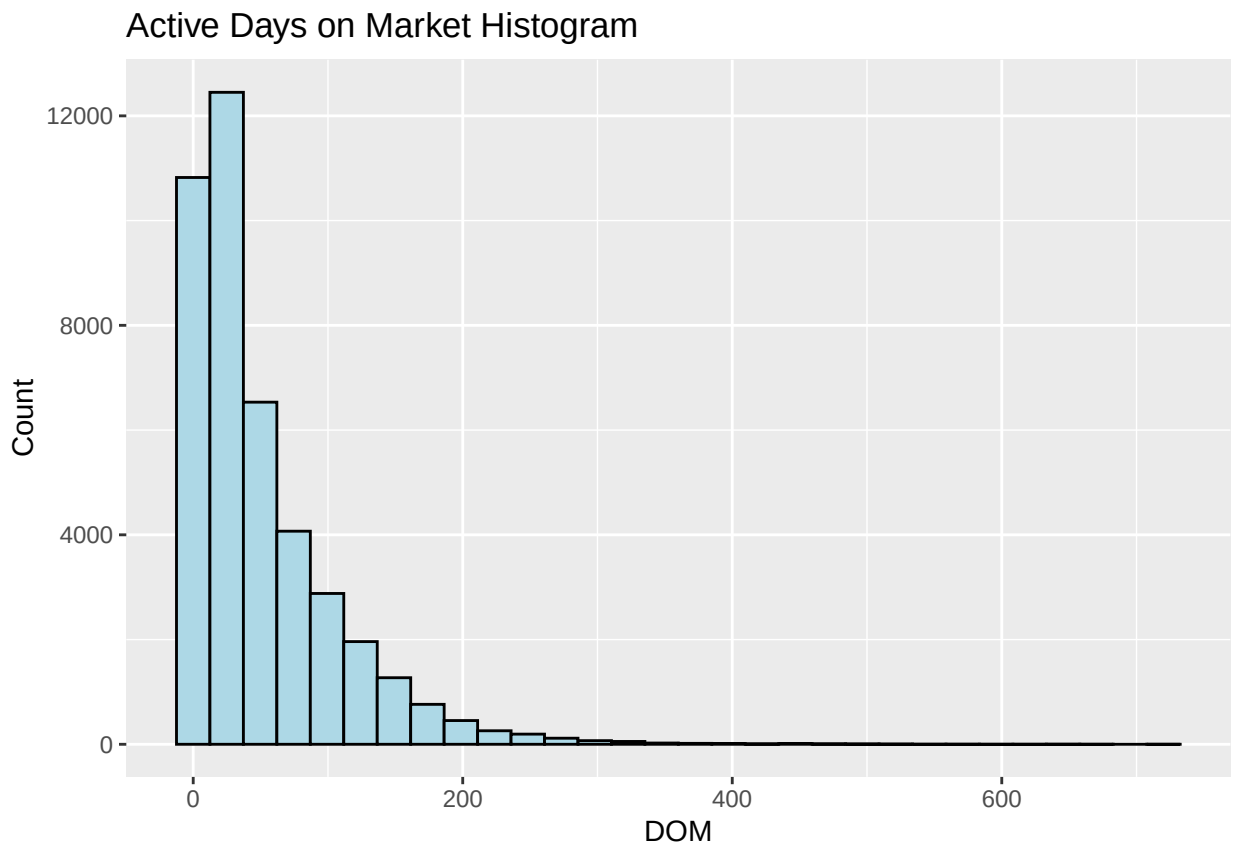
Apartments located in the middle of a building were traded the most in 2017.

DOM

The graph of number of days on market is heavily skewed to the right.

```
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`.
```

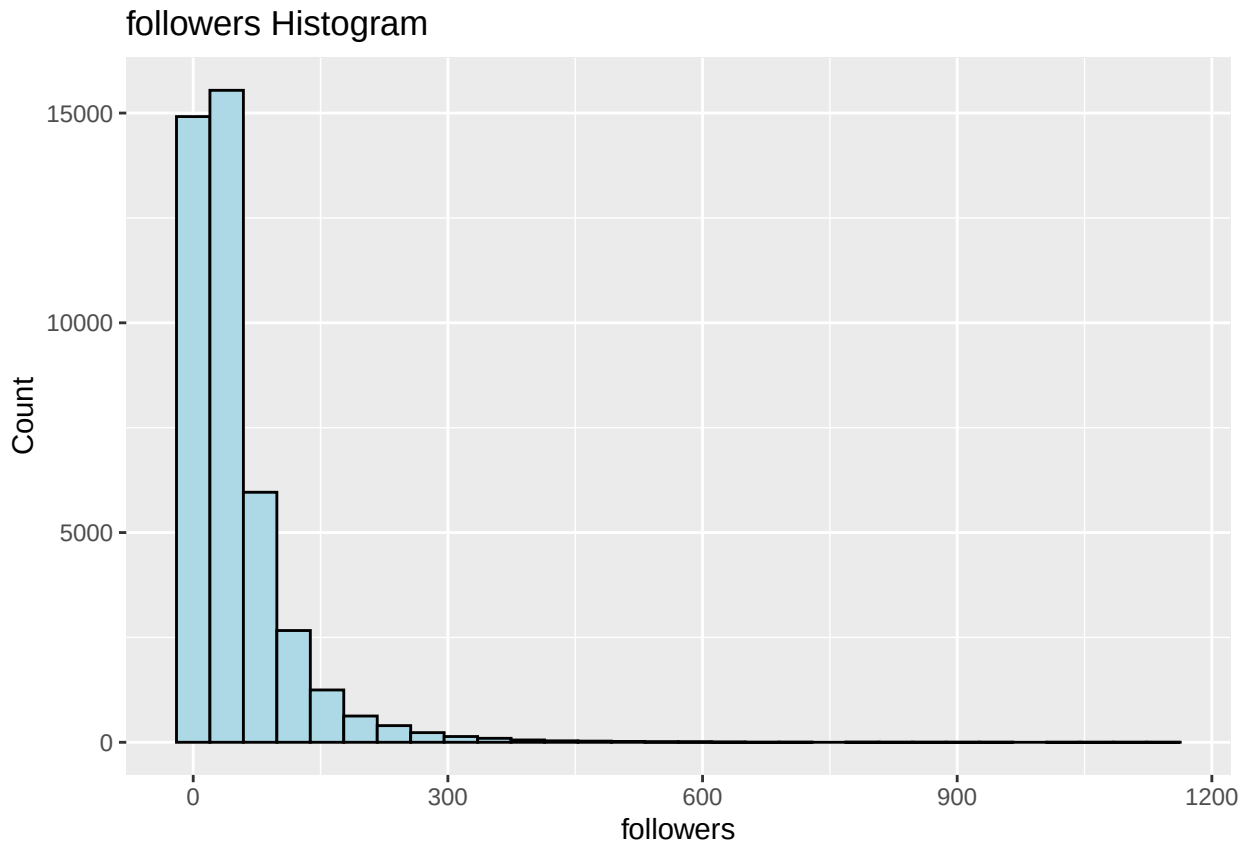


followers

This graph heavily skewed to the right. More than 97% of houses have less than 300 followers. Most houses have number of followers below 300. There are 5617 houses with 0 follower. There are also a total of 5 houses with more than 1000 followers. Among them, there is 1 house has significant low total price compare with other houses. However, it also has relative small area and is 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`.
```




```
data2017 %>%
  select(id,followers,totalPrice,price,square,district) %>%
  slice_max(followers,n=10)
```

```
##              id followers totalPrice  price square district
## 1  101101590746      1143      380.0 116780  32.54       10
## 2  101100732988      1085       48.0  26667  18.00        6
## 3  101102303199      1045      514.0  87415  58.80        8
## 4  101100746033      1015      297.5  78393  37.95        7
## 5  101101451868       945      252.0  35478  71.03        6
## 6  101102156364       928      192.0  38248  50.20        2
## 7  101102240992       922      263.0  37242  70.62        7
## 8  101100791393       908      365.0  33128 110.18        6
## 9  101101266281       877      310.0  45602  67.98        7
## 10 101102001331       864      395.0  65550  60.26        1
```

```
lessthan300 <- data2017 %>%
  filter(followers<=300)
nrow(lessthan300)
```

```
## [1] 41600
```

```
portion_lessthan300 = nrow(lessthan300)/43217
portion_lessthan300
```

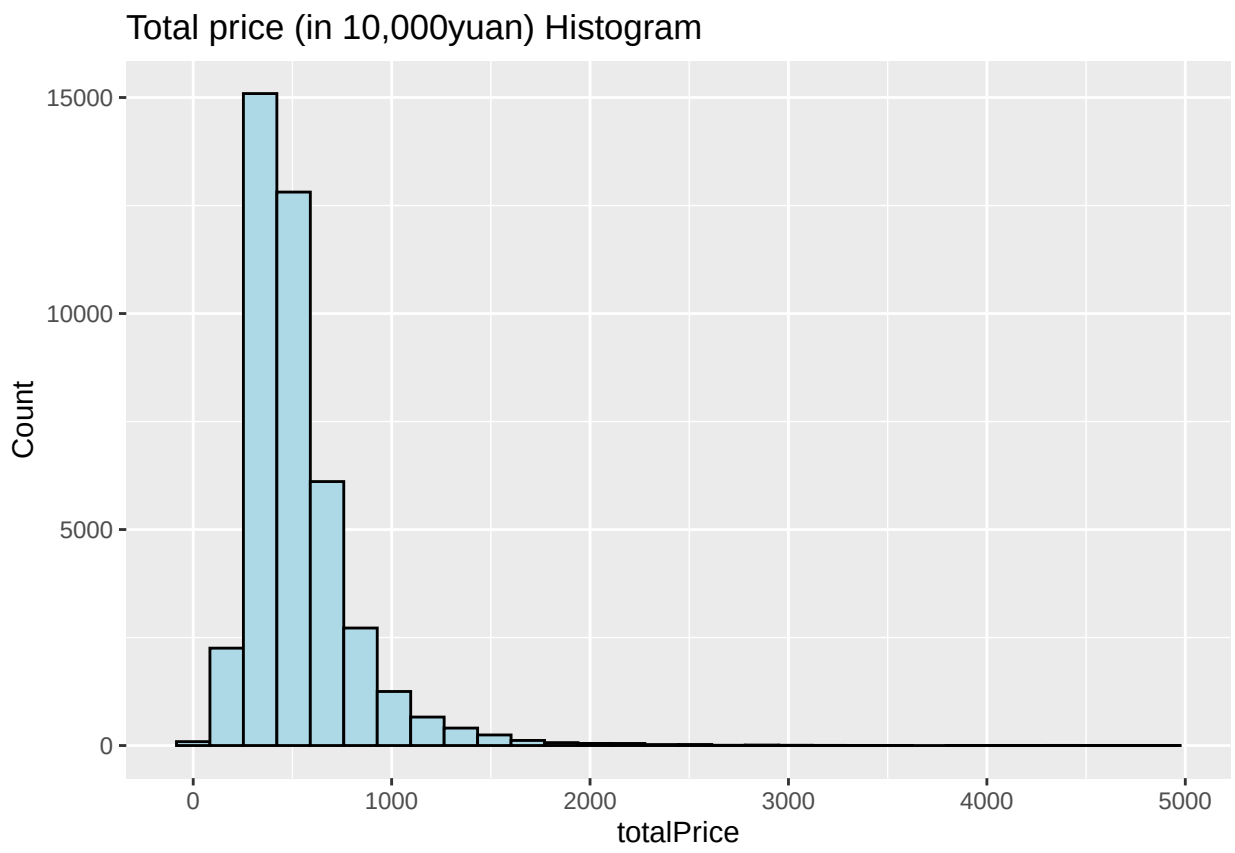
```
## [1] 0.9625842
```

totalPrice

This graph skews to the right.

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

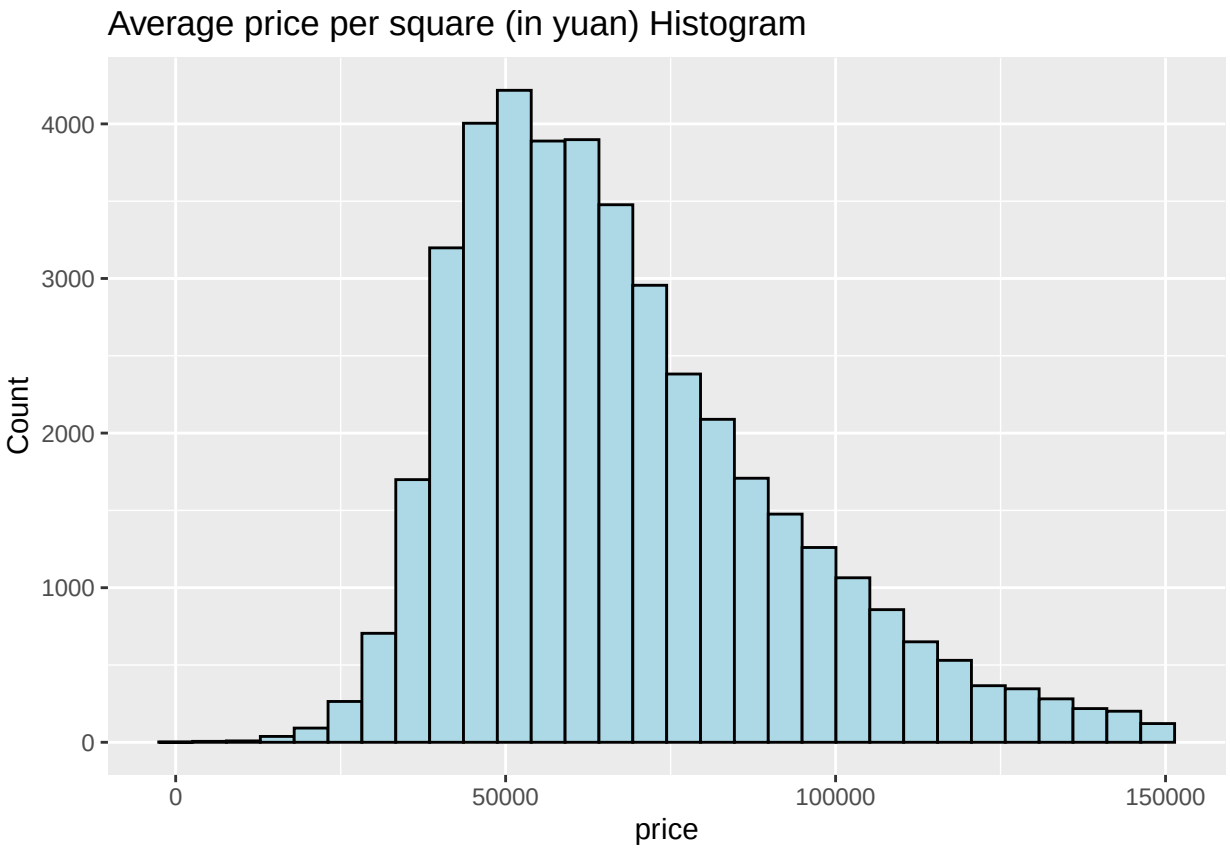
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



price

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

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

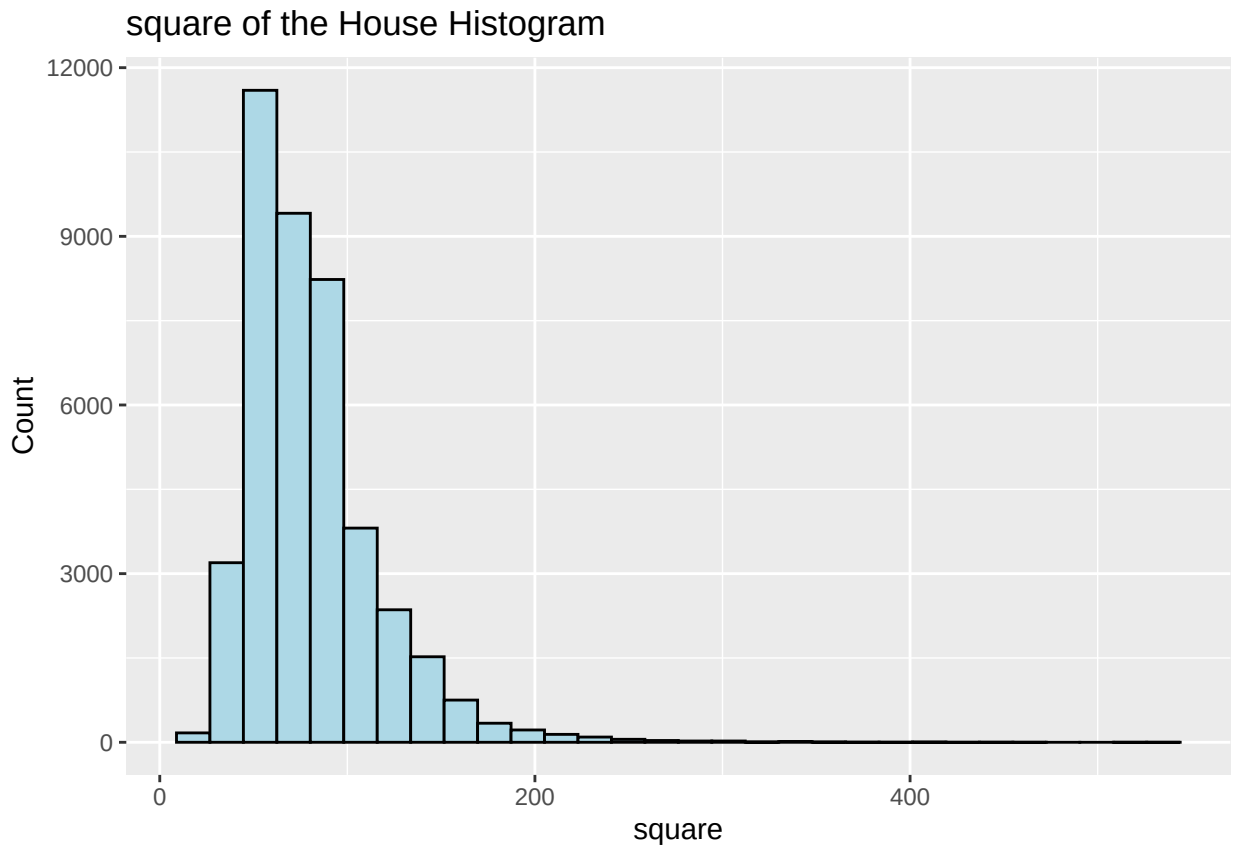


square

This graph skews to the right.

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

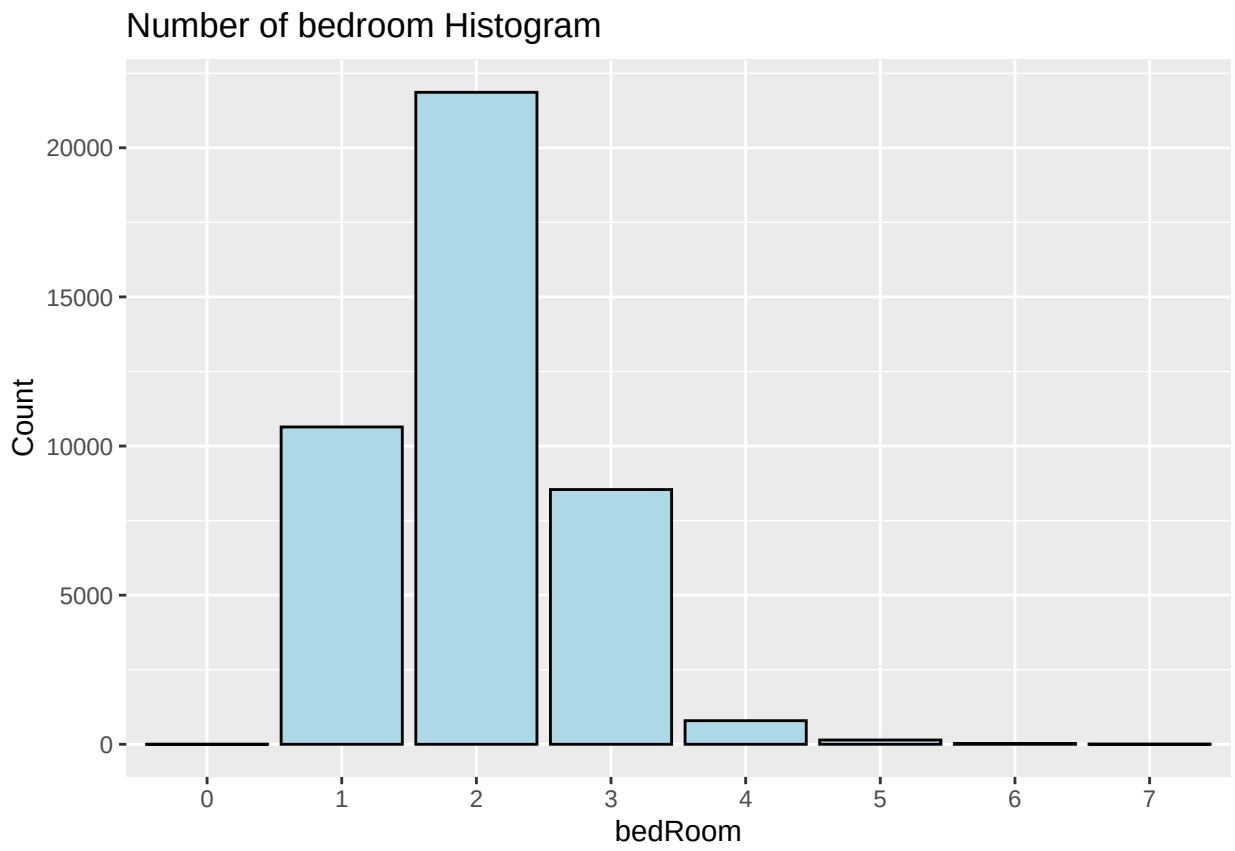
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



bedRoom

One thing need to be noticed is that in this dataset, the variable 'livingRoom' could be understand as 'bedroom' in the US. Most of the houses have 2 bedrooms.

```
ggplot(data2017, aes(x=livingRoom))+
  geom_bar(fill='lightblue', color='black') + # use geom_bar() instead of geom_histogram()
  xlab("bedRoom")+ylab("Count")+ggtitle("Number of bedroom Histogram")
```



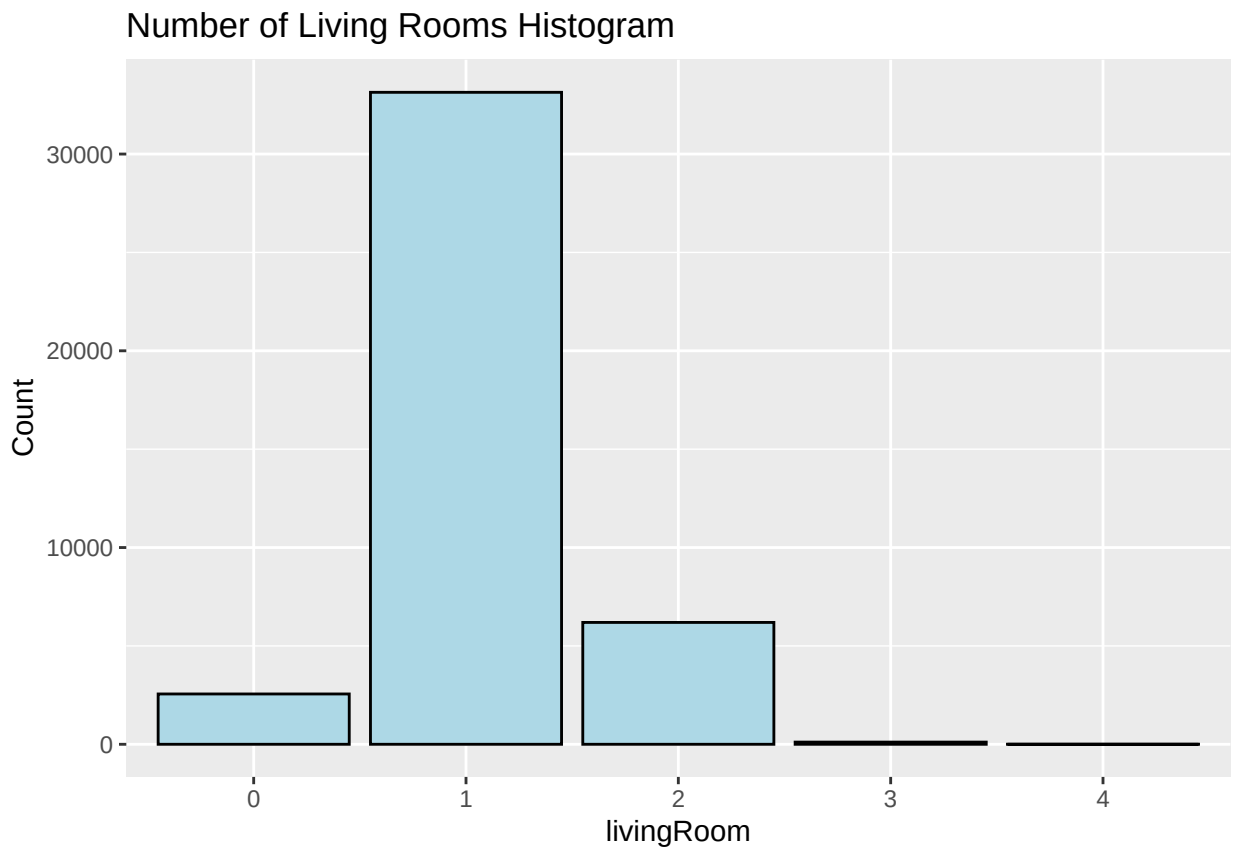

```
# 1 house with 0 living room
livingRoom0 <- data2017 %>%
  filter(livingRoom==0)
nrow(livingRoom0)
```

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

livingRoom

The number of living rooms for all houses is between 0 to 4. Most houses sold in 2017 have 0 to 2 living rooms and nearly 78% of houses sold in 2017 have 1 living room. Houses with 0 living room is the third greatest choice, and it is reasonable for a house to have no living room, especially in the city.

```
ggplot(data2017, aes(x=drawingRoom))+
  geom_bar(fill='lightblue', color='black') + # use geom_bar() instead of geom_histogram()
  # scale_y_continuous(trans="log10") +
  xlab("livingRoom")+
  ylab("Count")+
  ggtitle("Number of Living Rooms Histogram")
```



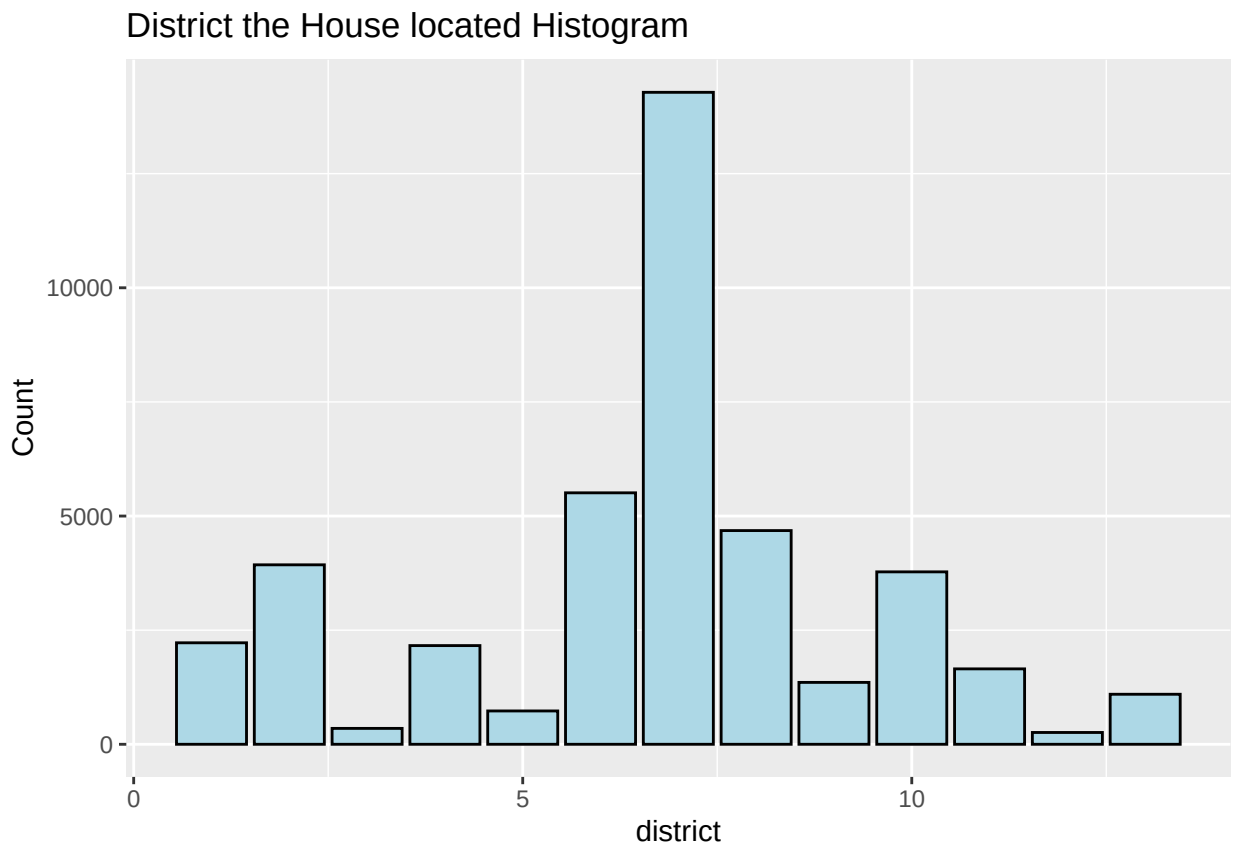

```
drawingRoom1 <- data2017 %>%
  filter(drawingRoom==1) %>%
  nrow()
portion_dr1 = drawingRoom1 / 43217
portion_dr1
```

```
## [1] 0.7667353
```

district (Categorical Variable)

I want to change the numbers into district names on the axis

```
library(ggplot2)
ggplot(data2017, aes(x=district))+
  geom_bar(fill='lightblue', color='black') +
  xlab("district")+
    ylab("Count")+
  ggtitle("District the House located Histogram")
```



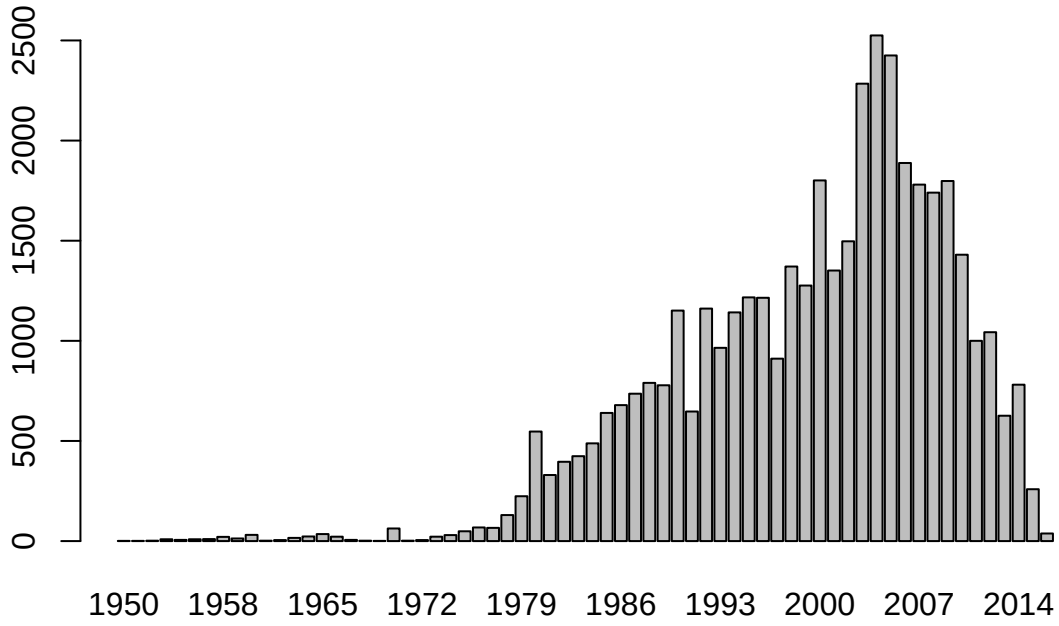

```
# display all x labels
#district_text <- c("DongCheng", "FengTai", "YiZhuang", "DaXing", "FangShan", "ChangPing", "ChaoYang",
```

summary statistics and distributional shape - construction time Construction time would be the year that the property is built

```
table(data2017$constructionTime)
```

```
##
## 1950 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966
##    1    1    2    9    6    9   10   21   13   31    2    5   16   23   35   22
## 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982
##    6    2    1   63    2    5   22   30   49   68   66  130  224  547  330  396
## 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998
##  424  488  640  679  736  790  778 1151  647 1161  965 1142 1217 1215  911 1371
## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014
## 1276 1801 1351 1497 2284 2525 2425 1888 1780 1740 1798 1430 1000 1043  626  781
## 2015 2016
##  259   38
```

```
barplot(table(data2017$constructionTime))
```

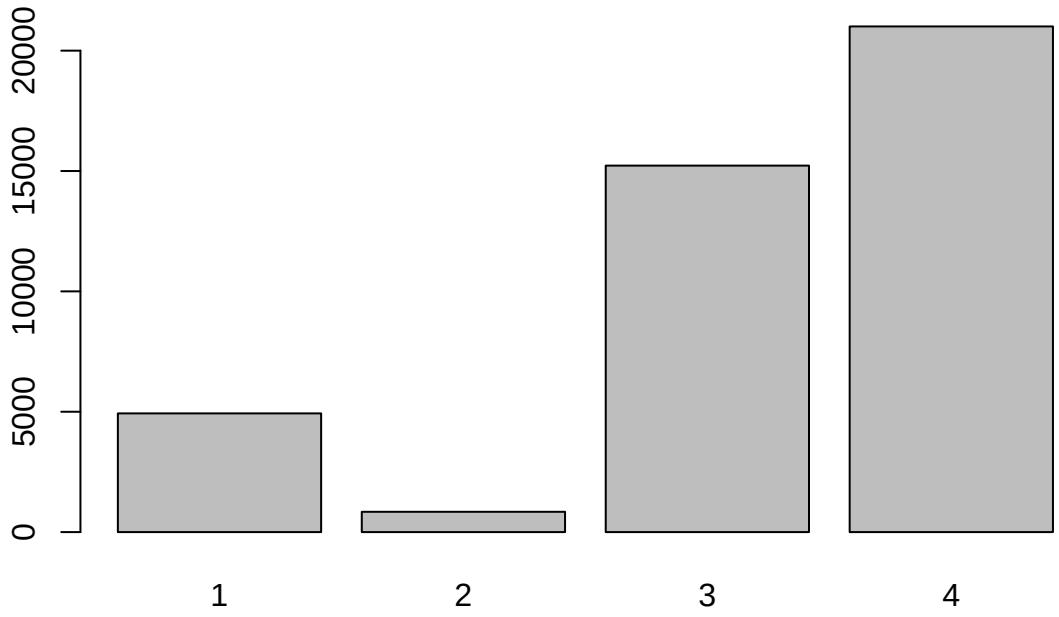


- renovation condition Renovation Condition refers to the readiness for the next property owner to move in with 4 being the most to move in and 1 being the least ready to move in

```
table(data2017$renovationCondition)
```

```
##
##     1     2     3     4
##  4931   843 15224 21006
```

```
barplot(table(data2017$renovationCondition))
```

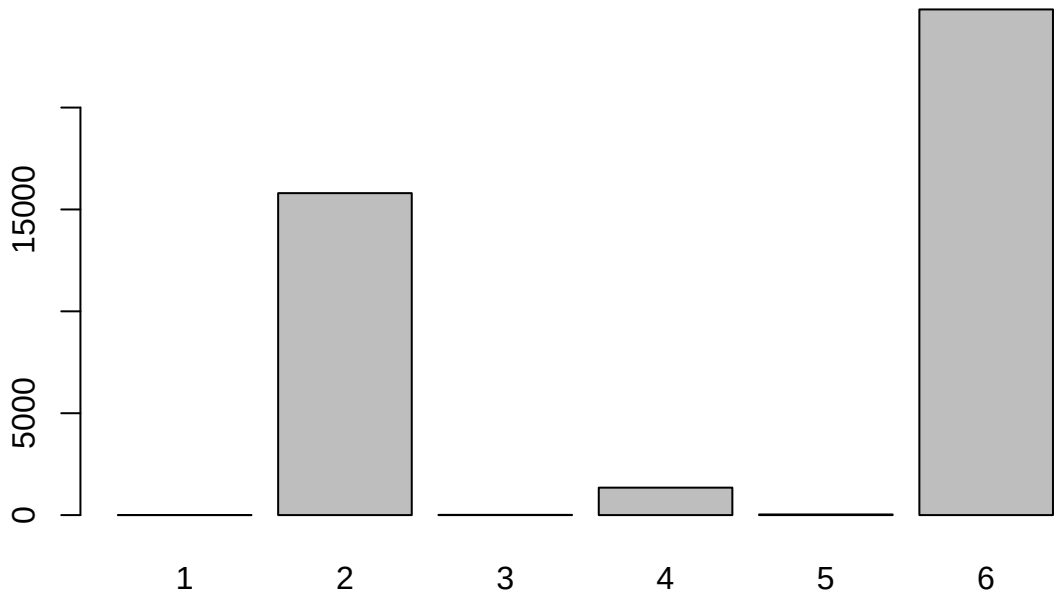


- building structure Building Structure would be the main material that the building is made out of

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

```
##
##     1     2     3     4     5     6
##     2 15800    10  1346    30 24816
```

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



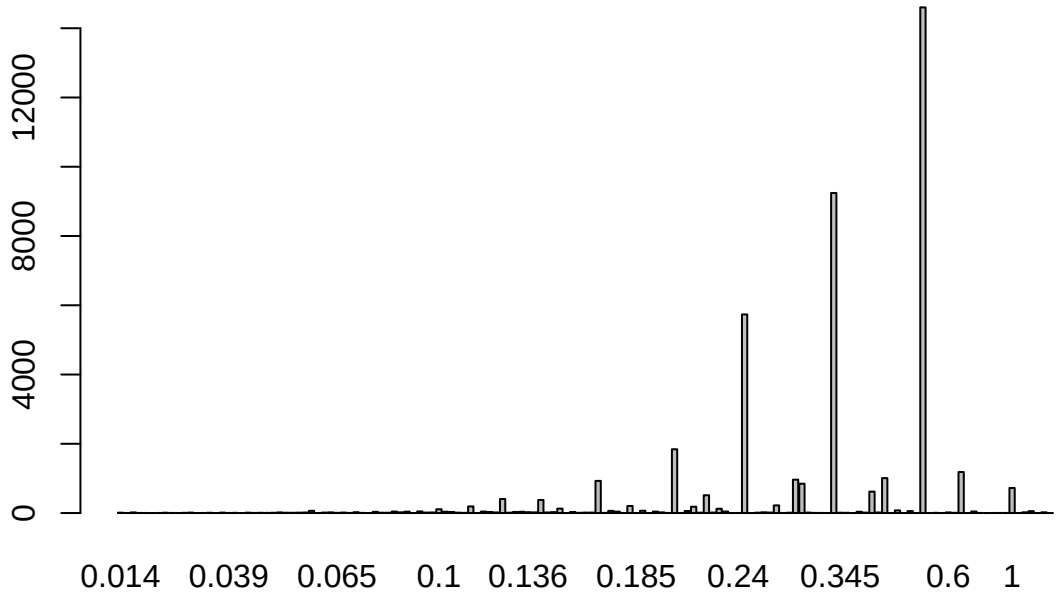
- ladder ratio

```
table(data2017$ladderRatio)
```

```
##
## 0.014 0.015  0.02 0.022 0.023 0.025 0.026 0.028 0.029  0.03 0.031 0.033 0.034
##    10     2    18     3     2     1     2     9     3     2     6    12     1
## 0.035 0.036 0.037 0.038 0.039  0.04 0.041 0.042 0.043 0.045 0.047 0.048  0.05
##     2     8     1    12     3     6     1    11     4     7     5     8    18
## 0.053 0.054 0.056 0.057 0.059  0.06 0.061 0.062 0.065 0.067 0.069 0.071 0.074
##     9     8    11    15    64     1    13    19     6    14     4    24     5
## 0.075 0.077  0.08 0.081 0.083 0.086 0.087 0.088 0.091 0.095 0.098   0.1 0.103
##     1    35     9     9    42    20    39     1    46    14    16   109    38
## 0.105 0.107 0.108 0.111 0.114 0.118  0.12 0.121 0.125 0.129  0.13 0.133 0.136
##    31    11     6   191     1    42    29    18   406    10    31    37    25
## 0.138 0.143 0.148  0.15 0.154 0.156 0.158  0.16 0.161 0.162 0.167 0.171 0.174
##    19   377    16    26   128     1    31     7    12    13   929     5    62
## 0.176 0.179 0.182 0.185 0.188 0.189  0.19 0.192 0.198   0.2 0.208 0.211 0.214
##    40     2   204     1    66     1    41    19     3  1842     2    59   182
## 0.217 0.222 0.227 0.231 0.235 0.238  0.24  0.25 0.255 0.261 0.263 0.267 0.273
##    11   514    15   124    45     1     2  5735     1    11    21    14   221
## 0.276 0.278 0.286   0.3 0.304 0.308 0.312 0.318 0.333 0.345 0.346 0.353 0.364
##     2     9   963   847    13     5     3     2  9242     9     7     1    37
##  0.37 0.375 0.385   0.4 0.417 0.429 0.438 0.444 0.467   0.5 0.538 0.571 0.583
##    12   618    16  1009     1    76     2    56     1 14602     2     6     2
##   0.6 0.625 0.667 0.714  0.75   0.8 0.818 0.833  0.87 0.889     1  1.25 1.333
##    18     7  1184     1    45     3     1     2     3     3   724     2    21
```

```
##   1.5 1.667     2     3
##    54     1    21     1
```

```
barplot(table(data2017$ladderRatio))
```

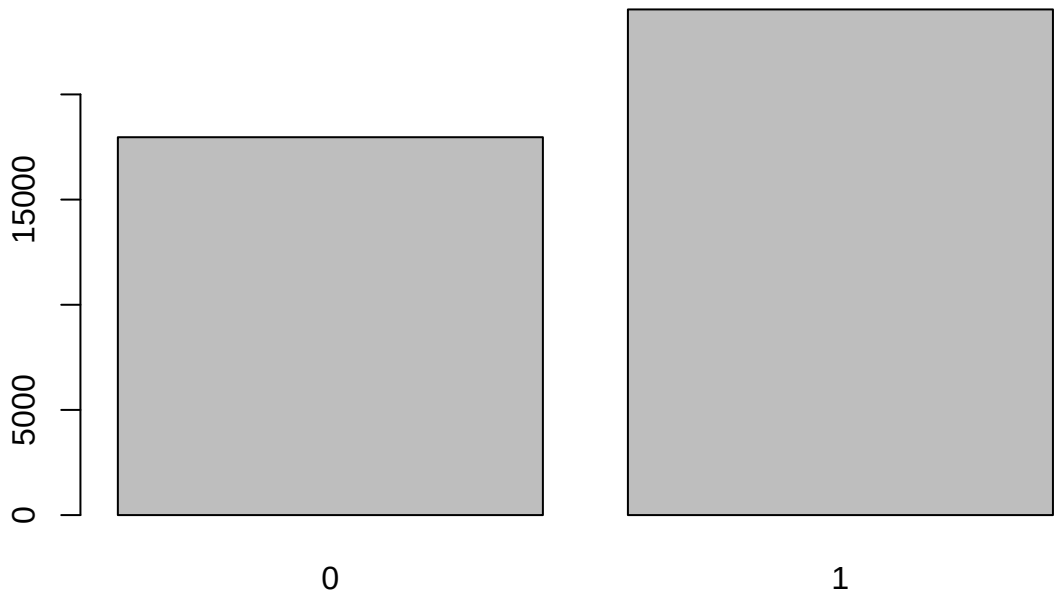


- elevator

```
table(data2017$elevator)
```

```
##
##     0     1
## 17964 24040
```

```
barplot(table(data2017$elevator))
```

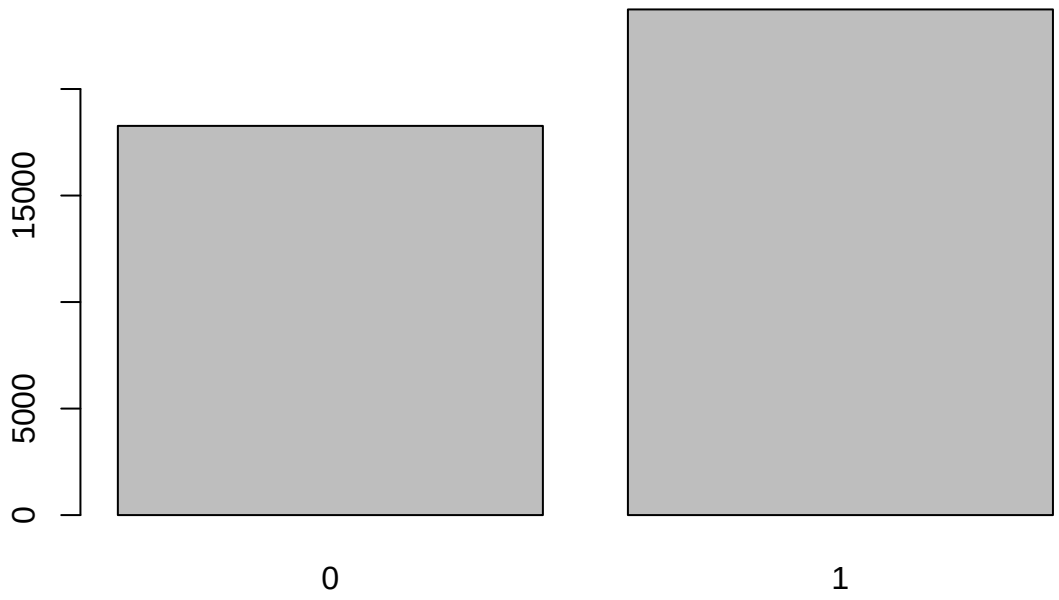


- five years property

```
table(data2017$fiveYearsProperty)
```

```
##
##     0     1
## 18269 23735
```

```
barplot(table(data2017$fiveYearsProperty))
```

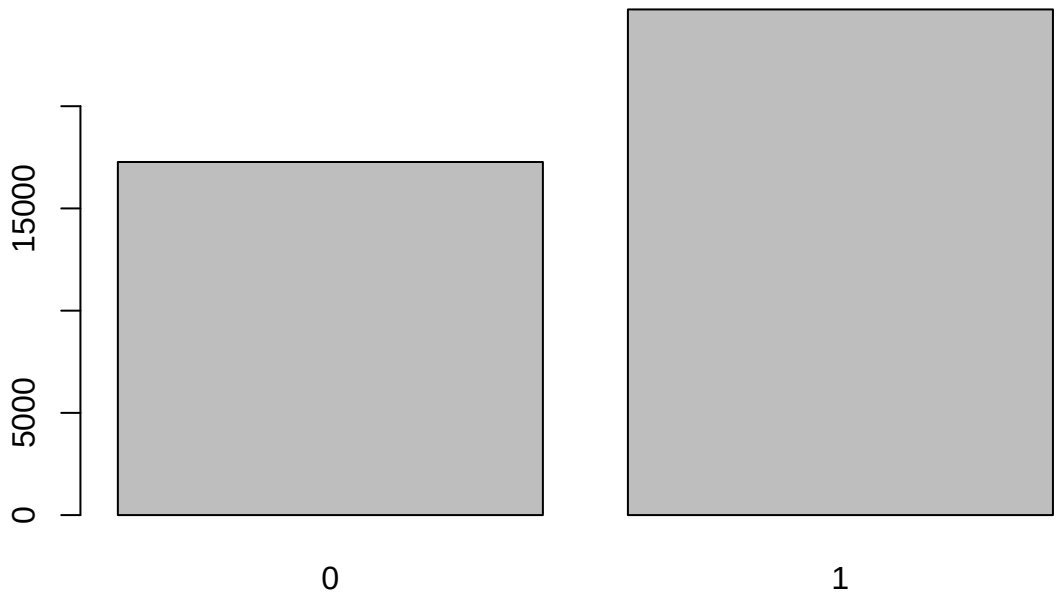


- subway

```
table(data2017$subway)
```

```
##
##     0     1
## 17271 24733
```

```
barplot(table(data2017$subway))
```

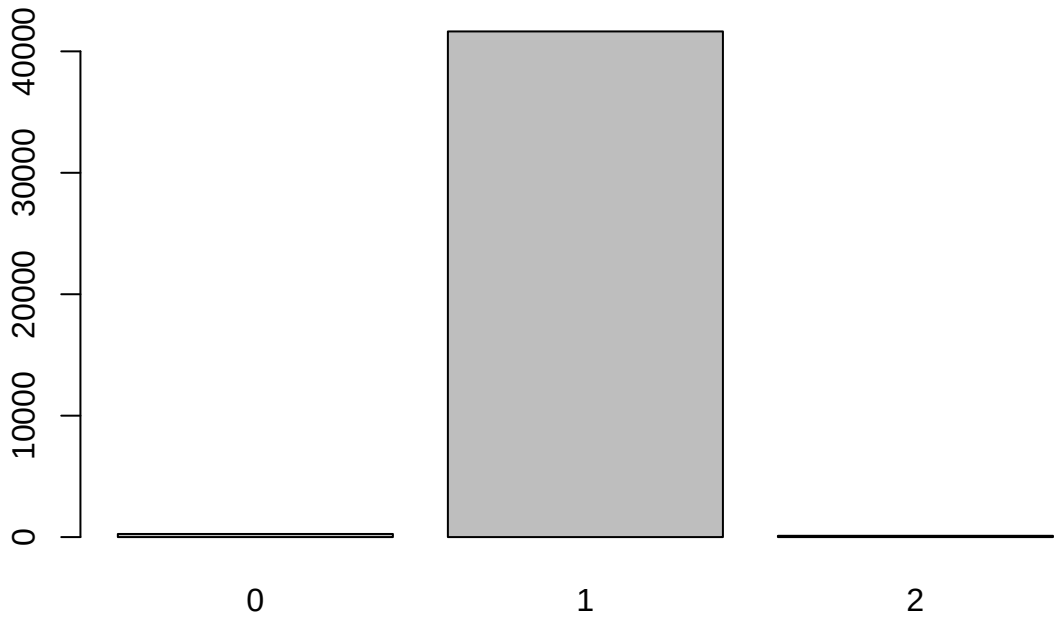


- kitchen Kitchen would be the number of kitchens the property has, with 1 kitchen being the most popular

```
table(data2017$kitchen)
```

```
##
##     0     1     2
##   257 41641   106
```

```
barplot(table(data2017$kitchen))
```

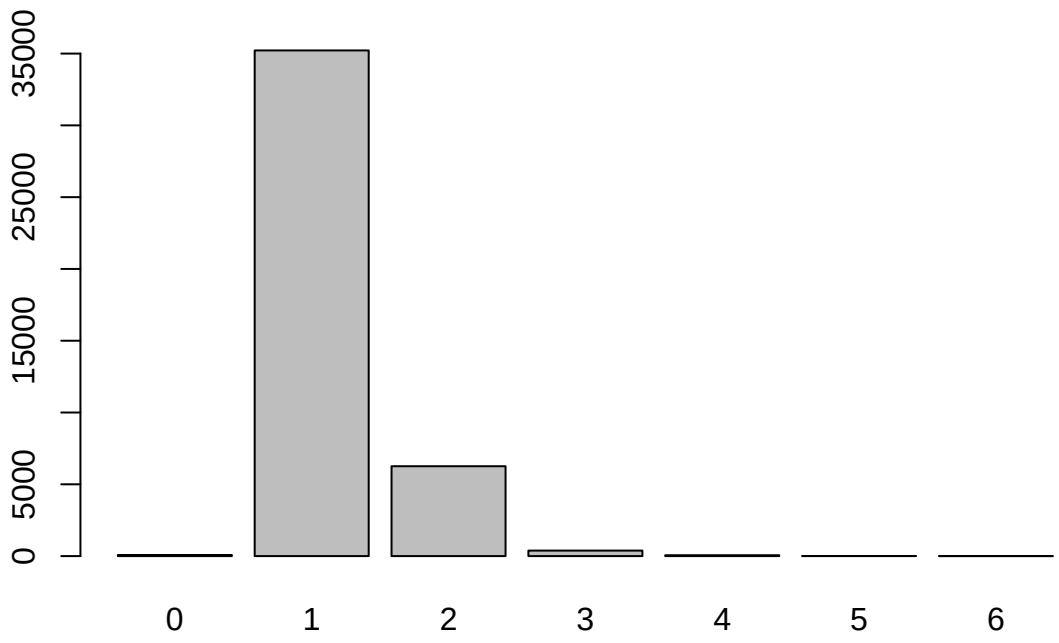


- bathroom Bathroom would be the number of bathrooms the property has, with 1 bathroom being the most popular

```
table(data2017$bathRoom)
```

```
##
##     0     1     2     3     4     5     6
##    77 35219  6259   383    59     6     1
```

```
barplot(table(data2017$bathRoom))
```



potential relationships that may exist in the data - construction time, renovation condition

- construction time, building structure