INFX 502

DATA ANALYSIS USING R TO ANALYZE US HOUSING PRICES

Name: Pinky Sitikhu

ULID: C00477712

Master's in Informatics, Fall 2022

TABLE OF CONTENTS

Introduction

This dataset is about the real estate listings in the US broken by state and zip code. The dataset was created by scrapping the information from a realtor website, a real estate listing website operated by the News Corp subsidiary Move, Inc and based in Santa Clara, California. This website is the second most visited real estate listing website in the US with over 100 million monthly active users.

Dataset Description

The CSV format of the dataset was obtained from Kaggle, which is used in this project for analysis and visualization. This dataset has 900K+ observations and contains house sale prices and features of the house in 12 different columns. The details about each feature of the dataset are explained below:

Variable Name	Description	Type
status	Housing status (on sale or other option)	character
price	Price in USD	numeric
bed	Bedroom Count	numeric
bath	Bathroom Count	numeric
acre_lot	Acre Lot	numeric
full_address	Full Address of the Location	character
street	Street Name	character
city	City Name	character
state	State Name	character
zip_code	Zip Code	numeric
house_size	House size in sqft	numeric
sold_date	The date when the house is sold	date

Expectation

The general expectation of this analysis is to explore the housing dataset and analyze and observe the relationship between variables. From this analysis, we wanted to see what features strongly affect housing prices. Our assumption is that all of these features affect the price of the house in a positive order. This means as these factor increases or is better, the price increases. For example, the price of a house with 5 beds and 4 baths will be higher than the price of a house with 1 bed and 1 bath.

Import Libraries

To start the exploration, first, we import all the required packages and libraries. We use `readr` package to import CSV files for our experimentation. Other libraries that we are using are `ggplot2, tidyverse, moments` and `ggcorrplot`. The `moments` package is used to compute the skewness and kurtosis and `ggcorrplot` is used to compute the correlation between all the variables. The packages that are not installed can be installed using `install.packages()` command.

```
# install.packages("moments")
library(moments)
library(ggcorrplot)
library(readr)
library(ggplot2)
library(tidyverse)
```

Loading Dataset

Since we have the dataset in CSV format, we used the `read_csv` function to load the dataset. Further, to briefly observe the dataset, `head` command is used to preview the rows of our dataset.

```
data <- read_csv("realtor-data.csv", show_col_types = FALSE)
head(data)
```

A tibble: 6 × 12

status <chr>	price <dbl>	bed <dbl>	bath <dbl>	acre_lot <dbl>	full_address <chr>
for_sale	105000	3	2	0.12	Sector Yahuecas Titulo # V84, Adjuntas, PR, 00601
for_sale	80000	4	2	0.08	Km 78 9 Carr # 135, Adjuntas, PR, 00601
for_sale	67000	2	1	0.15	556G 556-G 16 St, Juana Diaz, PR, 00795
for_sale	145000	4	2	0.10	R5 Comunidad El Paraso Calle De Oro R-5 Ponce, Ponce, PR, 00731
for_sale	65000	6	2	0.05	14 Navarro, Mayaguez, PR, 00680
for_sale	179000	4	3	0.46	Bo Calabazas San Sebastian, San Sebastian, PR, 00612

6 rows | 1-6 of 12 columns

A tibble: 6 × 12

street <chr>	city <chr>	state <chr>	zip_code <dbl>	house_size <dbl>	sold_date <date>
Sector Yahuecas Titulo # V84	Adjuntas	Puerto Rico	601	920	<NA>
Km 78 9 Carr # 135	Adjuntas	Puerto Rico	601	1527	<NA>
556G 556-G 16 St	Juana Diaz	Puerto Rico	795	748	<NA>
R5 Comunidad El Paraso Calle De Oro R-5 Ponce	Ponce	Puerto Rico	731	1800	<NA>
14 Navarro	Mayaguez	Puerto Rico	680	*NA*	<NA>
Bo Calabazas San Sebastian	San Sebastian	Puerto Rico	612	2520	<NA>

6 rows | 7-12 of 12 columns

Dataset Structure

The figure with the structure of the dataset is given by the `str` command.

```
str(data)
```

```
## spec_tbl_df [923,159 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ status      : chr [1:923159] "for_sale" "for_sale" "for_sale"
"for_sale" ...
##  $ price       : num [1:923159] 105000 80000 67000 145000 65000 179000
50000 71600 100000 300000 ...
##  $ bed         : num [1:923159] 3 4 2 4 6 4 3 3 2 5 ...
##  $ bath        : num [1:923159] 2 2 1 2 2 3 1 2 1 3 ...
##  $ acre_lot    : num [1:923159] 0.12 0.08 0.15 0.1 0.05 0.46 0.2 0.08 0.09
7.46 ...
##  $ full_address: chr [1:923159] "Sector Yahuecas Titulo # V84, Adjuntas,
PR, 00601" "Km 78 9 Carr # 135, Adjuntas, PR, 00601" "556G 556-G 16 St, Juana
Diaz, PR, 00795" "R5 Comunidad El Paraso Calle De Oro R-5 Ponce, Ponce, PR,
00731" ...
##  $ street      : chr [1:923159] "Sector Yahuecas Titulo # V84" "Km 78 9
Carr # 135" "556G 556-G 16 St" "R5 Comunidad El Paraso Calle De Oro R-5
Ponce" ...
##  $ city        : chr [1:923159] "Adjuntas" "Adjuntas" "Juana Diaz" "Ponce"
...
##  $ state       : chr [1:923159] "Puerto Rico" "Puerto Rico" "Puerto Rico"
"Puerto Rico" ...
##  $ zip_code    : num [1:923159] 601 601 795 731 680 612 639 731 730 670
...
##  $ house_size  : num [1:923159] 920 1527 748 1800 NA ...
##  $ sold_date   : Date[1:923159], format: NA NA ...
```

Data Preprocessing

Conversion into required datatype

In this dataset, status, street, city, and state variables are of character type. Similarly, zip_code is of numeric type. All these variables need to be converted into factor types for further processing. We converted them by using the following command.

```
data$status <- as.factor(data$status)
data$street <- as.factor(data$street)
data$city <- as.factor(data$city)
data$state <- as.factor(data$state)
data$zip_code <- as.factor(data$zip_code)
```

After converting their datatype, we verify the dataset using `str` command.

```
str(data)
```

```
## spec_tbl_df [923,159 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ status      : Factor w/ 2 levels "for_sale","ready_to_build": 1 1 1 1 1
1 1 1 1 1 ...
##  $ price       : num [1:923159] 105000 80000 67000 145000 65000 179000
50000 71600 100000 300000 ...
##  $ bed         : num [1:923159] 3 4 2 4 6 4 3 3 2 5 ...
##  $ bath        : num [1:923159] 2 2 1 2 2 3 1 2 1 3 ...
##  $ acre_lot    : num [1:923159] 0.12 0.08 0.15 0.1 0.05 0.46 0.2 0.08 0.09
7.46 ...
##  $ full_address: chr [1:923159] "Sector Yahuecas Titulo # V84, Adjuntas,
PR, 00601" "Km 78 9 Carr # 135, Adjuntas, PR, 00601" "556G 556-G 16 St, Juana
Diaz, PR, 00795" "R5 Comunidad El Paraso Calle De Oro R-5 Ponce, Ponce, PR,
00731" ...
##  $ street      : Factor w/ 110324 levels "0 0 Bay St /Judson St /Church
St",..: 109072 106514 80404 108480 17948 104061 74233 60107 41528 108351 ...
##  $ city        : Factor w/ 2542 levels "Abbot","Aberdeen",..: 15 15 1078
1765 1318 1947 415 1765 1765 1149 ...
##  $ state       : Factor w/ 18 levels "Connecticut",..: 10 10 10 10 10 10
10 10 10 10 ...
##  $ zip_code    : Factor w/ 3191 levels "601","602","603",..: 1 1 95 67 39
8 19 67 66 34 ...
##  $ house_size  : num [1:923159] 920 1527 748 1800 NA ...
##  $ sold_date   : Date[1:923159], format: NA NA ...
```

The number of rows and columns in the dataset can be obtained using the following commands.

```
nrow(data)
ncol(data)
```

Output:

```
[1] 923159
[1] 12
```

We can see that there are more than 900k rows and 12 columns in the dataset. From the given structure of the dataset, we can see that the variables are either factor or numeric type, and of date type.

Checking missing values

Generally, in large datasets, there are a lot of missing values in different columns. So, we checked whether missing values exist in our dataset by using the following command.

```
any(is.na(data))
```

```
> [1] TRUE
```

This indicates that there were some missing values in the dataset. So, we further investigated to check which of the columns has missing values.

```
apply(is.na(data), 2, any)
```

```
     status         price           bed          bath     acre_lot full_address
street          city         state      zip_code    house_size
       FALSE          TRUE          TRUE          TRUE          TRUE         FALSE
TRUE          TRUE         FALSE          TRUE          TRUE
   sold_date
        TRUE
```

We saw that most of the columns have missing values. Similarly, we checked which columns have the most missing values, so that we can take appropriate steps to process the missing values.

```
colSums(is.na(data))
     status         price           bed          bath     acre_lot
full_address       street          city         state     zip_code
house_size
           0           71       131703        115192       273623
0          2138           74             0          205        297843
   sold_date
      466763
```

From the output above, we saw that there were more than 400k observations that do not have sold dates. There are certain columns that might not have a significant influence/impact on house prices. Based on the observations, the columns like "status", "street", "full_address", and "sold_date" can be removed as they do not seem to have a direct influence over other variables. But before removing the "status" column, we performed some processing to ensure the removal

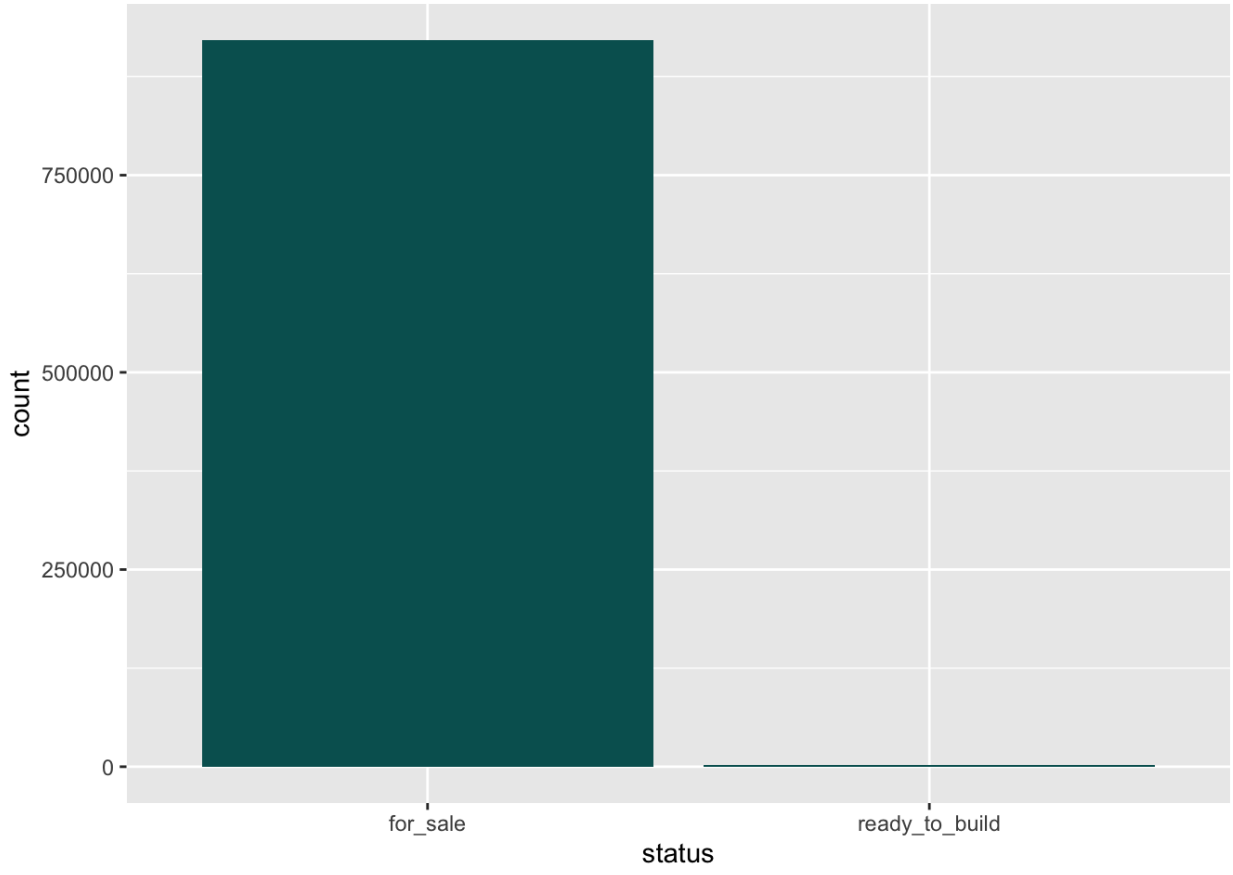
of this column. For that, we checked what are the unique values in the status column and plotted their distribution.

```
unique(data$status)
```

```
> [1] for_sale       ready_to_build
Levels: for_sale ready_to_build
```

There were two unique values: for_sale and ready_to_build. Now, we plotted their distribution using ggplot2.

```
ggplot(data, aes(x=reorder(status, status, function(x)-length(x)))) +
geom_bar(fill='#006060') +  labs(x='status')
```



From the plot above, we clearly saw that there were a huge number of observations for `for_sale` category as compared to `ready_to_build`. Since there were a large data disparity and imbalance category values, we excluded this column from further exploration.

Removing less significant columns

Then, we removed the columns that are less important and created a new dataframe.

```
new <- c("price", "bed", "bath", "acre_lot", "city", "state",
"zip_code", "house_size")
df <- data[new]
head(df)
```

A tibble: 6 x 8

price <dbl>	bed <dbl>	bath <dbl>	acre_lot <dbl>	city <fctr>	state <fctr>	zip_code <fctr>	house_size <dbl>
105000	3	2	0.12	Adjuntas	Puerto Rico	601	920
80000	4	2	0.08	Adjuntas	Puerto Rico	601	1527
67000	2	1	0.15	Juana Diaz	Puerto Rico	795	748
145000	4	2	0.10	Ponce	Puerto Rico	731	1800
65000	6	2	0.05	Mayaguez	Puerto Rico	680	*NA*
179000	4	3	0.46	San Sebastian	Puerto Rico	612	2520

6 rows

We checked the missing values in each observation and handle them. For that, first, we created a function that gave the count of missing values and non-missing values in each column.

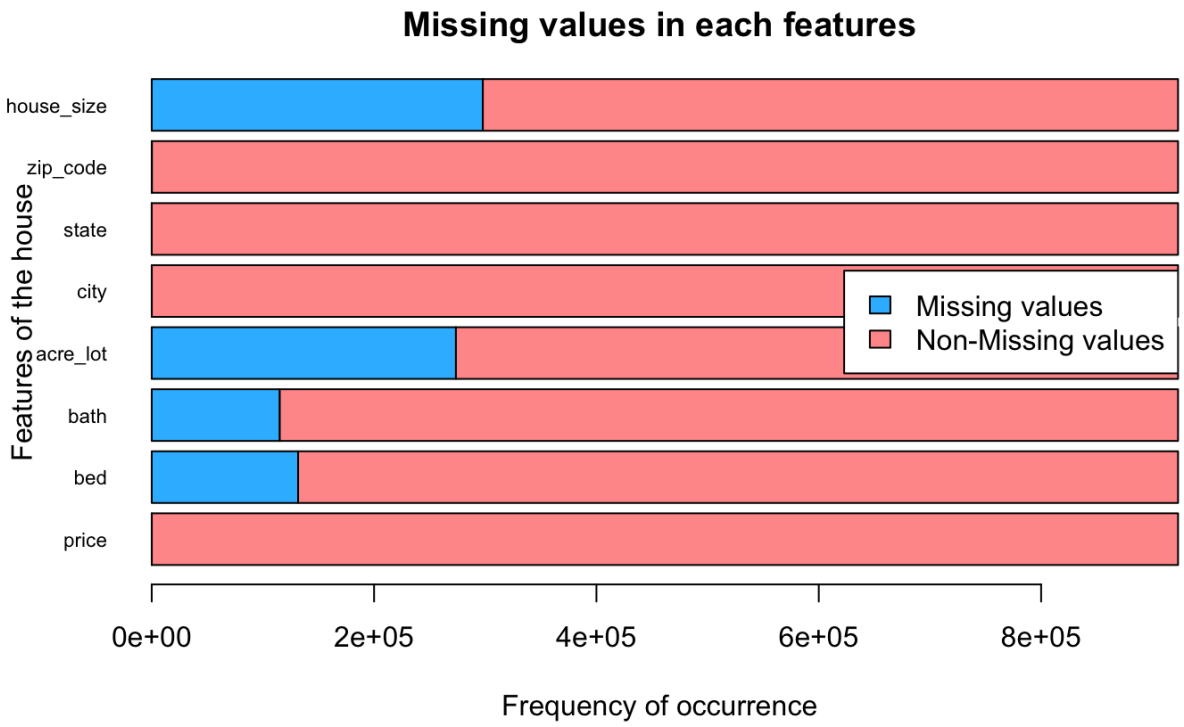
```
missing_count_func <- function(df){
  m<-c()
  for (i in colnames(df)){
    x<-sum(is.na(df[,i]))
    # count missing value
    m<-append(m,x)
    # count non-missing value
    m<-append(m,nrow(df)-x)
  }

  a<-matrix(m, nrow = 2)
  rownames(a)<-c("TRUE", "FALSE")
  colnames(a)<-colnames(df)
  return(a)
}

f=missing_count_func(df)
f
```

```
>      price    bed   bath acre_lot   city  state zip_code house_size
TRUE      71 131703 115192   273623     74      0      205     297843
FALSE 923088 791456 807967   649536 923085 923159   922954     625316
```

We plotted these values in a barplot for easier understanding.



From this plot, we saw that a lot of missing values existed in house_size, acre_lot, bath, and bed variables. There are various ways to handle NA or missing values, but in this use case, removing NA values and less significant columns sounded promising as there were more than 900k observations in the dataset. Also, using mean or median values or applying a regression approach to fill in missing values will not provide an accurate result for predicting prices. So, we removed the NA values using the following command.

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

We again verified whether the NA values are completely removed or not and computed the total number of remaining rows.

```
any(is.na(df))
> [1] FALSE
```

```
nrow(df)
> [1] 421227
```

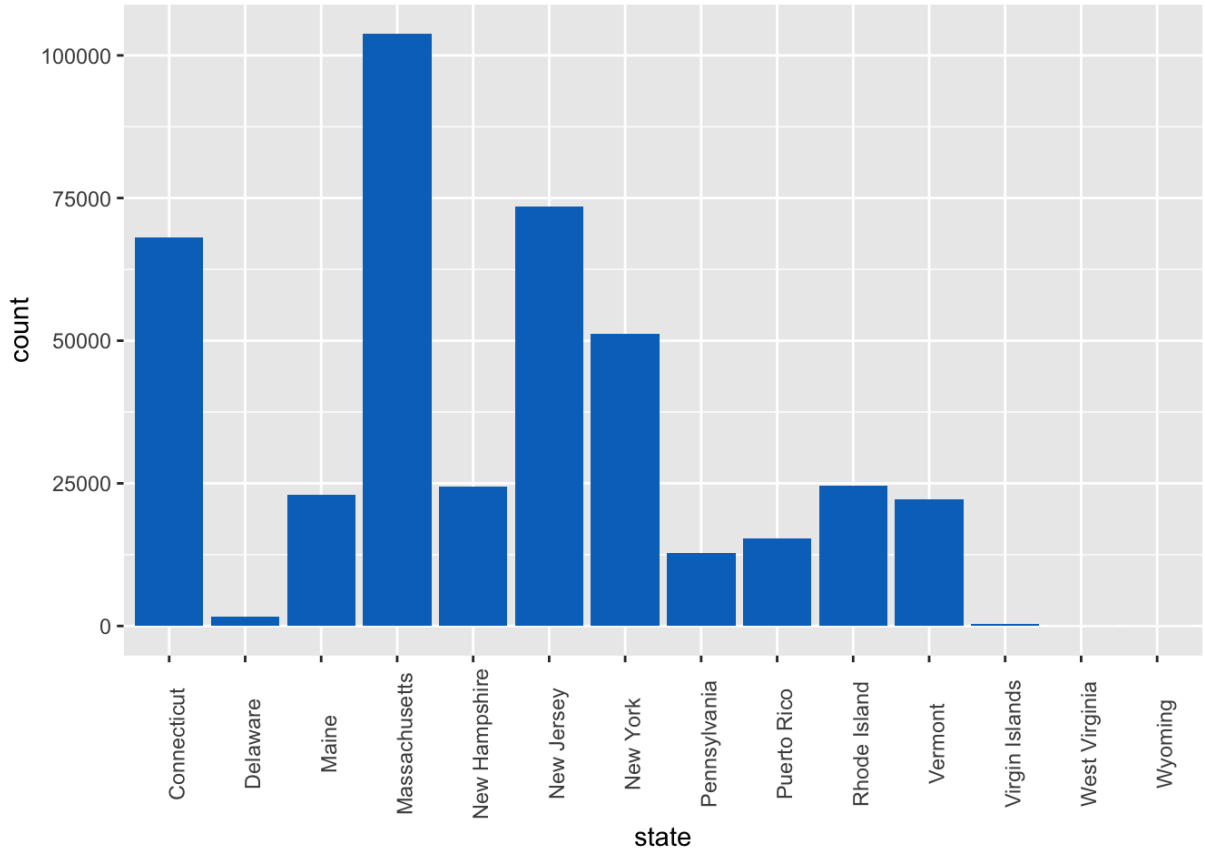
After removing all missing values from different columns, we still had more than 400k observations left. Though we removed a lot of observations based on missing values, we could proceed forward with 400k observations for further analysis.

Exploratory Data Analysis

Univariate Analysis

This analysis refers to analyzing only a single variable without its interaction with other variables. Using this analysis, first, we explored some categorical variables of our dataset. The variables city, state, and zip_code are categorical variables. We started by observing the distribution of data/observation with respect to states by plotting a histogram of the count of houses in each state.

```
ggplot(data=df, mapping=aes(x=state)) +
geom_histogram(stat="count")+geom_bar(fill="#0073C2FF")+theme(axis.te
xt.x=element_text(angle=90))
```

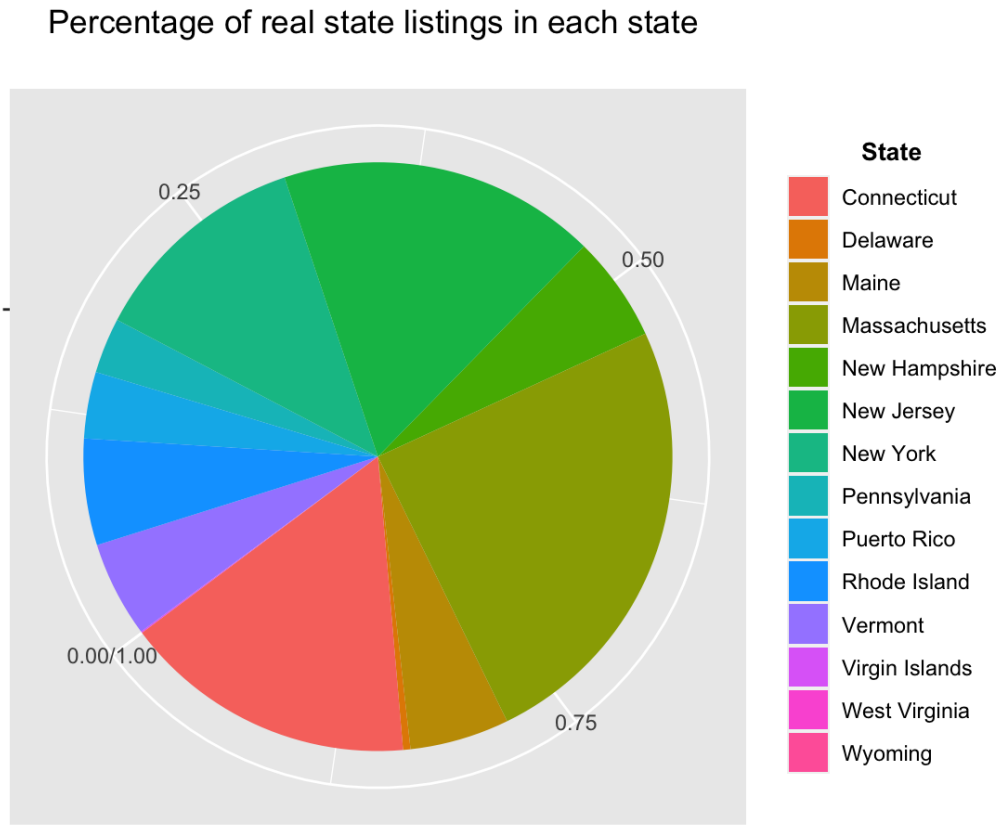


Interpretation

From this histogram, we saw that the state Massachusetts has the highest number of house listing in our dataset whereas the states Delaware, virgin islands, west virginia and Wyoming has significantly less number of house listing. We further analyzed deeper to see the percentage distribution of observation in each state with a pie chart. We grouped the dataset by state and counted the number of observations and computed its percentage over all the dataset and finally plot the result in a pie chart which is shown below:

```
df_state <- df %>%
  group_by(state) %>%
  count() %>%
  ungroup() %>%
  mutate(perc = `n` / sum(`n`)) %>%
  arrange(perc) %>%
  mutate(labels = scales::percent(perc))
ggplot(df_state, aes(x = "", y = perc, fill = state)) +
     geom_bar(width = 1, stat = "identity") +
     coord_polar(theta = "y", start = 180) +
```

```
     labs(x = "", y = "", title = "Percentage of real state listings
in each state \n",
          fill = "State") +
     theme(plot.title = element_text(hjust = 0.5),
           legend.title = element_text(hjust = 0.5, face="bold", size
= 10))
```



This chart clarified even more and showed that about 25% of the overall data were about different cities of Massachusetts state. This meant that lots of real estates housing are ready for sale mostly in big cities of the states like Massachusetts, New Hampshire, New Jersey, and New york.

Then, we performed some analysis using the numerical variable. Arithmetic mean, median, quartiles, and mode are commonly used summary statistics for measuring numerical data. Skewness and kurtosis are commonly used statistics to describe the shape of the distribution of a collection of data. So, we started by checking the summary of the dataset using the following command.

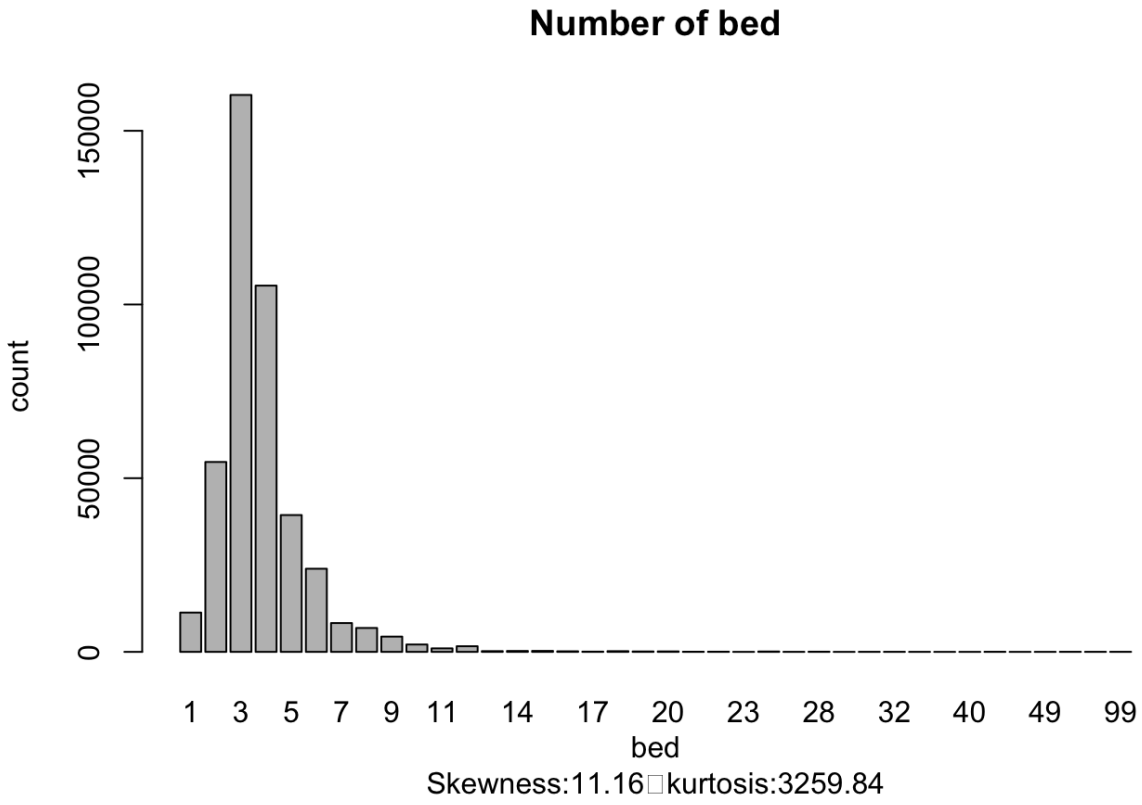
```
summary(df)
```

```
     price               bed              bath          acre_lot                 city                 state          zip_code    
 Min.   :      500   Min.   : 1.000   Min.   :  1.000   Min.   :0.00e+00   Boston       : 13414   Massachusetts:103770   6010   :  1896  
 1st Qu.:   285000   1st Qu.: 3.000   1st Qu.:  2.000   1st Qu.:1.10e-01   New York City:  8136   New Jersey   : 73571   2895   :  1766  
 Median :   459900   Median : 3.000   Median :  2.000   Median :2.60e-01   Philadelphia :  7663   Connecticut  : 68033   1201   :  1731  
 Mean   :   784424   Mean   : 3.814   Mean   :  2.704   Mean   :9.44e+00   Staten Island:  7064   New York     : 51270   6790   :  1421  
 3rd Qu.:   789000   3rd Qu.: 4.000   3rd Qu.:  3.000   3rd Qu.:8.90e-01   Brooklyn     :  6878   Rhode Island : 24620   2127   :  1400  
 Max.   :169000000   Max.   :99.000   Max.   :198.000   Max.   :1.00e+05   Bronx        :  4394   New Hampshire: 24454   6082   :  1399  
                                                                           (Other)      :373678   (Other)      : 75509   (Other):411614  
   house_size      
 Min.   :    122  
 1st Qu.:   1333  
 Median :   1894  
 Mean   :   2412  
 3rd Qu.:   2784  
 Max.   :1450112  
```

The above summary showed the statistics for all the available variables. We explored the bed and bath variable using a bar plot and computing its skewness and kurtosis using following command.

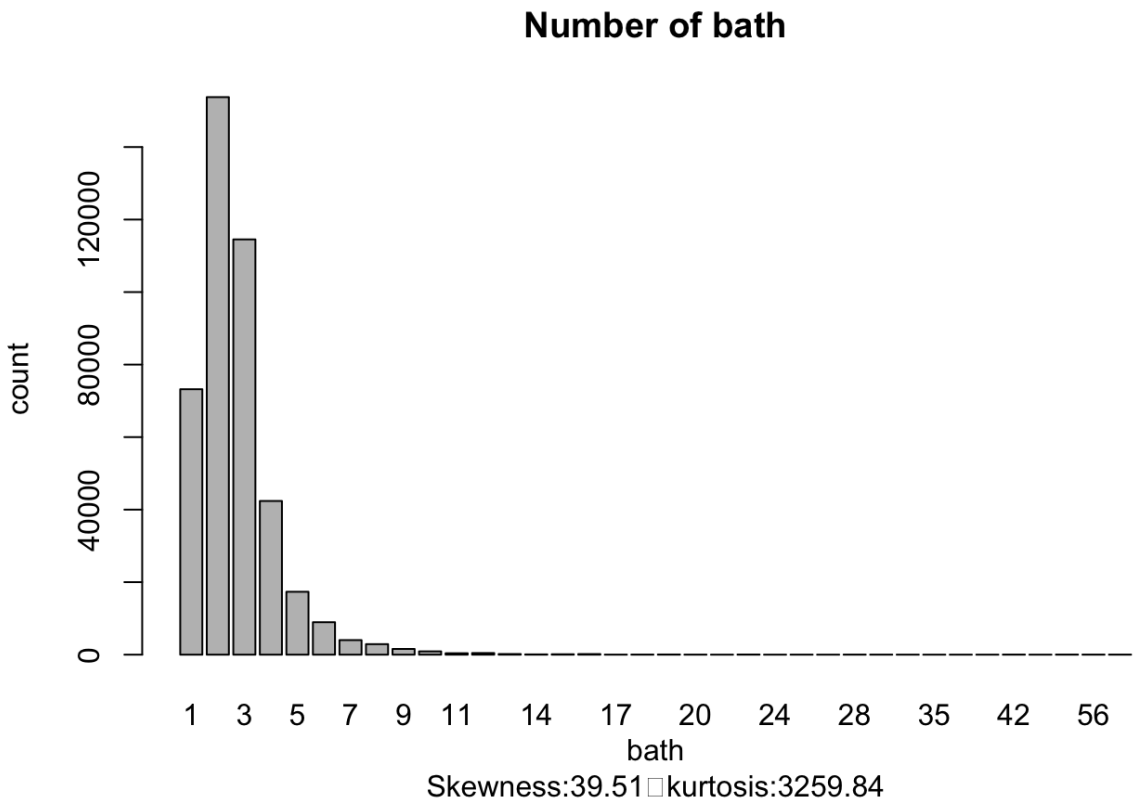
For bed

```
barplot(table(df$bed), main="Number of bed", xlab=paste0("bed", '\n',
'Skewness:', round(skewness(x=df$bed),2), '\t', 'kurtosis:',
round(kurtosis(df$bath),2)), ylab="count")
```



For bath

```
barplot(table(df$bath), main="Number of bath", xlab=paste0("bath",
'\n', 'Skewness:', round(skewness(x=df$bath),2), '\t', 'kurtosis:',
round(kurtosis(df$bath),2)), ylab="count")
```



Interpretation

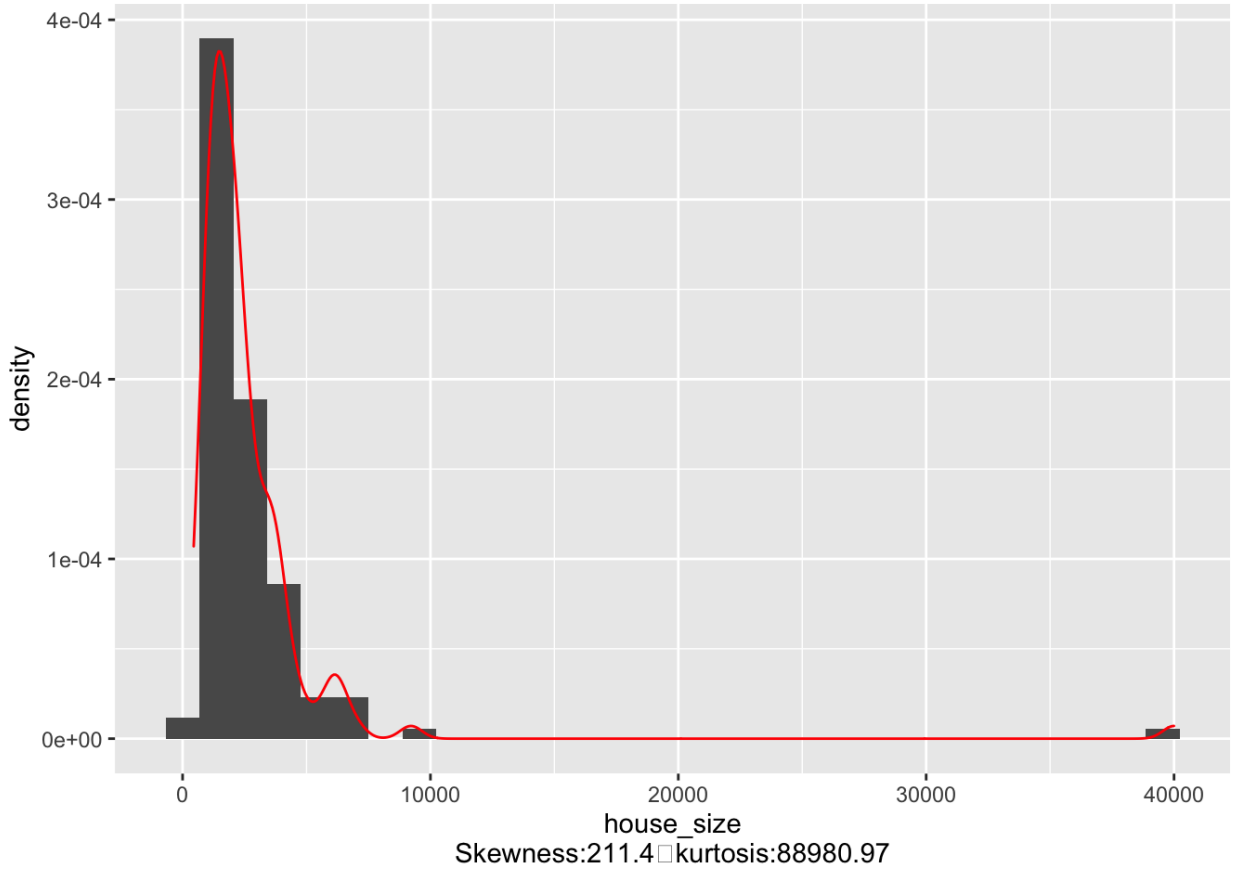
Skewness measures the asymmetry of a distribution around its mean and kurtosis measures how heavy the tails of the distribution are around its mean.

From the plot above, and the computed skewness and kurtosis values, it is clear that the bed and bath variables are right-skewed distributions. Their median values are less than their means. The positive kurtosis value indicates that the tail is heavier than the normal distribution which means this data has more outliers than a normal distribution. This can be true because the maximum number of beds and baths are 99 and 198 respectively, which means the price, acre_lot, and house_size need to be maximum in order to validate these numbers. We perform outlier detecting approaches and plot box plots for it.

Since these two variables are skewed, we explore the remaining variables and see their distribution using density plots. For better visualization, for other variables, we took a small random sample of 128 observations and drew the plots using that data. We also computed the skewness and kurtosis values.

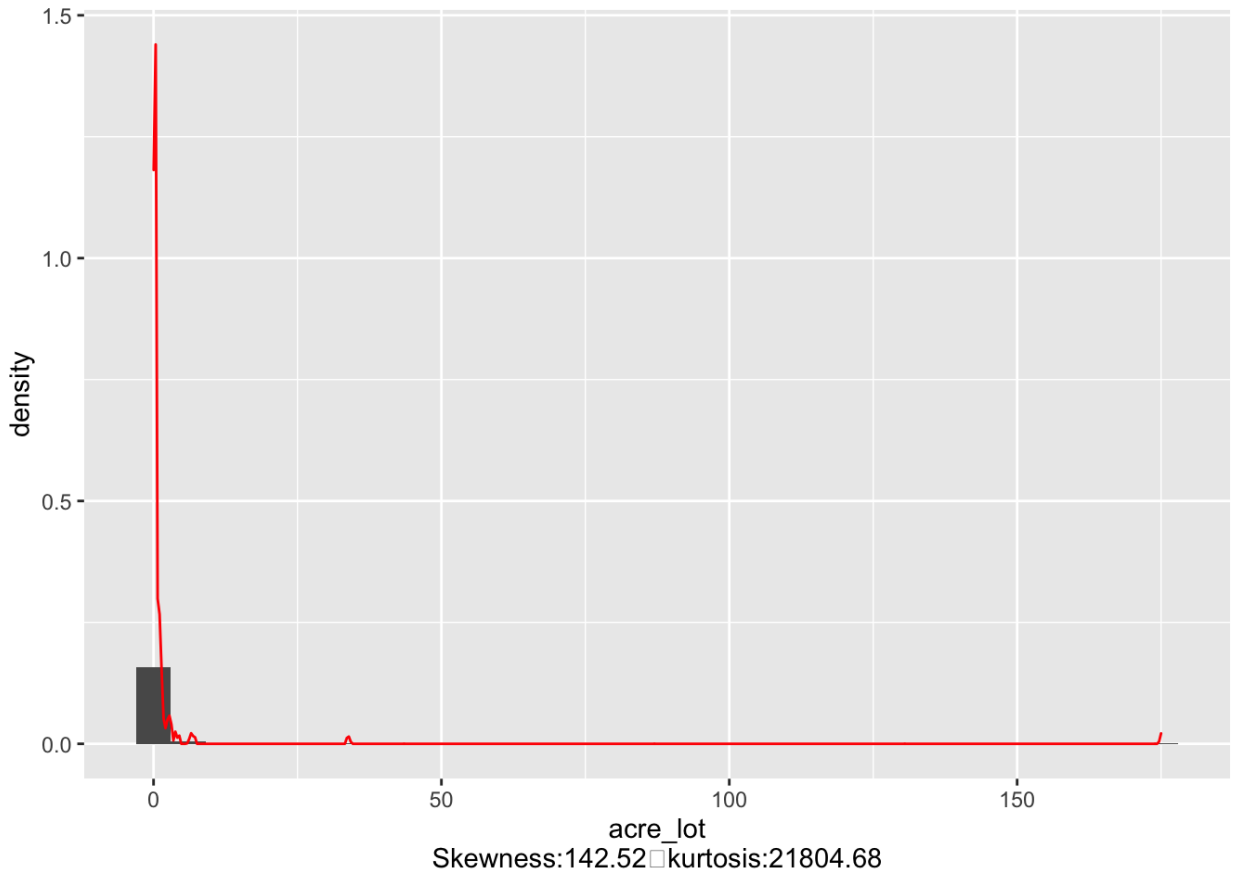
For house_size

```
dsample <- df[sample(nrow(df), 128), ]
ggplot(data=dsample, mapping=aes(x=house_size)) +
geom_histogram(aes(y=..density..), bins=30) +
geom_density(color="red")+labs(x=paste0("house_size", '\n',
'Skewness:', round(skewness(x=df$house_size),2), '\t', 'kurtosis:',
round(kurtosis(df$house_size),2)))
```



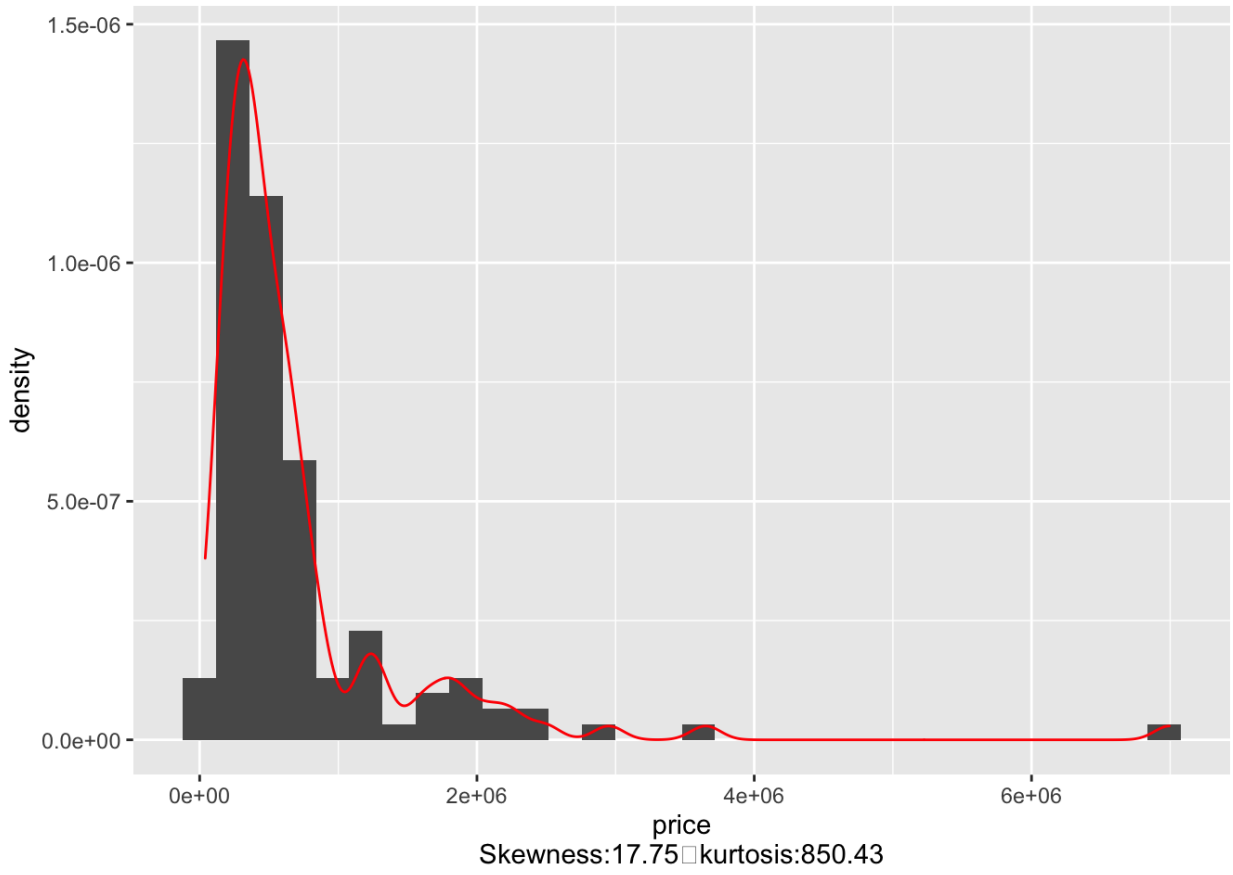
For acre_lot

```
ggplot(data=dsample, mapping=aes(x=acre_lot)) +
geom_histogram(aes(y=..density..), bins=30) +
geom_density(color="red")+labs(x=paste0("acre_lot", '\n',
'Skewness:', round(skewness(x=df$acre_lot),2), '\t', 'kurtosis:',
round(kurtosis(df$acre_lot),2)))
```



For price

```
ggplot(data=dsample, mapping=aes(x=price)) +
geom_histogram(aes(y=..density..), bins=30) +
geom_density(color="red")+labs(x=paste0("price", '\n', 'Skewness:',
round(skewness(x=df$price),2), '\t', 'kurtosis:',
round(kurtosis(df$price),2)))
```



From the above plots, it is clear that these variables price, acre_lot, house_size are right skewed and has potential outliers within them.

Outlier Detection

Since all the variables are right skewed, they contain outliers within them. Among different methods for outlier detection, we used the IQR criterion approach and plotted the box plot to see the outliers. For this experiment, we took bed and bath variables and checked their outliers.

IQR criterion

For bed

```
lowerOutlierLimit <- quantile(df$bed, probs=0.25,
names=FALSE)-1.5*IQR(df$bed)
upperOutlierLimit <- quantile(df$bed, probs=0.75,
names=FALSE)+1.5*IQR(df$bed)
```

```
bed_outliers<-df$bed[df$bed<lowerOutlierLimit |
df$bed>upperOutlierLimit]
length(bed_outliers)
unique(bed_outliers)
```

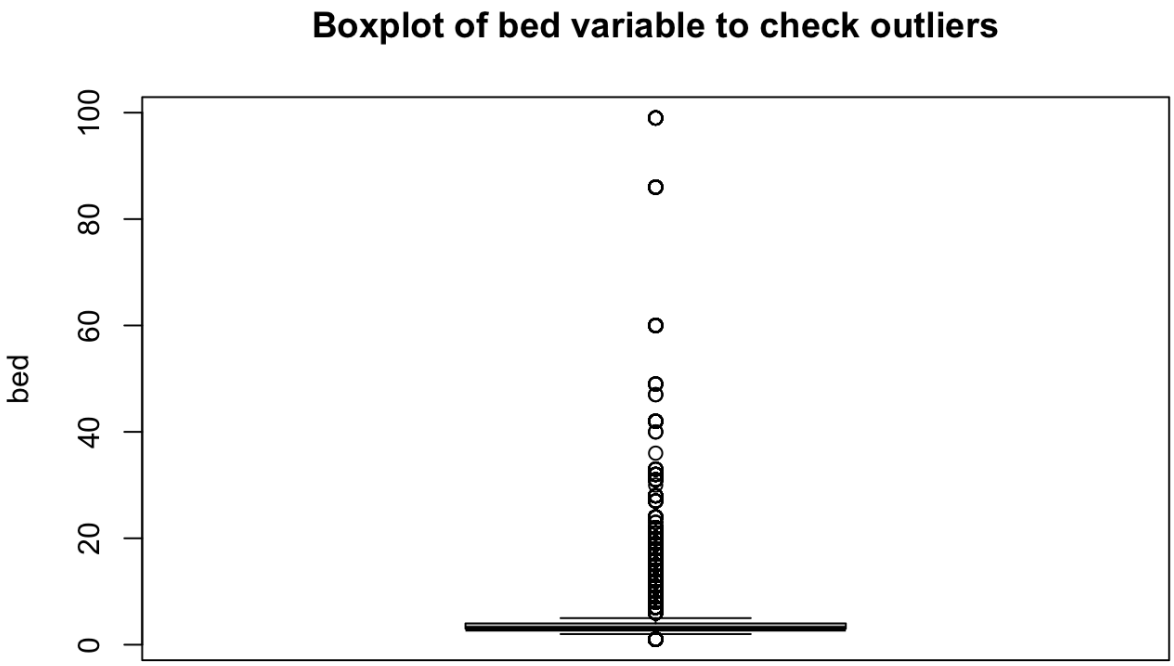
```
> [1] 61421
 [1]  6  1  9  7  8 12 13 10 11 33 24 28 14 18 20 16 15 19 17 40 21
86 31 27 42 60 22 32 99 49 30 23 47 36
```

Interpretation

There are 61421 potential outliers in bed variable and the unique list of potential outliers is listed above. The potential outliers are even more clear from the box plot below. These boxplots shows the minimum, maximum, median, 1st quartile, 3rd quartile and outliers contained in the data. All the points above the Q3 are outliers and it is perfectly depicted in the box plot.

```
boxplot(df$bed,
  ylab = "bed",
  main = "Boxplot of bed variable to check outliers"
)
```

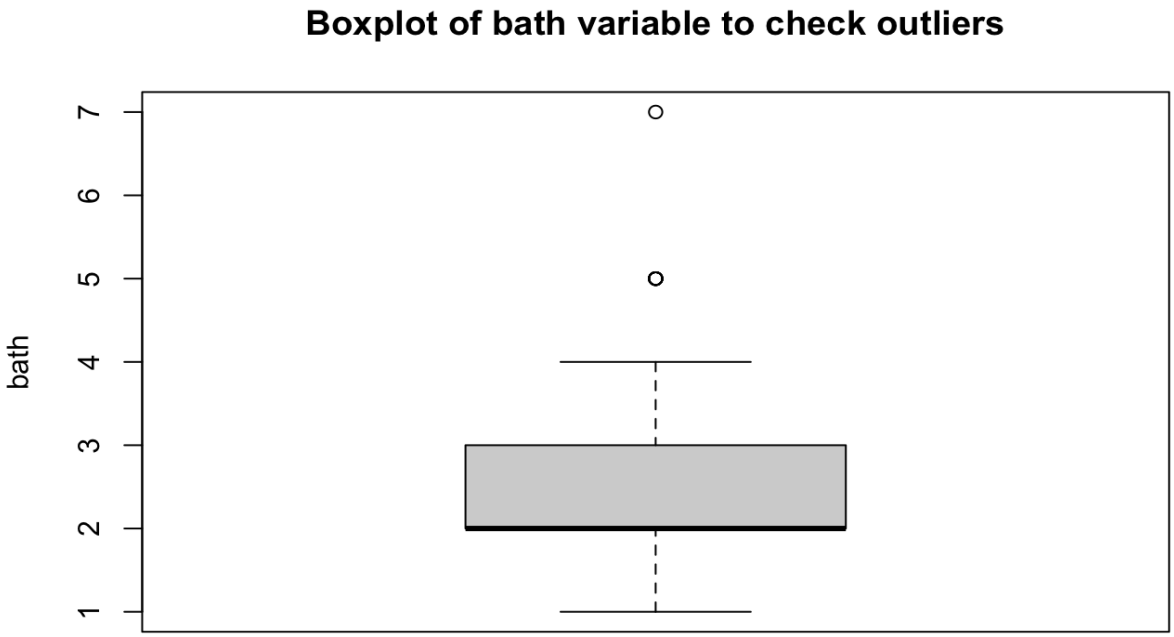
Fr



For bath

For the boxplot of bath variables, we used the 128 samples of our dataset for better readability of the image. The following box plot is even more clear as it properly shows the median, Q1, Q3, and outliers of the bath variables.

```
dsample <- df[sample(nrow(df), 128), ]
boxplot(dsample$bath,
  ylab = "bath",
  main = "Boxplot of bath variable to check outliers"
)
```

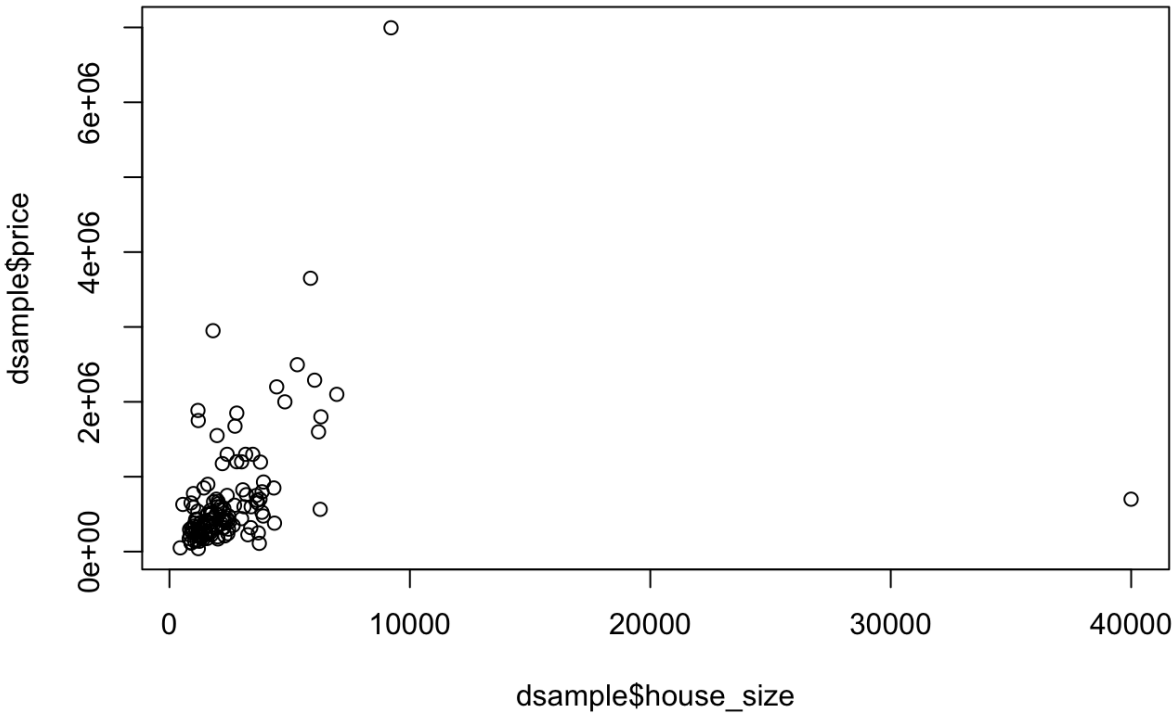


Bivariate Analysis

In this analysis, we checked the factors affecting prices and analyzed the relationship of other variables with price variable. Covariance, correlation, and chi-square test are the common approaches/measures for bivariate data analysis. Covariance measures how two variables vary together, i.e if higher values of one variable are associated with the higher or lower values of the other variable. A positive covariance means both variables get larger and smaller together and vice versa. Correlation measures the strength and direction of a linear relationship between two variables. A value close to 1 indicates a very strong positive correlation and a value close to -1 means a strong negative correlation. And a value close to 0 indicates a lack of correlation between the two variables.

We started by plotting a scatterplot to see the relationship between house_size and price variable along with their covariance, and correlation and performed a chi-square test.

```
plot(dsample$house_size, dsample$price)
```




```
cov(df$house_size, df$price)
cor(df$house_size, df$price)
```

```
> [1] covariance:  1366950579
> [1] correlation:  0.2770014
```

```
chisq.test(df$house_size, df$price)
> Pearson's Chi-squared test

data:  df$house_size and df$price
X-squared = 321627695, df = 26595132, p-value < 2.2e-16
```

Interpretation

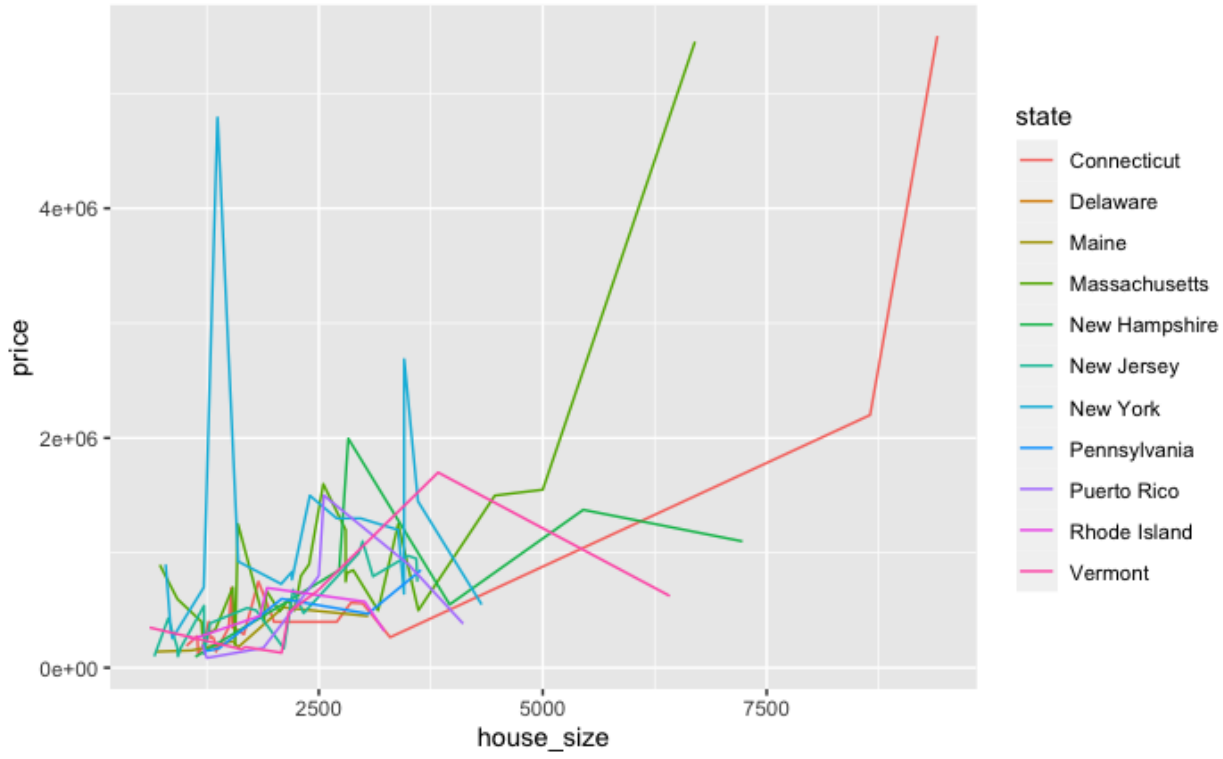
We can see that the covariance is positive and higher value, which means if the house size

increases, house price increases. Similarly, the correlation value is greater than 0, but not too close to 1, which means house size and price are somewhat correlated. Since the p-value is so smaller than 0.05, we have some evidence to reject the null hypothesis and assume that there is a relation between variables house size and price, and the relationship has been explained by the covariance value.

Multivariate Analysis

Here, we explored the relationship between bed, bath, acre_lot, and price variables by creating multiple plots. For this exploration, we used a small data sample, as the plot is more readable. We started by showing the relationship between house size and its price for each state of the dataset.

```
ggplot(data=dsample, mapping=aes(x=house_size, y=price, color=state))
+
geom_line()
```

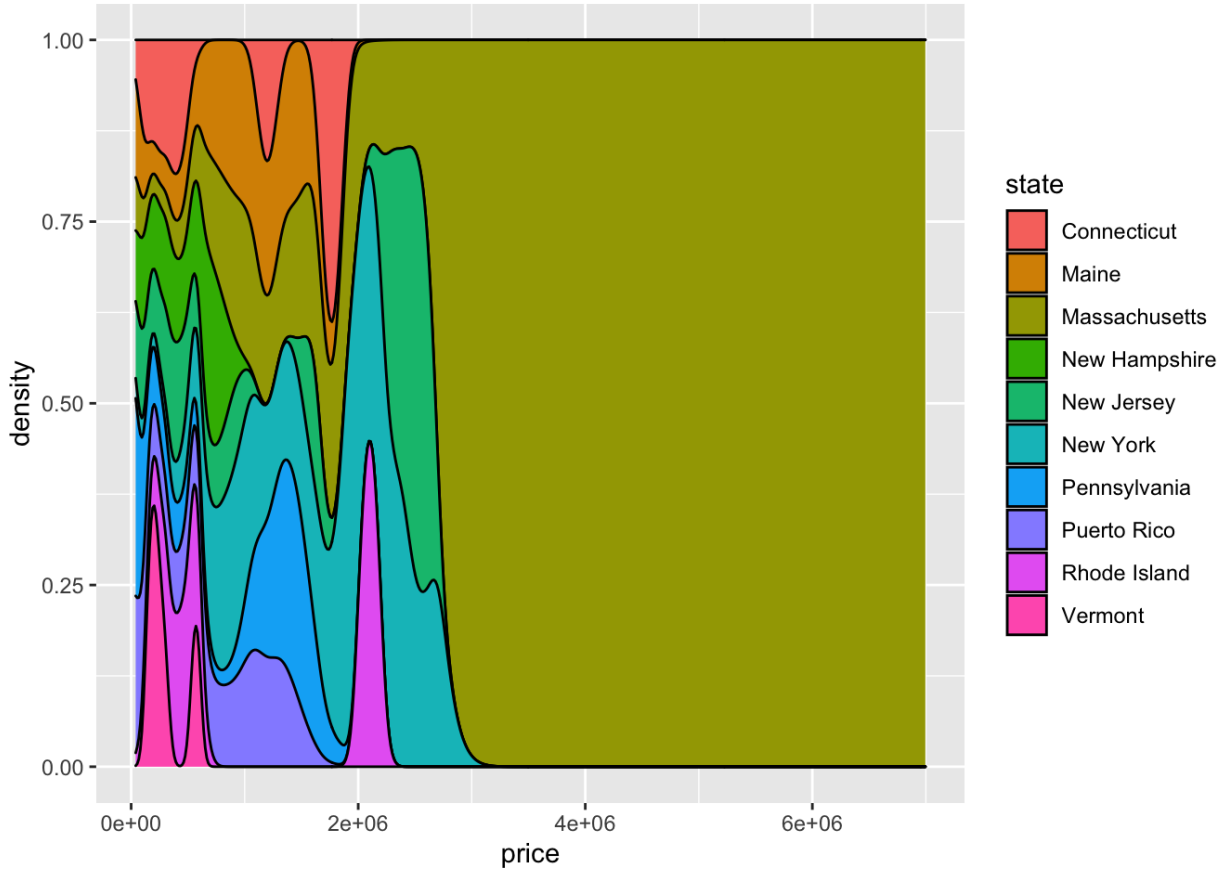


Interpretation

The states like Massachusetts, New Hampshire which have the highest price and include more number of houses within the states. Similarly, the price of houses increases as the house size increase, which verifies the previous results.

The following density plot shows the relationship between price and state.

```
ggplot(data=dsample, mapping=aes(x=price, fill=state)) +
geom_density(position="fill")
```

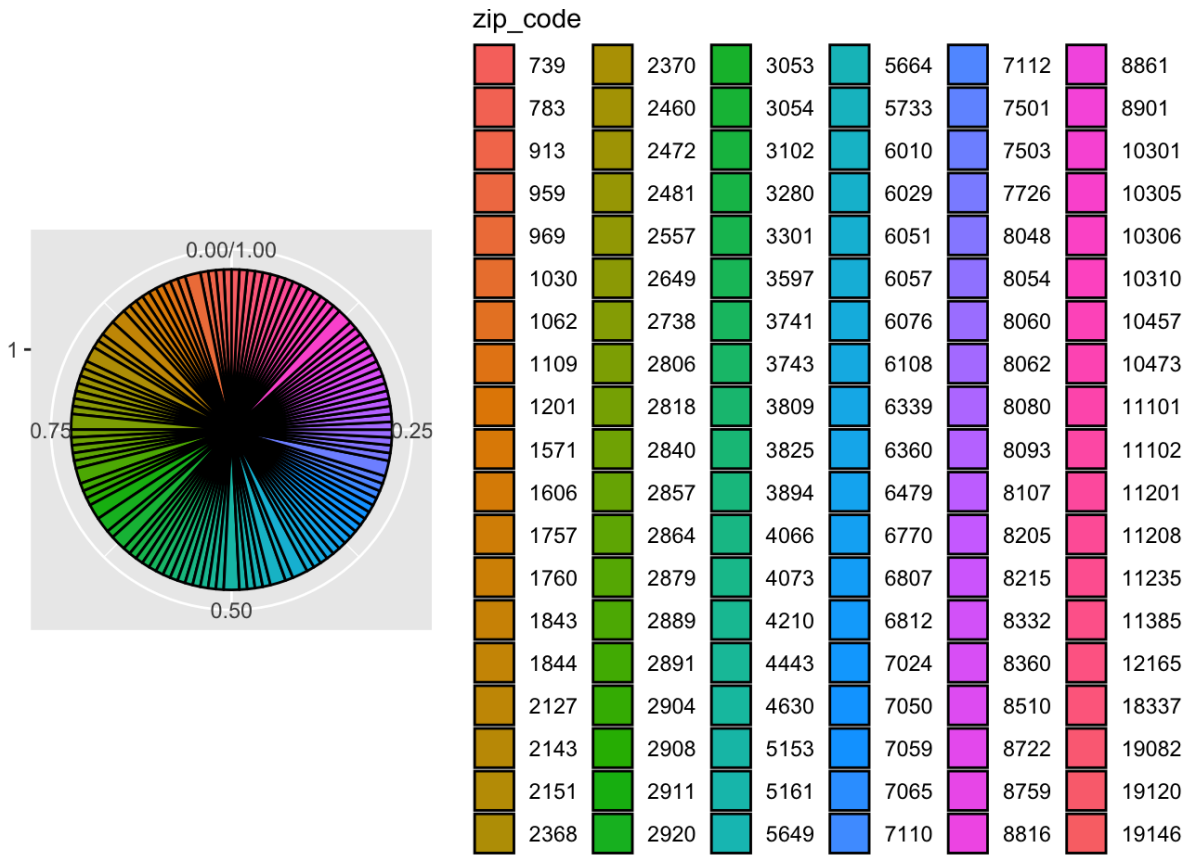


Interpretation

The states like Massachusetts, New Hampshire covered a lot area and looked dense, which is correct as a lot of real estate listed are from these states.

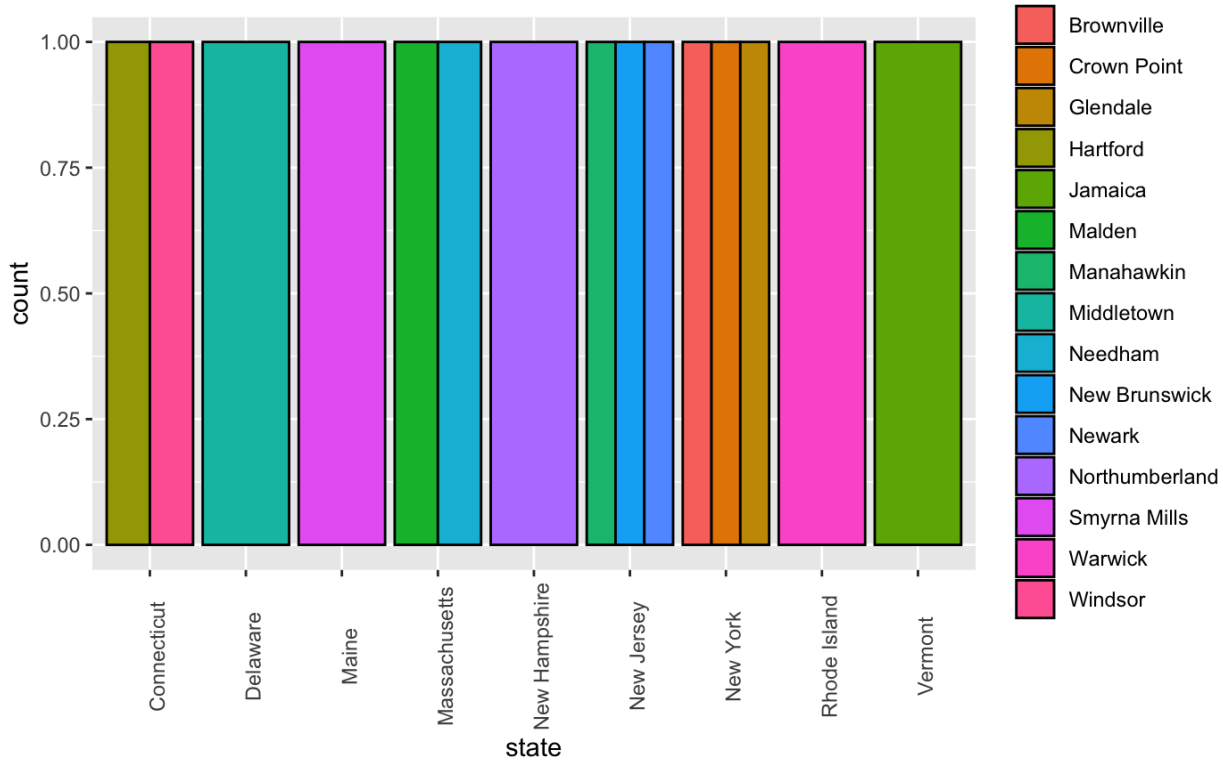
The following pie plot shows the distribution of data samples on the basis of its zip_code.

```
ggplot(data=dsample) + geom_bar(mapping=aes(x=factor(1),
fill=zip_code), width=1,
    position="fill", color="black") + coord_polar(theta="y") +
scale_y_continuous(
    name="") + scale_x_discrete(name="")
```



The following plot shows the distribution of the number of cities in each state. This shows that the number of cities is more in the states like Massachusetts, New Jersey, New York and less in Delaware, Vermont, resulting less number of houses for listing.

```
ggplot(data=dsample[1:15,], mapping=aes(x=state, fill=city)) +
geom_bar(color="black", position="dodge")
```



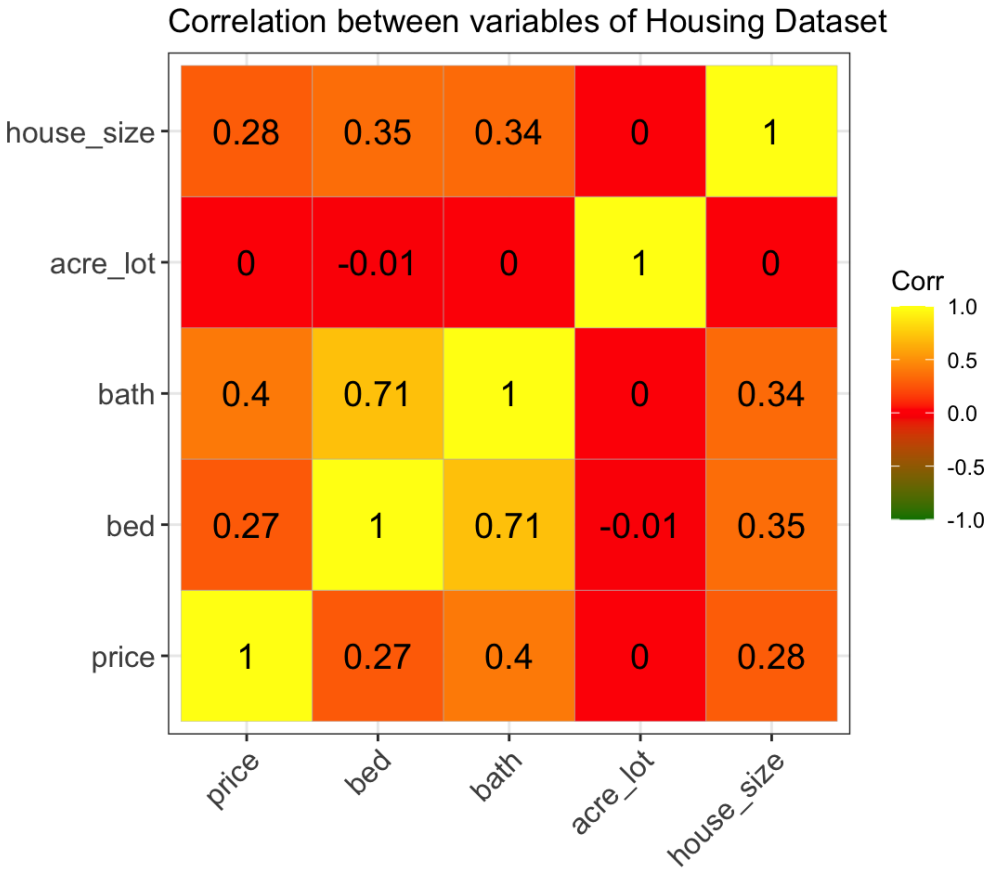
Correlation

Finally, we computed the overall correlation between all the numeric variables and plotted the values and analyzed the result. A subset of the dataset is created which included only the numeric variables and used the cor command to find the correlation.

```
                price          bed       bath      acre_lot   house_size
price      1.0000000000  0.274215330  0.3989525  0.0004017237  0.277001388
bed        0.2742153299  1.000000000  0.7106327 -0.0051144208  0.345525151
bath       0.3989524971  0.710632667  1.0000000 -0.0017237000  0.341363458
acre_lot   0.0004017237 -0.005114421 -0.0017237  1.0000000000 -0.001015233
house_size 0.2770013885  0.345525151  0.3413635 -0.0010152331  1.000000000
```

We used the ggcorplot library to plot all these pairwise correlation of our housing dataset.

```
ggcorrplot(round(cor(df_new),4),
           type = "full",
           lab = TRUE,
           lab_size = 5,
           colors = c("#008000", "#ff0001", "#ffff10"),
           title="Correlation between variables of Housing Dataset",
           ggtheme=theme_bw)
```



Interpretation

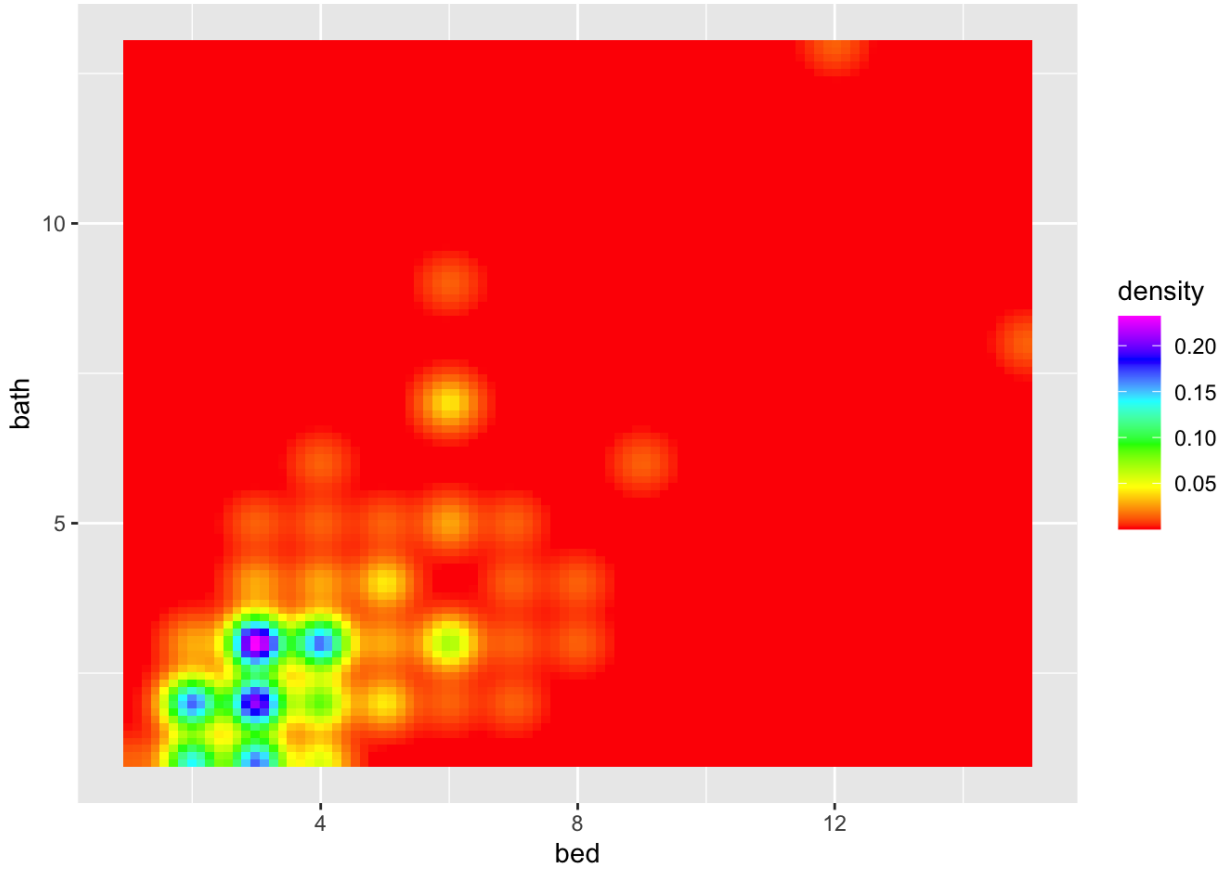
From this correlation plot, we can see that there is a high correlation between the number of beds and bath in a house. Variable acre_lot has a very low correlation which is almost 0. There might be various reasons for this. One of the reasons might be the range of acre_lot, which is very low and below all the other variables. Normalization can be done to make the scale of this variable similar to other variables. Another possible reason is there might be some non-linear relationship with this variable, and since correlation measures, linear association between two given variables and cannot measure non-linear relation.

The other interesting thing is the correlation between bed and acre_lot is negative. A negative or inverse correlation between two variables indicates that one variable increases while the other decreases. Since there is not much description of what acre_lot means, as per the correlation value obtained it seems that as the number of beds increases, the acre_lot decreases. It can be interpreted as the size of empty land decreases as we create more beds for the house.

Besides those other variables like bed, bath, and house_size has a positive correlation with price, which means as these variables increase price also increases.

Since we saw that bath and bed are highly correlated as compared to other variables, we plotted the joint density of bed and bath using stat_density2d. stat_density2d is used to estimate the joint density of two variables.

```
ggplot(data=dsample, mapping=aes(x=bed, y=bath))  +
stat_density2d(geom="tile", contour=FALSE, aes(fill=..density..)) +
scale_fill_gradientn(colours = rainbow(6))
```



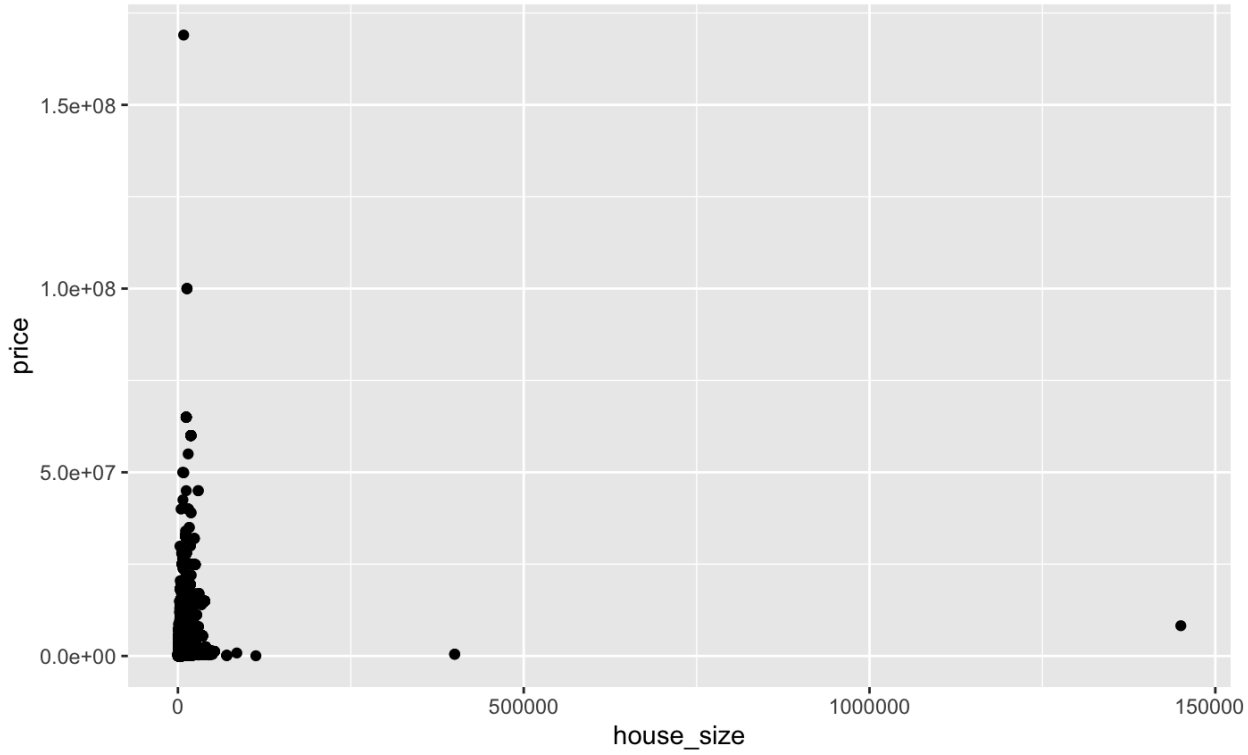
Interpretation

From the above heatmap, we can see that the density is quite high in the left bottom corner, showing the correlation between bath and bed, which shows most of the houses have a fairly equal number of bed and bath.

Cluster Analysis

This analysis helps to group the common observations together. For this analysis, we used house_size and its corresponding price values from our dataset.

```
d <- c("house_size", "price")
mydata <- df[d]
ggplot(data=mydata, mapping=aes(x=house_size, y=price)) +
geom_point()
```



Now, we used the kmeans algorithm to cluster these dataset into 5 different clusters.

```
set.seed(1001)
fit <- kmeans(x=mydata, centers=5)
fit
mydata$cluster <- factor(fit$cluster)
```

```
K-means clustering with 5 clusters of sizes 54078, 1885, 9033, 140,
356091

Cluster means:
  house_size      price
1   3705.335  1740675.3
2   9213.191 13768682.7
3   6469.938  5087778.6
```

```
4  13170.943 41456378.6
5   2072.857   445314.6
```

```
head(mydata)
```

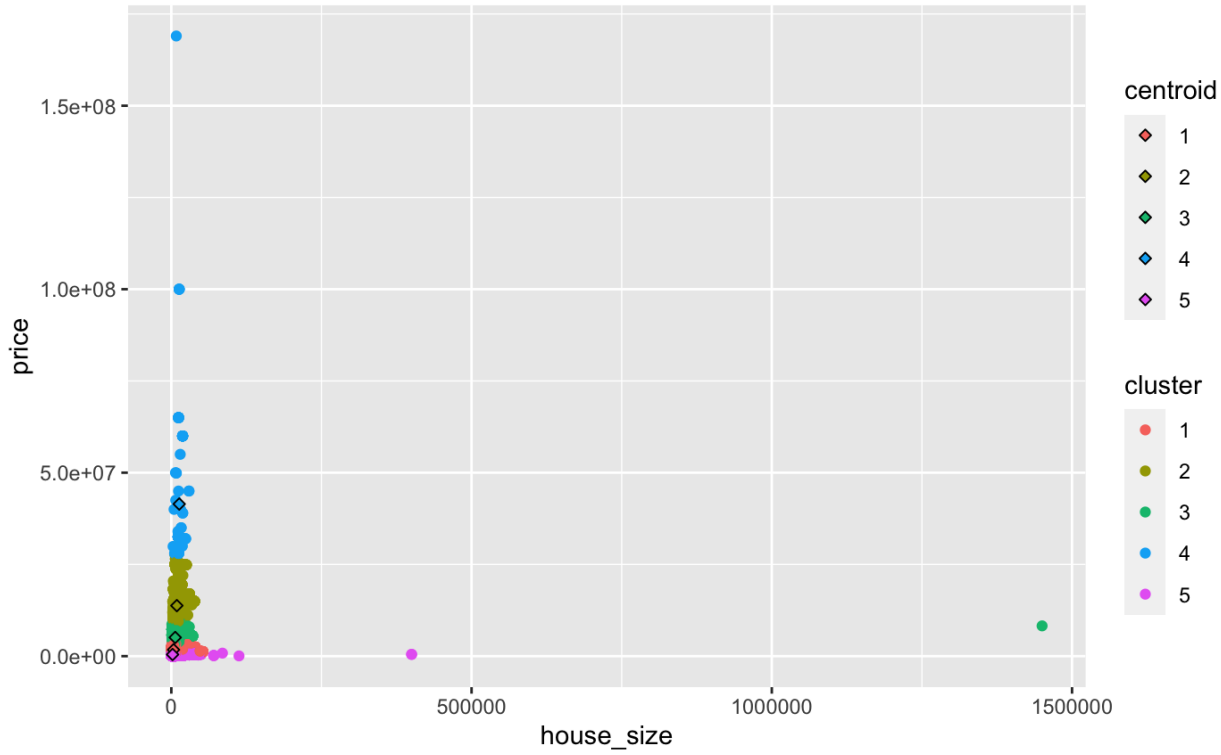
A tibble: 6 × 3

house_size <dbl>	price <dbl>	cluster <fctr>
920	105000	5
1527	80000	5
748	67000	5
1800	145000	5
2520	179000	5
2040	50000	5

6 rows

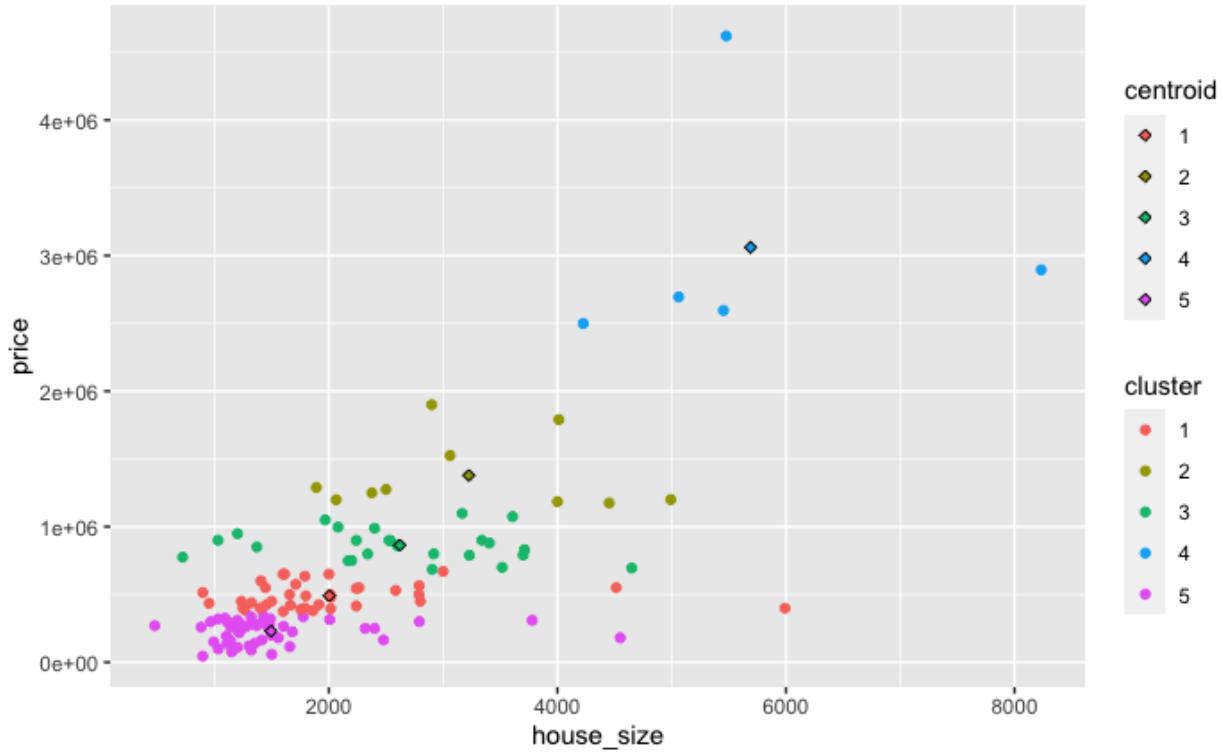
Plotting the extracted clusters in a scatterplot, we got the following plot.

```
ggplot() + geom_point(data=mydata, mapping=aes(x=house_size, y=price,
color=cluster)) + geom_point(data=data.frame(fit$centers,
centroid=as.factor(1:nrow(fit$centers))),
    mapping=aes(x=house_size, y=price, fill=centroid), shape=23,
color="black")
```



From the scatterplot above, we can see five different clusters group together for house_size and price. Since we used all 400k observations they looked as if they are dense in a particular range. So, we used the small sample of the dataset and plotted the cluster in scatterplot.

```
set.seed(1001)
dsample <- mydata[sample(nrow(mydata), 128), ]
fit <- kmeans(x=dsample, centers=5)
dsample$cluster <- factor(fit$cluster)
ggplot() + geom_point(data=dsample, mapping=aes(x=house_size,
y=price, color=cluster)) + geom_point(data=data.frame(fit$centers,
centroid=as.factor(1:nrow(fit$centers))),
    mapping=aes(x=house_size, y=price, fill=centroid), shape=23,
color="black")
```



This plot showed the clusters more clearly as compared to previous one. The cluster formation also showed that as the house size increases, the price increased and the expensive houses are clustered in a cluster whereas the cheap one are in another cluster.

Summary

The following are our findings from the data analysis of our housing dataset.

1. The initial expectation from this exploration was that all the variables have positive correlation with the price, which means as the size, bed, bath variables increases, the price factors will also increases. But, we found that the acre_lot has zero correlation with price, which shows that there might be some non-linear relationship between price and acre_lot. This non-linear relationship has not been explored in this analysis.
2. Another interesting correlation is between acre_lot and bed variables. If we consider acre_lot as the overall area of the land or plot of the house, both acre_lot and bed variables should have positive correlation. But, they have negative correlation, which might mean that acre_lot mean the remaining empty land area of the house. Since there is not much detail about these variables, we are not sure why they have negative correlation.
3. Beside these, all the other correlation are as we have expected. Bed and bath variables are highly correlated.
4. This dataset has a lot of outliers in each variables. Even after removing the missing values and less significant columns, there are huge number of outliers in each columns.

5. The states which has higher number of house listing also has many cities enlisted in the dataset. States with less number of real estate listing has only one or two cities in the dataset.
6. The chisquare test showed that the price variable is dependent on other features of the dataset. So for predicting the prices we can implement these features as independent variable and leverage their dependency with price to predict future prices of the real estate.
7. Cluster analysis shows the groups of houses on the basis of their price and house size.

Conclusion

This analysis of the housing dataset has helped to understand data analysis and visualization with R in an efficient way. Further improvement for this work can be done by implementing some predictive algorithm like logistic, linear regression to predict the prices and other clustering algorithms to understand the relationship with variables.