Quantitative Data Analysis

2246041

09 January, 2023

0. Instructions

```
# Add code here to load any required libraries with `library()`.
# We suggest you use `install.package()` for any required packages externally to this document
# since installation only need be done once.
#install.packages("ggplot2")
#install.packages("ggrepel")
#install.packages("tidyverse")
#install.packages("validate")
#install.packages('caret')

library(ggplot2) # for plotting
library(ggrepel) # for plot labels in pie chart
library(tidyverse) # for mutating data
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.5.0
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ✔ purrr   1.0.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(modeest) # for calculating mode
```

```
## Registered S3 method overwritten by 'rmutil':
##   method         from
##   print.response httr
```

```
library(validate) # for validator object
```

```
##
## Attaching package: 'validate'
##
## The following object is masked from 'package:dplyr':
##
##     expr
##
## The following object is masked from 'package:ggplot2':
##
##     expr
```

```
library(car) # for QQ plot method and vif
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(moments) # for skewness test
```

```
##
## Attaching package: 'moments'
##
## The following object is masked from 'package:modeest':
##
##     skewness
```

```
library(caret) # for confusion matrix
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

1. Organise and clean the data

1.1 Subset the data into the specific dataset allocated

```
# Assign your student id into the variable SID, for example:
SID <- 2246041                  # This is an example, replace 2101234 with your actual ID
SIDoffset <- (SID %% 100) + 1    # Your SID mod 100 + 1

load("house-analysis.RDa")
# Now subset the housing data set
# Pick every 100th observation starting from your offset
# Put into your data frame named mydf (you can rename it) -> renamed to housedf
housedf <- house.analysis[seq(from=SIDoffset,to=nrow(house.analysis),by=100),]
```

1.2 Data quality analysis

To check for data quality, we would follow the following steps:

Check the data against the provided metadata:

1. We will check the dimensions (rows and columns) of the data with *dim()*.
2. Next, we'll check variable names against metadata with *names()*.
3. Additionally, we'll check variable data type with *str()*
4. Furthermore, we'll briefly look at the data summary with *summary()*.
5. Finally, we'll check number of unique values in the column using *unique()* method. Intuitively, this gives us an idea about the categorical or numerical nature of variables.

Eyeball the data

We'll use *head*() or *View*() to get first glance at the data. This will also help us understand the variables and set up rules for the *validator* object.

Check for completeness of the data

We will count the rows with missing values, NAs, and blanks using *is.na*() and *colSums*(). This step is additional as *summary*() already shows NA values if any.

Check for Unusual / Implausible Values in individual columns

1. For variables with numerical values like *mq*, *floor*, *n_rooms*, etc., we'll use conditional operators in the *validator* object to check for negative or 0 values.
2. For variables having a fixed set of values, we'll check if their value belong to the listed factors using *validator* object.
3. During individual analysis, we'll check for extreme values such as extreme mq or unlikely number of rooms through variable summary and graphical visualizations.
4. We'll use the *summary()* and *table()* functions to check for discrepancies in the data by generating a summary for the numerical and frequency table for categorical variables, respectively.
5. Graphically, we'll visualize the data and look for any quality issues through the use of *boxplot* and *histogram* for numerical variables, and *barplot* and *piechart* (see [@r-charts] for adding labels outside chart) for categorical variables.

```
# Custom function for plotting used towards data quality analysis

#Defining functions for plotting via ggplot

#Custom Pie chart using ggplot
custom_pie <- function(var,  title="", xlab, ylab){
  if(title==""){
    title=paste("Pie Chart for",xlab)
  }

  df <- as.data.frame(table(var))
  df2 <- df %>%
    mutate(csum = rev(cumsum(rev(Freq))),
           pos = Freq/2 + lead(csum, 1),
           pos = if_else(is.na(pos), Freq/2, pos))
  ggplot(df, aes(x = "" , y = Freq, fill = var)) + geom_col(width = 1, color=alpha("white",alpha = 0.3)) +
  coord_polar(theta = "y") +
  geom_label_repel(data = df2,
                   aes(y = pos, label = paste0(round(100*Freq/sum(Freq),2), "%")),
                   size = 3, nudge_x = 1, show.legend = FALSE, color=alpha("#DDDDDD", alpha = 1)) +
  guides(fill = guide_legend(title = xlab)) + ggtitle(title) +
  theme_void()
}

#Custom Bar chart using ggplot
custom_bar <- function(var, title="", xlab, ylab){
  if(title==""){
    title=paste("Bar Graph for",xlab)
  }
  df <- as.data.frame(table(var))
  ggplot(df, aes(x=var, y=Freq, fill=var)) +
    geom_bar(width = 1, stat = "identity", color=alpha("white",alpha = 0.3))  +
    guides(fill=guide_legend(title=xlab)) +
    ggtitle(title) + xlab(xlab) + ylab(ylab)  +
    theme_minimal() +
    geom_text(aes(x = var, y = Freq + max(Freq)/50, label = Freq), size=3)
}

#Custom Box Plot using ggplot
custom_box <- function(var, title="", fill="#bfe9ff", xlab, ylab){
  if(title==""){
    title=paste("Box Plot for",xlab)
  }
  ggplot(data = housedf, aes(x="",y=var)) +
    geom_boxplot(fill=fill) +
    ggtitle(title) + xlab(xlab) + ylab(ylab)  +
    theme_minimal()
}

#Custom Histogram using ggplot
custom_hist <- function(data, var, title="", fill="#ff6e7f", xlab, ylab, binwidth=1){
  if(title=="") title=paste("Histogram for",xlab,"vs",ylab)
  ggplot(data = data, aes(x=var)) +
    geom_histogram(binwidth=binwidth, fill=fill,color=alpha("white",alpha = 0.3)) +
    ggtitle(title) + xlab(xlab) + ylab(ylab)  +
    theme_minimal() +
    stat_bin(aes(y=..count.., label=..count..), geom="text", vjust=-.5, binwidth = binwidth, size= 3)
}
```

```
# Checking dimensions of the data
dim(housedf)
```

```
## [1] 904  12
```

```
# Getting the variable names
names(housedf)
```

```
##  [1] "id"                   "price"                "mq"
##  [4] "floor"                "n_rooms"              "n_bathrooms"
##  [7] "has_terrace"          "has_alarm"            "heating"
## [10] "has_air_conditioning" "has_parking"          "is_furnished"
```

```
# Getting the structure and checking the data type
str(housedf)
```

```
## 'data.frame':    904 obs. of  12 variables:
##  $ id                  : int  84 274 477 669 943 1154 1368 1676 1970 2236 ...
##  $ price               : num  135000 38000 45000 189000 40000 38000 115000 12000 127000 44000 ...
##  $ mq                  : num  150 80 110 120 45 90 87 80 55 66 ...
##  $ floor               : num  1 2 1 1 1 1 2 1 2 1 ...
##  $ n_rooms             : num  3 5 4 4 2 4 3 3 3 2 ...
##  $ n_bathrooms         : num  1 2 1 2 1 2 2 2 2 1 ...
##  $ has_terrace         : int  0 0 0 0 0 0 1 0 1 0 ...
##  $ has_alarm           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ heating             : chr  "autonomous" "autonomous" "autonomous" "autonomous" ...
##  $ has_air_conditioning: int  1 0 0 0 1 0 0 0 1 0 ...
##  $ has_parking         : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ is_furnished        : int  1 0 0 0 0 0 0 0 0 0 ...
```

```
# Displaying the numerical summary of the dataframe
summary(housedf)
```

```
##        id              price              mq             floor       
##  Min.   :    84   Min.   :  1000   Min.   :  0.00   Min.   :1.000  
##  1st Qu.: 53956   1st Qu.: 79000   1st Qu.: 77.75   1st Qu.:1.000  
##  Median :113201   Median :125000   Median :100.00   Median :1.000  
##  Mean   :111106   Mean   :144084   Mean   :115.78   Mean   :1.762  
##  3rd Qu.:167356   3rd Qu.:187250   3rd Qu.:135.00   3rd Qu.:2.000  
##  Max.   :223309   Max.   :500000   Max.   :840.00   Max.   :7.000  
##     n_rooms        n_bathrooms     has_terrace       has_alarm       
##  Min.   :-1.000   Min.   :1.000   Min.   :0.0000   Min.   :0.00000  
##  1st Qu.: 3.000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.00000  
##  Median : 3.000   Median :1.000   Median :0.0000   Median :0.00000  
##  Mean   : 3.485   Mean   :1.425   Mean   :0.1261   Mean   :0.01106  
##  3rd Qu.: 4.000   3rd Qu.:2.000   3rd Qu.:0.0000   3rd Qu.:0.00000  
##  Max.   : 5.000   Max.   :3.000   Max.   :1.0000   Max.   :1.00000  
##    heating          has_air_conditioning  has_parking      is_furnished     
##  Length:904         Min.   :0.0000       Min.   :0.0000   Min.   :0.00000  
##  Class :character   1st Qu.:0.0000       1st Qu.:0.0000   1st Qu.:0.00000  
##  Mode  :character   Median :0.0000       Median :0.0000   Median :0.00000  
##                     Mean   :0.3208       Mean   :0.0177   Mean   :0.07965  
##                     3rd Qu.:1.0000       3rd Qu.:0.0000   3rd Qu.:0.00000  
##                     Max.   :1.0000       Max.   :1.0000   Max.   :1.00000
```

```
# Eyeballing the data
#View(housedf)
head(housedf)
```

	id	price	mq	floor	n_rooms	n_bathrooms	has_terrace	has_alarm	heating
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>	<chr>
84	84	135000	150	1	3	1	0	0	autonomous
274	274	38000	80	2	5	2	0	0	autonomous
477	477	45000	110	1	4	1	0	0	autonomous
669	669	189000	120	1	4	2	0	0	autonomous
943	943	40000	45	1	2	1	0	0	autonomous
1154	1154	38000	90	1	4	2	0	0	autonomous

6 rows | 1-10 of 13 columns

```
# Get no of Unique values in every column
# Loop through all columns in the dataset
for(col in names(housedf)) {
  # Get the unique values in the column
  unique_values <- unique(housedf[[col]])
  # Get no of unique values
  length_unique <- length(unique_values)

  # If no of unique values is less, e.g. 20 supposedly
  if(length_unique < 20) {
    # Print the length and value of unique values if unique values are limited
    print(paste(col, "has", length_unique,"unique values; Repeated values are ", toString(unique_values)))
    }

  else{
    # Print the length of unique values in the column if unique values are not limited
    print(paste(col, "has", length_unique,"unique values;"))
  }
}
```

```
## [1] "id has 904 unique values;"
## [1] "price has 230 unique values;"
## [1] "mq has 180 unique values;"
## [1] "floor has 7 unique values; Repeated values are  1, 2, 3, 4, 5, 7, 6"
## [1] "n_rooms has 5 unique values; Repeated values are  3, 5, 4, 2, -1"
## [1] "n_bathrooms has 3 unique values; Repeated values are  1, 2, 3"
## [1] "has_terrace has 2 unique values; Repeated values are  0, 1"
## [1] "has_alarm has 2 unique values; Repeated values are  0, 1"
## [1] "heating has 3 unique values; Repeated values are  autonomous, other, autonamous"
## [1] "has_air_conditioning has 2 unique values; Repeated values are  1, 0"
## [1] "has_parking has 2 unique values; Repeated values are  0, 1"
## [1] "is_furnished has 2 unique values; Repeated values are  1, 0"
```

```
# From metadata, uniqueness check above, and data summary,
# we can deduce the following:
# id: unique, numeric and non-negative
# price: numeric and non-negative
# mq: numeric and non-negative
# floor: categorical, 1 to 7
# n_rooms: categorical, -1, 2, 3, 4, 5
# n_bathrooms: categorical, 1, 2, 3
# has_terrace: binary, either 0 or 1
# has_alarm: binary, either 0 or 1
# heating: categorical, (either "autonomous" or "other" from the metadata)
# has_air_conditioning: binary, either 0 or 1
# has_parking: binary, either 0 or 1
# is_furnished: binary, either 0 or 1

# Data Quality Check using Validator
# the rules are built based on outputs from summary() and View()
house.rules <- validator(uniqId = is_unique(id),
                         posPrice = price > 0,
                         posMq = mq > 0,
                         posFloor = floor > 0,
                         posRooms = n_rooms > 0,
                         posBath = n_bathrooms > 0,
                         okTerc = is.element(has_terrace, c(0,1)),
                         okAlrm = is.element(has_alarm, c(0,1)),
                         okHeat = is.element(heating, c("autonomous","other")),
                         okAC = is.element(has_air_conditioning, c(0,1)),
                         okPrk = is.element(has_parking, c(0,1)),
                         okFur = is.element(is_furnished, c(0,1))
                         )
housedf.qual.chk <- confront(housedf, house.rules)
summary(housedf.qual.chk)
```

name	items	passes	fails	n...	error	warning	expression
<chr>	<int>	<int>	<int>	<int>	<lgl>	<lgl>	<chr>
uniqId	904	904	0	0	FALSE	FALSE	is_unique(id)
posPrice	904	904	0	0	FALSE	FALSE	price > 0
posMq	904	903	1	0	FALSE	FALSE	mq > 0
posFloor	904	904	0	0	FALSE	FALSE	floor > 0
posRooms	904	903	1	0	FALSE	FALSE	n_rooms > 0
posBath	904	904	0	0	FALSE	FALSE	n_bathrooms > 0

okTerc	904	904	0	0	FALSE	FALSE	is.element(has_terrace, c(0, 1))
okAlrm	904	904	0	0	FALSE	FALSE	is.element(has_alarm, c(0, 1))
okHeat	904	903	1	0	FALSE	FALSE	is.element(heating, c("autonomous", "other"))
okAC	904	904	0	0	FALSE	FALSE	is.element(has_air_conditioning, c(0, 1))

1-10 of 12 rows Previous **1** 2 Next

```
# Counting NA values
colSums(is.na(housedf))
```

```
##                   id                price                   mq
##                    0                    0                    0
##                floor              n_rooms          n_bathrooms
##                    0                    0                    0
##          has_terrace            has_alarm              heating
##                    0                    0                    0
## has_air_conditioning          has_parking         is_furnished
##                    0                    0                    0
```

```
# Analysing id variable via numerical summary and checking if id is unique
summary(housedf$id)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      84   53956  113201  111106  167356  223309
```

```
nrow(housedf)==sum(housedf$id==unique(housedf$id))
```

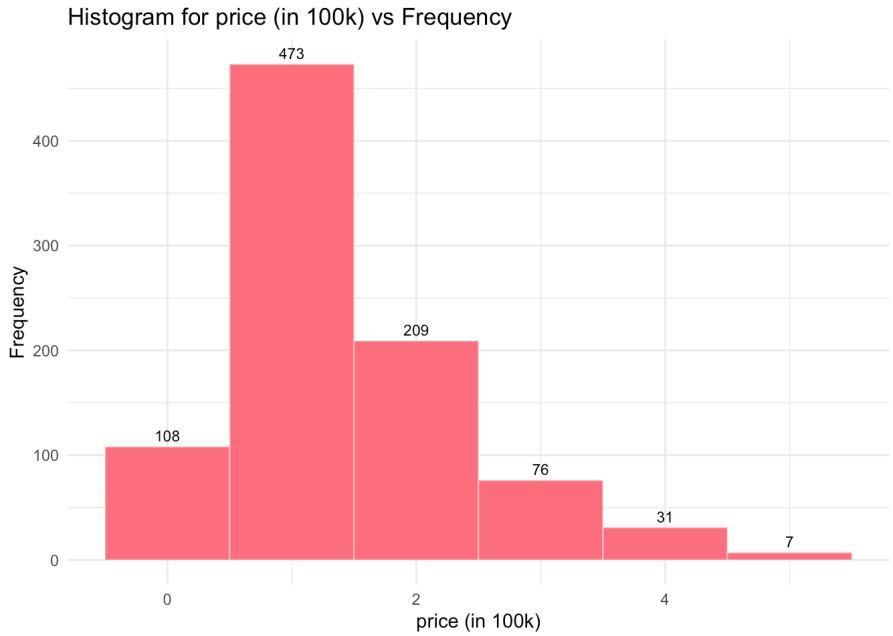
```
## [1] TRUE
```

```
# Numerical Summaries, Boxplot and Histogram for numerical variables
# Analysing price
summary(housedf$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000   79000  125000  144084  187250  500000
```

```
custom_box(housedf$price/100000, xlab="price", ylab="in 100k")
```

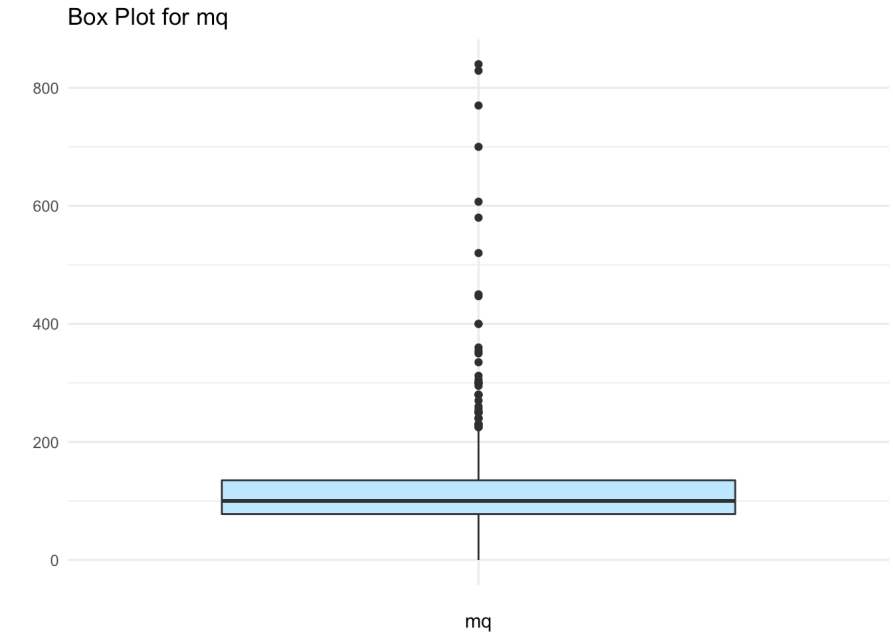

```
custom_hist(data=housedf, var=housedf$price/100000, xlab="price (in 100k)", ylab="Frequency")
```



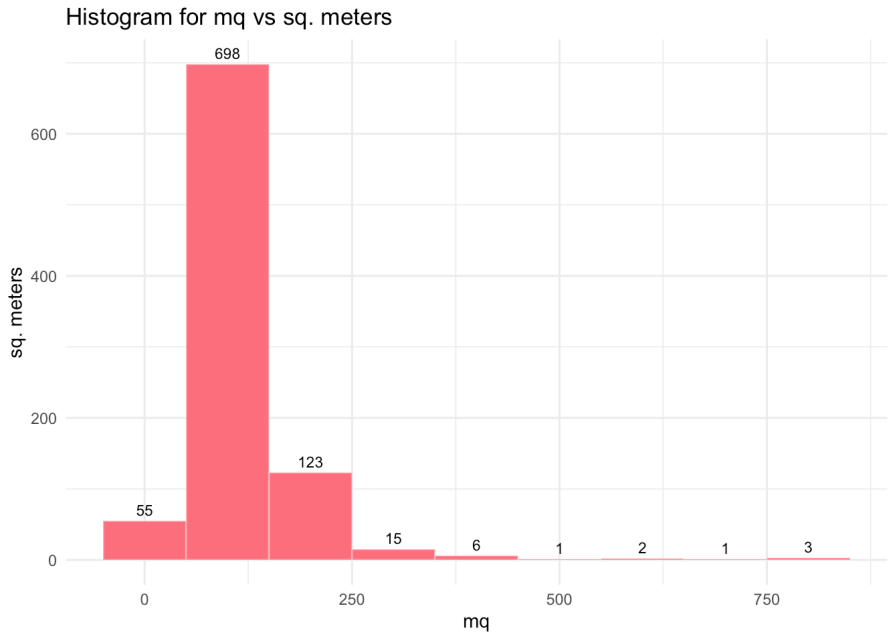

```
# Analysing mq
summary(housedf$mq)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   77.75  100.00  115.78  135.00  840.00
```

```
custom_box(housedf$mq, xlab="mq", ylab="")
```



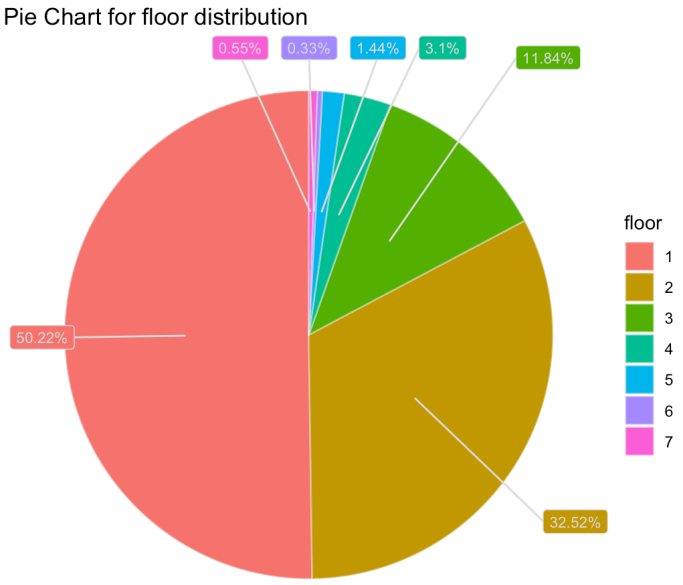

```
custom_hist(data = housedf, var = housedf$mq, xlab = "mq", ylab = "sq. meters", binwidth = 100)
```



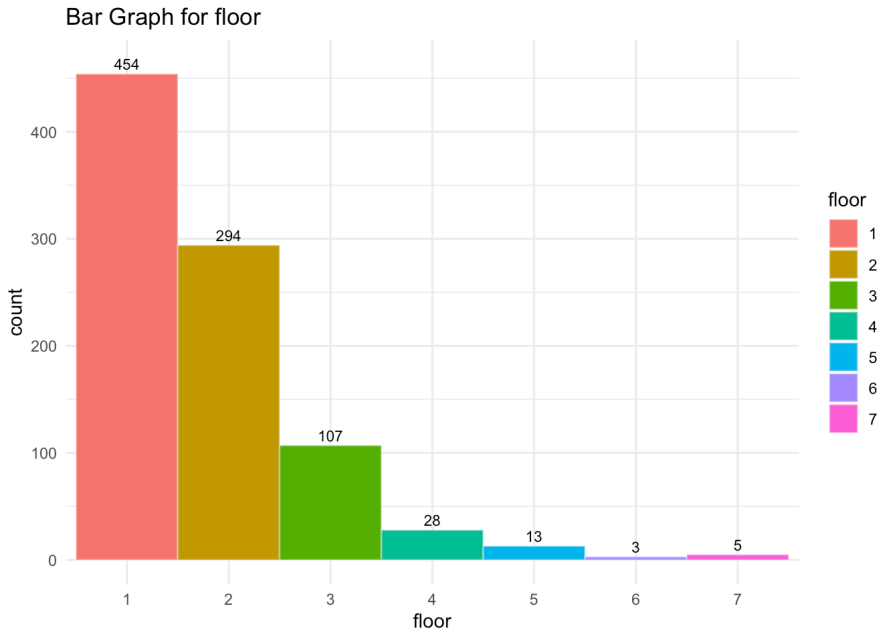

```
# Frequency Distribution Table, Piechart and Barplot for Categorical Variables
# Analysing floor
table(housedf$floor)
```

```
##
##   1   2   3   4   5   6   7
## 454 294 107  28  13   3   5
```

```
custom_pie(var=housedf$floor, title = "Pie Chart for floor distribution", xlab= "floor", ylab ="count")
```



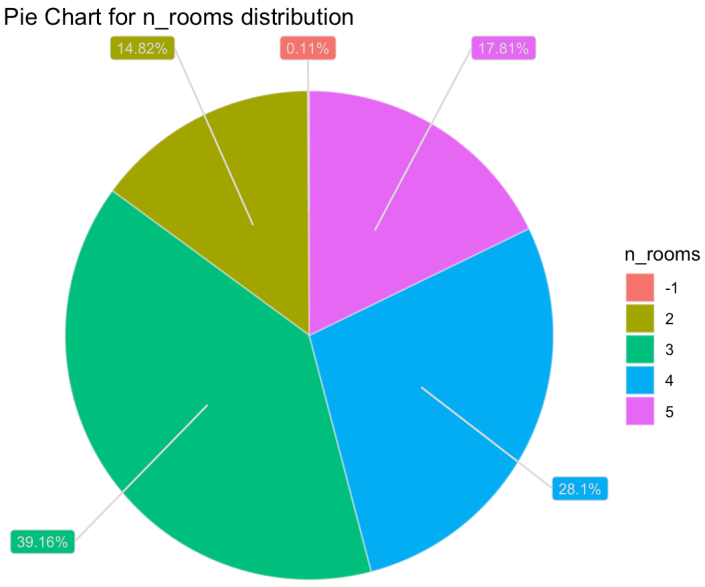

```
custom_bar(var=housedf$floor, xlab= "floor", ylab ="count")
```



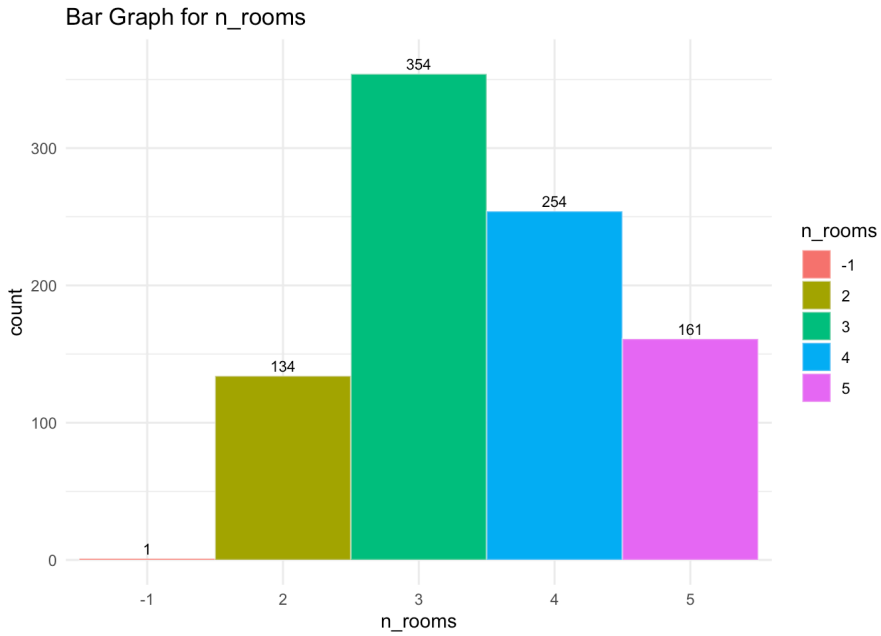

```
# Analysing n_rooms
table(housedf$n_rooms)
```

```
##
##  -1   2   3   4   5
##   1 134 354 254 161
```

```
custom_pie(var=housedf$n_rooms, title = "Pie Chart for n_rooms distribution", xlab= "n_rooms", ylab ="count")
```



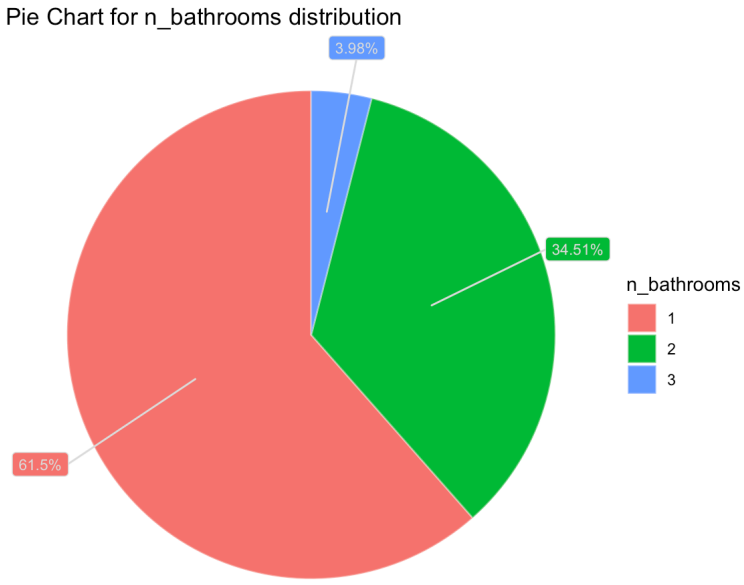

```
custom_bar(var=housedf$n_rooms, xlab= "n_rooms", ylab ="count")
```



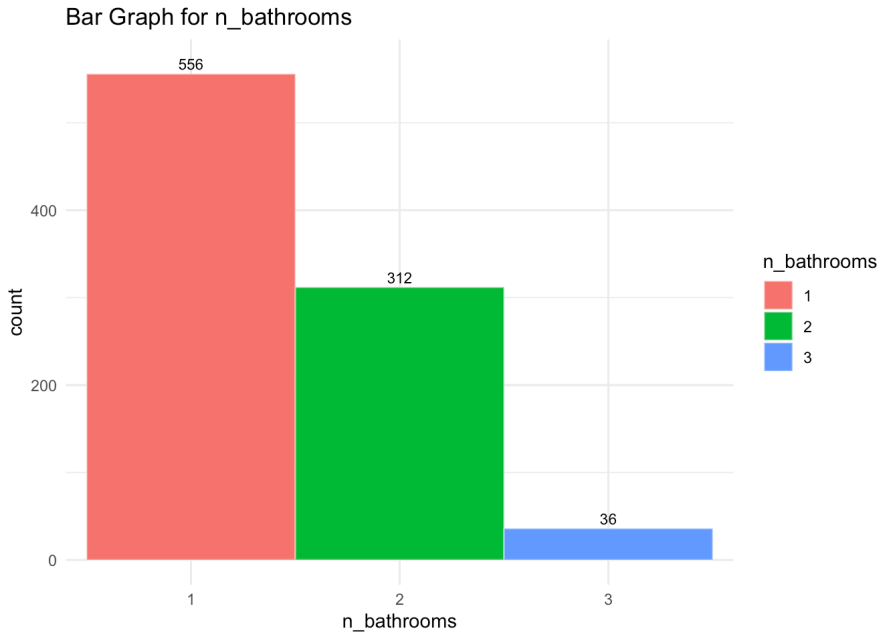

```
# Analysing n_bathrooms
table(housedf$n_bathrooms)
```

```
##
##   1   2   3
## 556 312  36
```

```
custom_pie(var=housedf$n_bathrooms, title = "Pie Chart for n_bathrooms distribution", xlab= "n_bathrooms", ylab ="count")
```



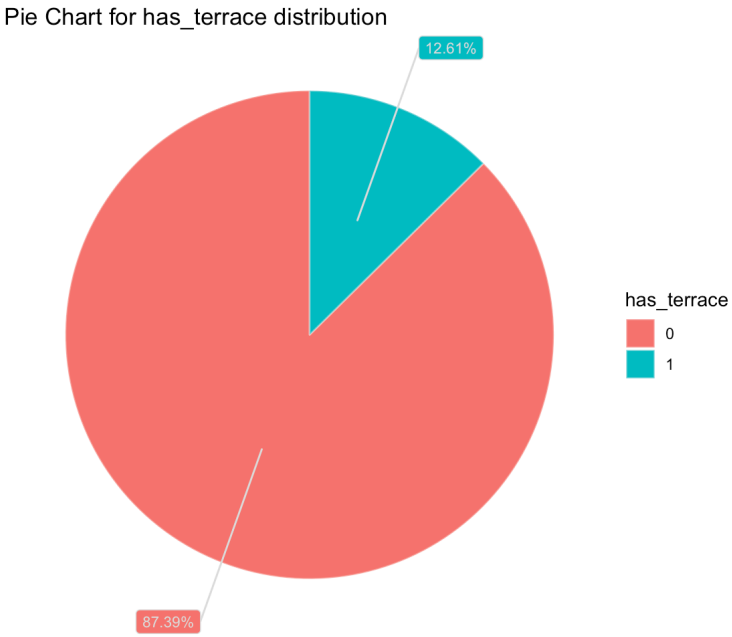

```
custom_bar(var=housedf$n_bathrooms, xlab= "n_bathrooms", ylab ="count")
```



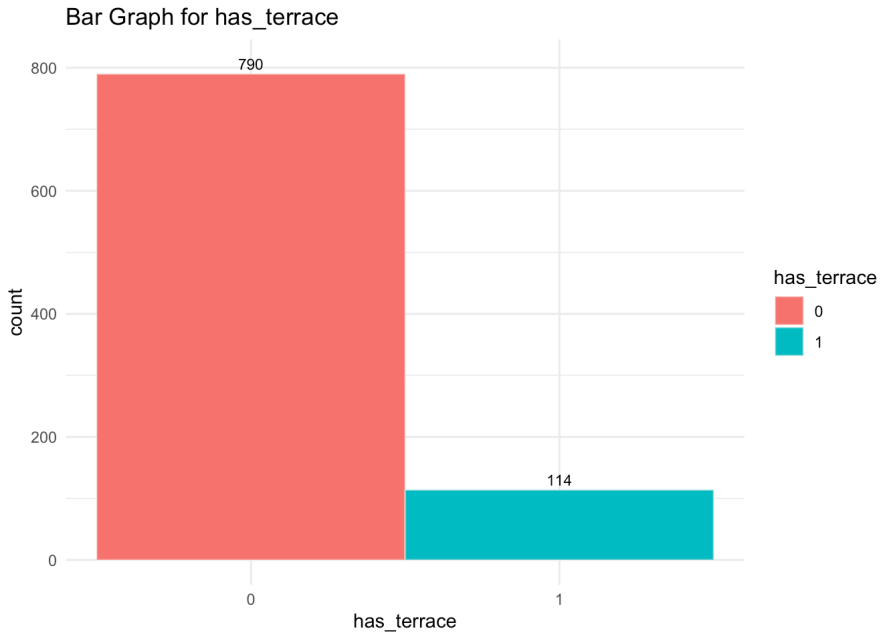

```
# Analysing has_terrace
table(housedf$has_terrace)
```

```
##
##   0   1
## 790 114
```

```
custom_pie(var=housedf$has_terrace, title = "Pie Chart for has_terrace distribution", xlab= "has_terrace", ylab ="count")
```



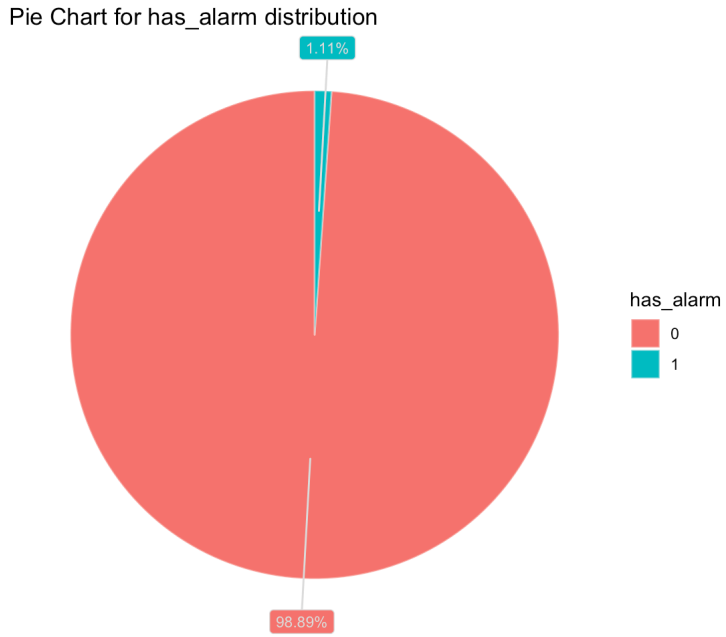

```
custom_bar(var=housedf$has_terrace, xlab= "has_terrace", ylab ="count")
```



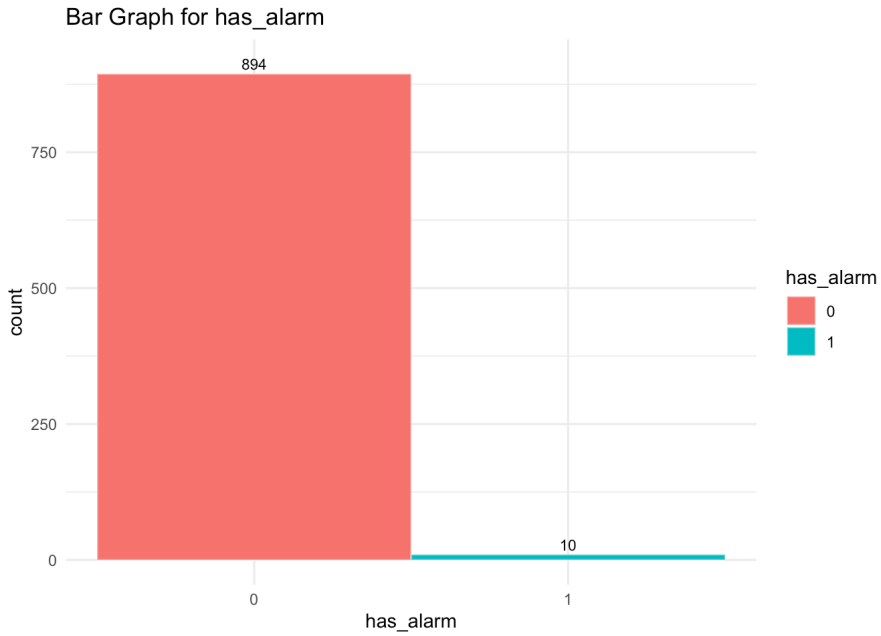

```
# Analysing has_alarm
table(housedf$has_alarm)
```

```
##
##   0   1
## 894  10
```

```
custom_pie(var=housedf$has_alarm, title = "Pie Chart for has_alarm distribution", xlab= "has_alarm", ylab ="coun
t")
```



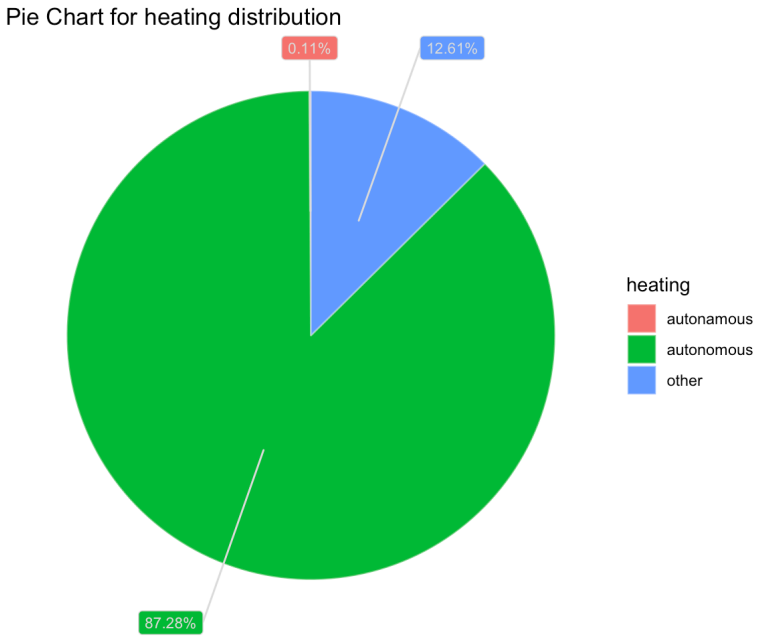

```
custom_bar(var=housedf$has_alarm, xlab= "has_alarm", ylab ="count")
```



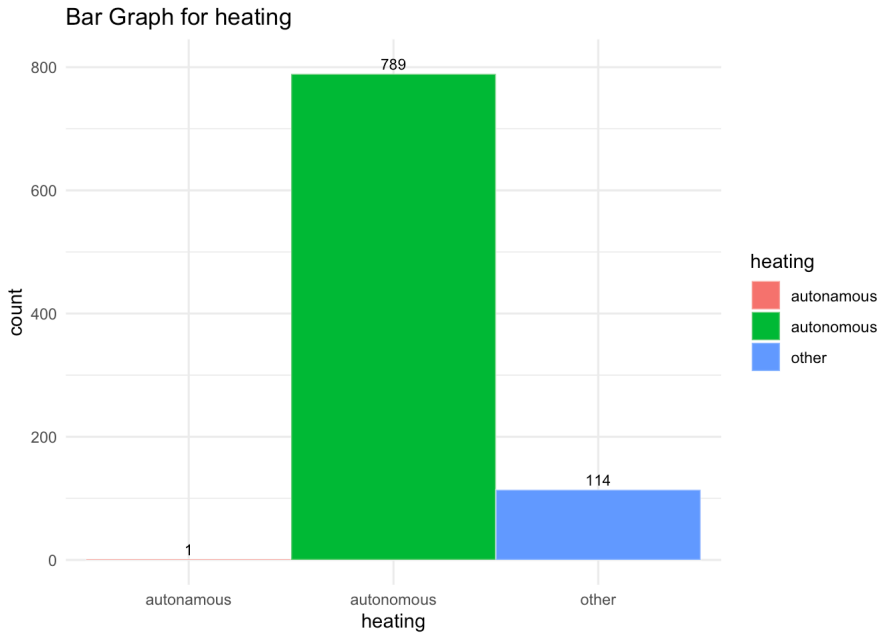

```
# Analysing heating
table(housedf$heating)
```

```
##
## autonamous autonomous      other
##          1        789        114
```

```
custom_pie(var=housedf$heating, title = "Pie Chart for heating distribution", xlab= "heating", ylab ="count")
```



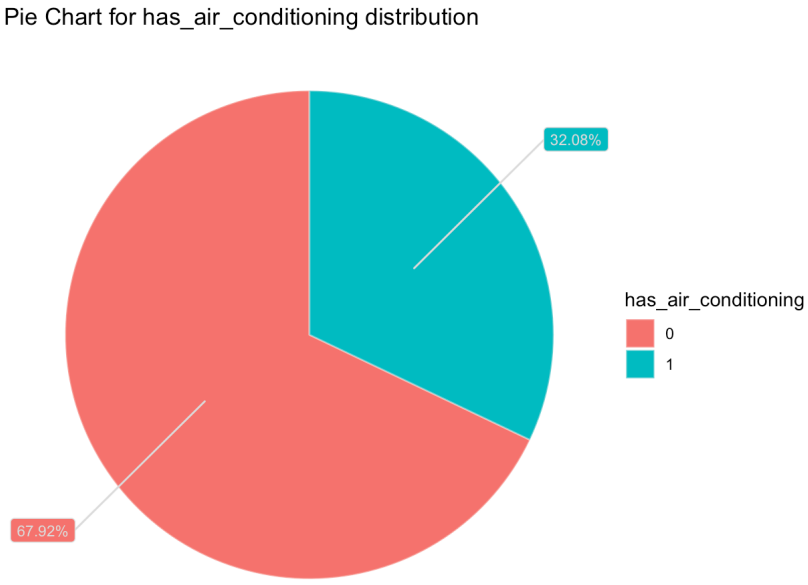

```
custom_bar(var=housedf$heating, xlab= "heating", ylab ="count")
```



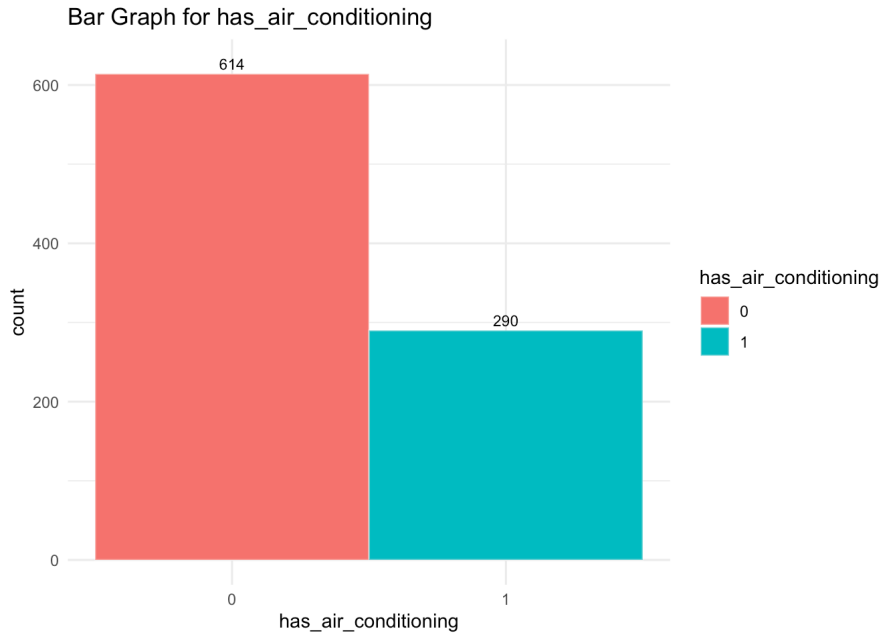

```
# Analysing has_air_conditioning
table(housedf$has_air_conditioning)
```

```
##
##   0   1
## 614 290
```

```
custom_pie(var=housedf$has_air_conditioning, title = "Pie Chart for has_air_conditioning distribution", xlab= "ha
s_air_conditioning", ylab ="count")
```



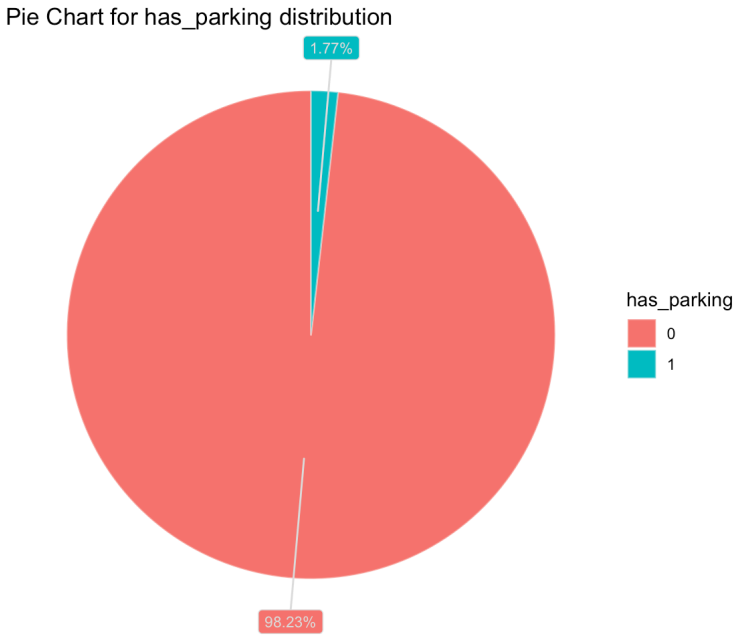

```
custom_bar(var=housedf$has_air_conditioning, xlab= "has_air_conditioning", ylab ="count")
```



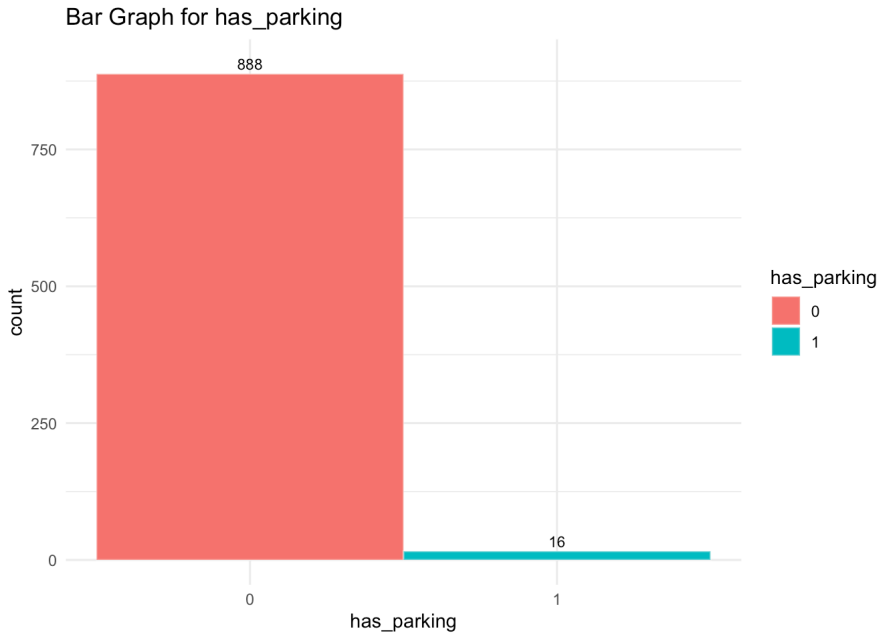

```
# Analysing has_parking
table(housedf$has_parking)
```

```
##
##   0   1
## 888  16
```

```
custom_pie(var=housedf$has_parking, title = "Pie Chart for has_parking distribution", xlab= "has_parking", ylab ="count")
```



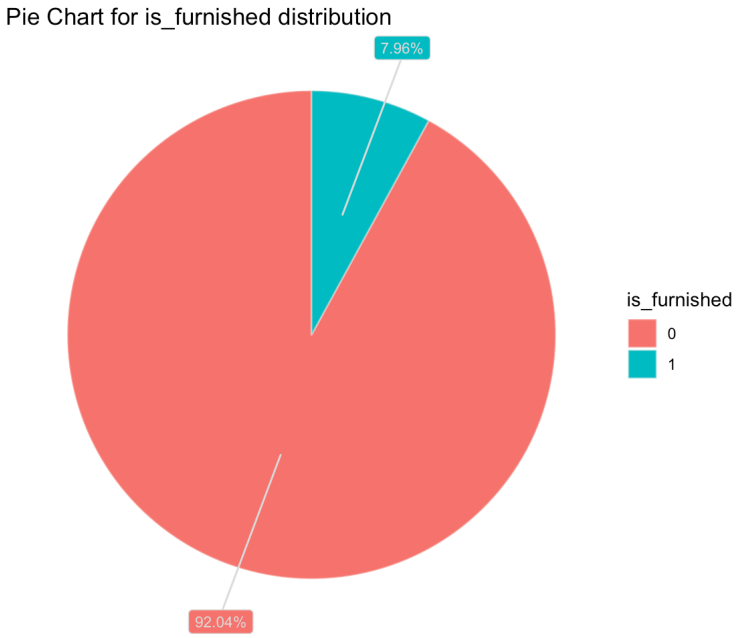

```
custom_bar(var=housedf$has_parking, xlab= "has_parking", ylab ="count")
```



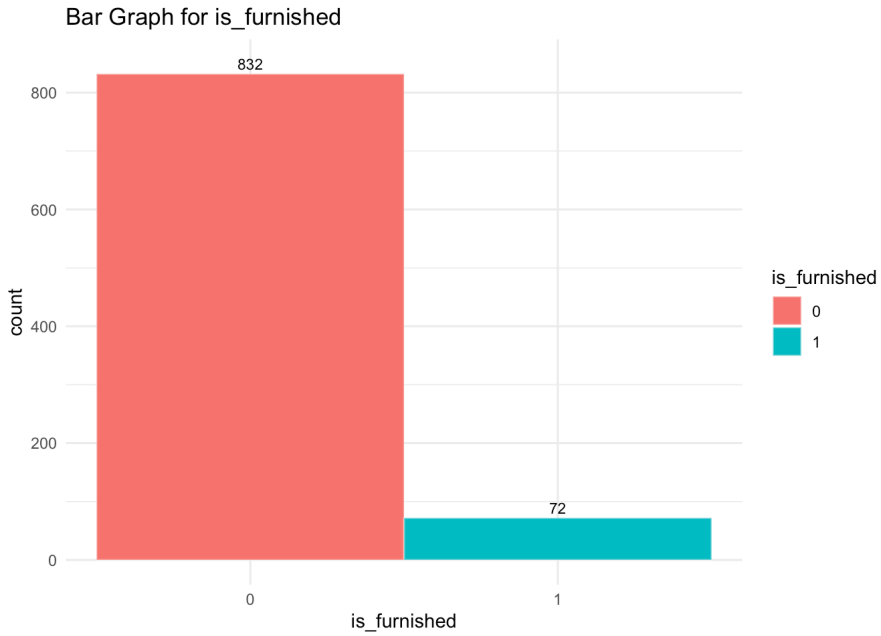

```
# Analysing is_furnished
table(housedf$is_furnished)
```

```
##
##   0   1
## 832  72
```

```
custom_pie(var=housedf$is_furnished, title = "Pie Chart for is_furnished distribution", xlab= "is_furnished", yla
b ="count")
```




```
custom_bar(var=housedf$is_furnished, xlab= "is_furnished", ylab ="count")
```



From the data, we have the following observations:

1. The dataframe has 904 observations and 12 variables whose names match the metadata.
2. The 'id' column is the index and is unique.
3. The 'price' column ranges from 1000 to 50000, with a mean of 144084 and a median of 125000.
4. The 'mq' column ranges from 0 to 840, with a mean of 115.78 and a median of 100.
5. The 'floor' column has 7 unique values, with 1 and 6 occurring most and least frequently.
6. The 'n_rooms' column has 5 unique values, with 3 and -1 occurring most and least frequently.
7. The 'n_bathrooms' column has 3 unique values, with 3 and 1 occurring most and least frequently.
8. The 'has_terrace', 'has_alarm', 'has_air_conditioning', 'has_parking', and 'is_furnished' columns each have 2 unique values - 0 and 1.
9. The 'heating' column has 3 unique values - "autonomous", "other", and "autonamous
10. There are no NAs in the data.

1.3 Data cleaning

The following data quality issues were identified in the above step:

1. Using the *str*() and *unique*() method, we see the inconsistency that *floor*, *n_rooms*, *n_bathrooms*, *has_terrace*, *has_alarm*, *has_air_conditioning*, *has_parking*, and *is_furnished* columns are implemented as integers, which does not clearly indicate their categorical nature.
2. The *confront*() method reveals data quality issues, such as a minimum value of 0 for *mq*, a minimum value of -1 for *n_rooms*, and multiple factors in the *heating* variable.
3. The boxplots of the *price* and *mq* variables show that both have outliers that could cause problems during model fitting.
4. The numerical summary of the *mq* variable shows a minimum value of 0, which is not possible for the area of a house.
5. The numerical summary of the *n_rooms* variable shows a minimum value of -1, with a frequency of 1 and frequency distribution of 0.11%. It is impossible for a house to have negative rooms.
6. The frequency table and pie chart for *heating* show a third factor 'autonamous' with a frequency of 1 and a frequency distribution of 0.11%.
7. As per metadata, *heating* should take values from either of "autonomous" or "other", and thus should be implemented as a factor rather than character.

To clean the data, we'll copy the existing data into a new dataframe *housedf_clean* and follow the listed steps in sequence:

1. We'll view the row where *mq* has an extreme value of 0 and gather similar rows. Next a correlation test between mq and price for these rows would reveal if the *p-value* > 0.05, implying no correlation. In such case, we'll remove the single observation to avoid skewing the data.
2. Next, we'll impute the *n_rooms* = -1 with the mode of *n_rooms*, as it is a single observation and it is better to merge it with existing category to avoid data being skewed.
3. We'll correct the "autonamous" to "autonomous" in the *heating* column, as it seems like a typing error and closely matches the "autonomous" factor in the column.
4. After this correction, we'll implement *floor*, *n_rooms*, *n_bathrooms*, *has_terrace*, *has_alarm*, *has_air_conditioning*, *has_parking*, *is_furnished* and *heating* as categorical variables by converting them to factors

```
# Data Cleaning

# Create new dataframe
housedf_clean <- housedf

# Viewing / correcting mq
housedf_clean[housedf_clean$mq==0,]
```

	id	price	mq	floor	n_rooms	n_bathrooms	has_terrace	has_alarm	heating

	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>	<chr>
197225	197225	72000	0	1	3	2	0	0	autonomous

1 row | 1-10 of 13 columns

```
similar_set <- housedf[housedf$floor==1 & housedf$n_rooms==3 & housedf$n_bathrooms==2 & housedf$has_terrace==0 &
                         housedf$has_alarm==0 & housedf$heating=="autonomous" & housedf$has_air_conditioning==0 &
                         housedf$has_parking==0 & housedf$is_furnished==0,]
cor.test(similar_set$price, similar_set$mq)
```

```
##
##  Pearson's product-moment correlation
##
## data:  similar_set$price and similar_set$mq
## t = -0.46526, df = 21, p-value = 0.6465
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4926966  0.3247136
## sample estimates:
##        cor
## -0.1010083
```

```
# In case of high correlation, we impute by the mean of mq from similar_set
#housedf_clean$mq[housedf_clean$mq==0] <- round(mean(similar_set$mq))
# In case of low / no correlation, we remove the observation having mq as 0
housedf_clean <- housedf_clean[which(housedf_clean$mq != 0),]

# Correcting n_rooms
# replace n_rooms == -1 with mode of n_rooms
housedf_clean$n_rooms[which(housedf$n_rooms == -1)] <- mlv(housedf_clean$n_rooms, method = "mfv")

# Correcting heating
housedf_clean$heating[housedf_clean$heating=='autonamous'] <- 'autonomous'

# Alternate corrections
# replace mq < 5, with mean of mq
#housedf_clean$mq[which(housedf_clean$mq <5)] <- mean(housedf_clean$mq)
# Replace n_rooms = -1 with positive value
#housedf_clean$n_rooms <- abs(housedf_clean$n_rooms)


# Converting the categorical columns into factors
categorical_vars <- c("floor", "n_rooms", "n_bathrooms", "has_terrace","has_alarm","heating","has_air_conditionin
g", "has_parking", "is_furnished")
for(var in categorical_vars){
  housedf_clean[,var] <- as.factor(housedf_clean[,var])
}

# Checking data types and summary for housedf_clean
summary(housedf_clean)
```

```
##        id             price              mq        floor   n_rooms n_bathrooms
##  Min.   :    84   Min.   :  1000   Min.   :  1.0   1:453   2:134   1:556
##  1st Qu.: 53898   1st Qu.: 79000   1st Qu.: 78.0   2:294   3:354   2:311
##  Median :113063   Median :125000   Median :100.0   3:107   4:254   3: 36
##  Mean   :111010   Mean   :144164   Mean   :115.9   4: 28   5:161
##  3rd Qu.:167181   3rd Qu.:187500   3rd Qu.:135.0   5: 13
##  Max.   :223309   Max.   :500000   Max.   :840.0   6:  3
##                                                    7:  5
##  has_terrace has_alarm       heating    has_air_conditioning has_parking
##  0:789       0:893     autonomous:789   0:613                0:887
##  1:114       1: 10     other     :114   1:290                1: 16
##
##
##
##
##
##  is_furnished
##  0:831
##  1: 72
##
##
##
##
##
```

2. Exploratory Data Analysis (EDA)

2.1 EDA plan

To perform exploratory data analysis on our data, we will follow the following steps:

1. We'll individually analyse each variable using a custom *mysummary()* method, which generates numerical summary for each variable, alongside skewness check and histogram for numerical variables, and bar graph for categorical variables.
2. Next, we'll check the correlation between the numerical variables *price* and *mq* using *cor.test()* and *scatter plots*.
3. To check the difference of average mean for *price* and *mq* according to different categories of the categorical variables, we'll calculate the aggregate mean and test the significant difference using ANOVA. Additionally, we'll draw *boxplots* between pairs of numerical and categorical variables for the same purpose.
4. For checking dependency between categorical variables, we'll use the Chi-Square test. Since some of the floors have <5 observations, we'll use fisher's exact test.

2.2 EDA and summary of results

From the above exploratory data analysis, the following conclusions can be made:

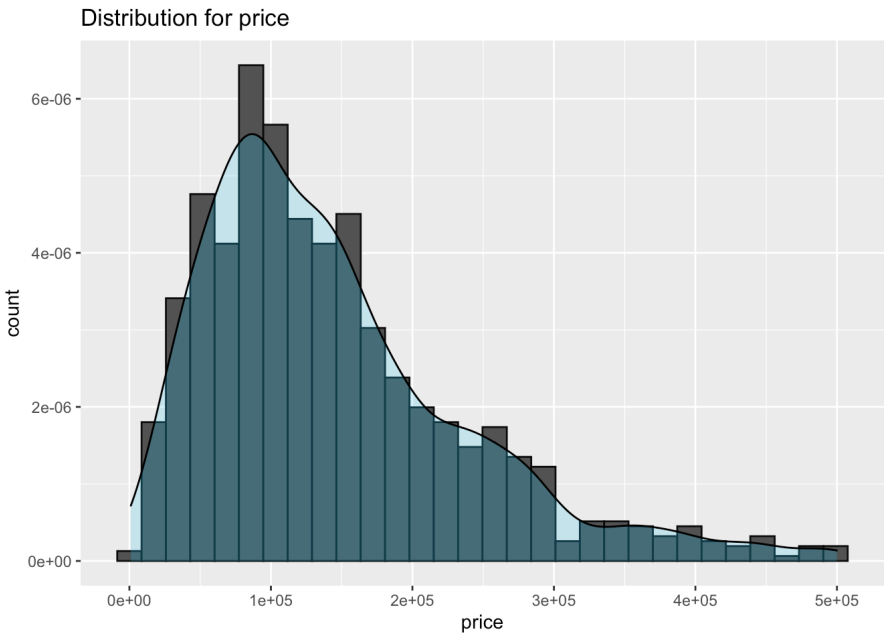
1. As evident by the histogram, qqplot and skewness check, the distributions for *price* and *mq* are not normally distributed. In fact, as both the skewness values are positive (1.23 and 4.54 respectively), the distributions are right skewed.
2. A significant majority of the houses are located on lower floors, resulting in a right-skewed distribution of the *floor* variable.
3. The most common *n_rooms* is 3, with a frequency of 354, which decreases on either side.
4. *n_bathrooms* are concentrated on the lower end with maximum 556 observations having n_bathrooms1. The observations' frequency decreases as *n_bathrooms* increases.
5. The majority of houses lack certain amenities, such as terrace (789), alarms (893), air conditioning (613), parking (887), and furnishings (831).
6. 789 houses have autonomous heating, while 114 have other.
7. There is a moderate positive correlation (0.317) between price and mq variables, as shown in the cor.test output and model fit line in scatter plot.
8. The *Chi-Square and Fisher Exact test* (for *floor*) shows there is a significant dependence between certain pairs of categorical variables (p-value < 0.05):
    - *floor* with *heating* and *has_air_conditioning*
    - *n_rooms* with *n_bathrooms*
    - *n_bathrooms* with *has_terrace, has_alarm,* and *heating*
    - *has_terrace* with *has_air_conditioning, has_parking, is_furnished* and *has_air_conditioning*
    - *has_alarm* with *has_air_conditioning*
    - *has_air_conditioning* with *has_parking* and *is_furnished*

      Errors occurred during the test due to non-normal distribution of the variables. We are using simulate.p.value = T for ignoring those errors.
9. There is no significant difference or trend in *price* or *mq* across groups of *floor*, *has_air_conditioning*, *has_parking* and *is_furnished*. These variables may not significantly affect house price or size.
10. From aggregate mean and boxplots, we see that as the number of rooms and bathrooms increases, the average mq and price also tend to increase.
11. Houses with terrace or alarm are significantly more expensive (shown by ANOVA output) and slightly larger (but not significantly so) compared to houses without a terrace or alarm.
12. Houses with autonomous heating are significantly larger, despite not being significantly expensive from those with other heating modes.

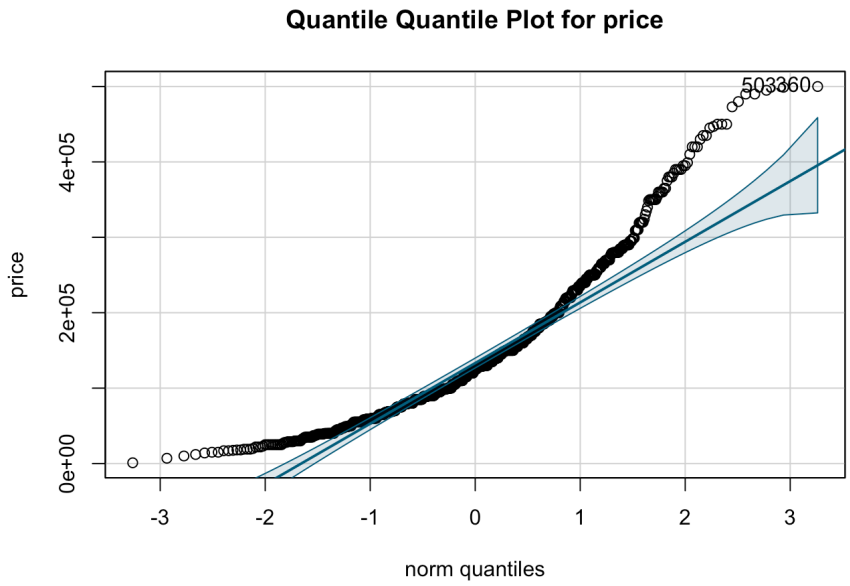
```
# Defining a custom function mysummary() to show variable summary
# and visualise the data in the form of histogram and bar plot
mysummary <- function(var, var_name){
  print(summary(var))
  # For numerical variables
  if(class(var) == "numeric" || class(var) == "integer"){
    # Histogram with density line
    print(ggplot(housedf_clean, aes(x = var)) +
            geom_histogram(aes(y=..density..), color="#111111", fill="#555555") +
            geom_density(alpha=.2, fill="#00C0F0") +
            labs(title = paste("Distribution for", var_name), x = var_name, y = "count"))
    # Skewness
    print(skewness(var))
    # QQplot
    qqPlot(var, distribution="norm", ylab = var_name, main = paste("Quantile Quantile Plot for",var_name), col.li
nes = "#006080")
  }
  # For Categorical Variables
  else{
    freq.df <- as.data.frame(table(var))
    # Bar Plot
    print(ggplot(data = housedf_clean) + aes(x=var) + geom_bar(color="#111111", fill="#555555") +
            geom_text(data = freq.df, aes(x = var, y = Freq + max(Freq)/50, label = Freq), size=3) +
            xlab(var_name) + ylab("Frequency") + ggtitle(paste("Distribution for", var_name)))
  }
}

# Analysing Individual Variable

# Analysing price
mysummary(housedf_clean$price, var_name = "price")
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000   79000  125000  144164  187500  500000
```

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




```
## [1] 1.233913
```

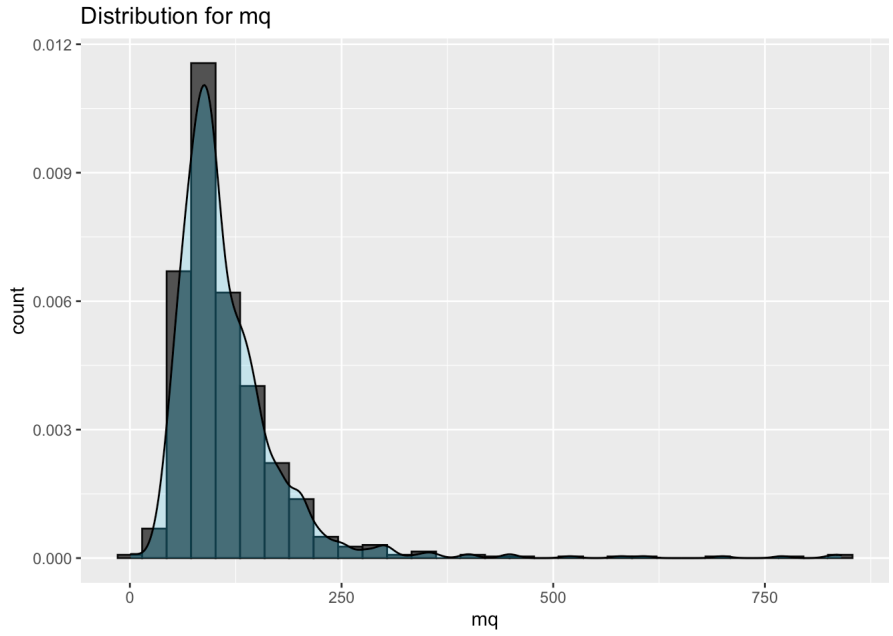



```
## [1] 360 503
```

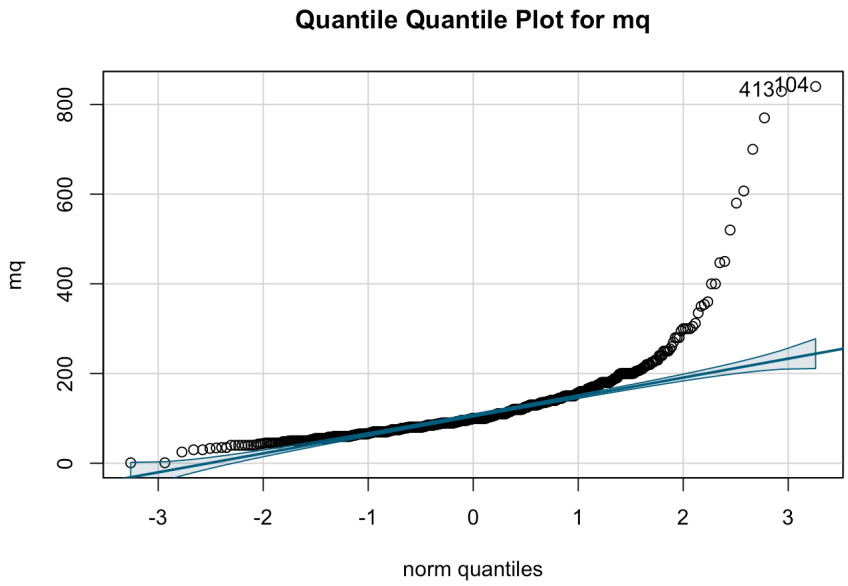
```
# Analysing mq
mysummary(housedf_clean$mq, var_name = "mq")
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    78.0   100.0   115.9   135.0   840.0
```

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



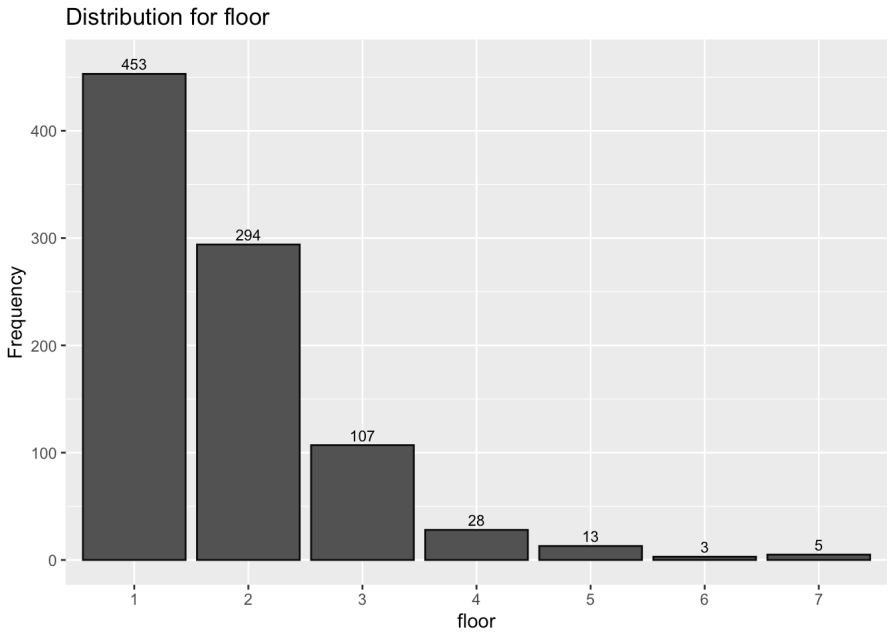

```
## [1] 4.546405
```




```
## [1] 104 413
```

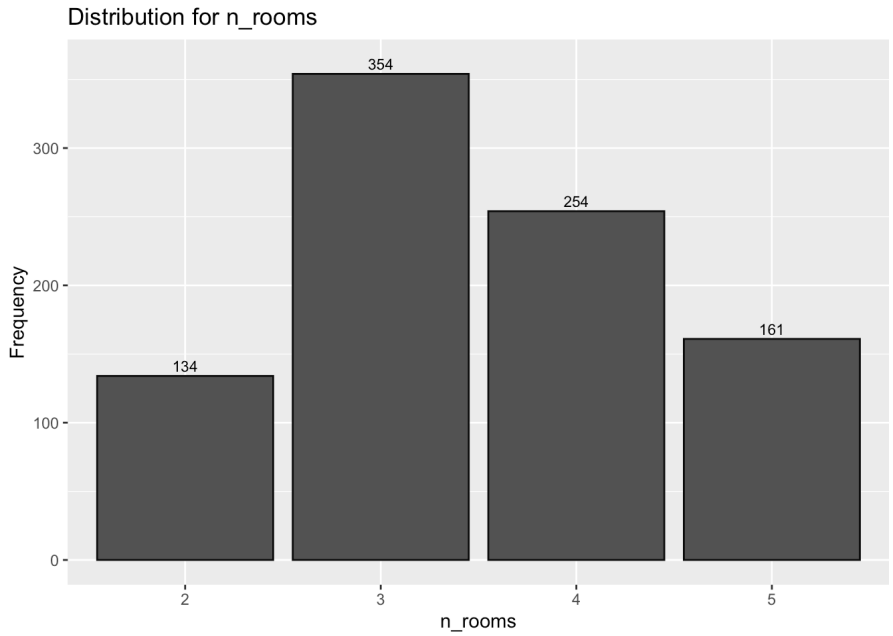
```
# Analysing floor
mysummary(housedf_clean$floor, var_name = "floor")
```

```
##   1   2   3   4   5   6   7
## 453 294 107  28  13   3   5
```



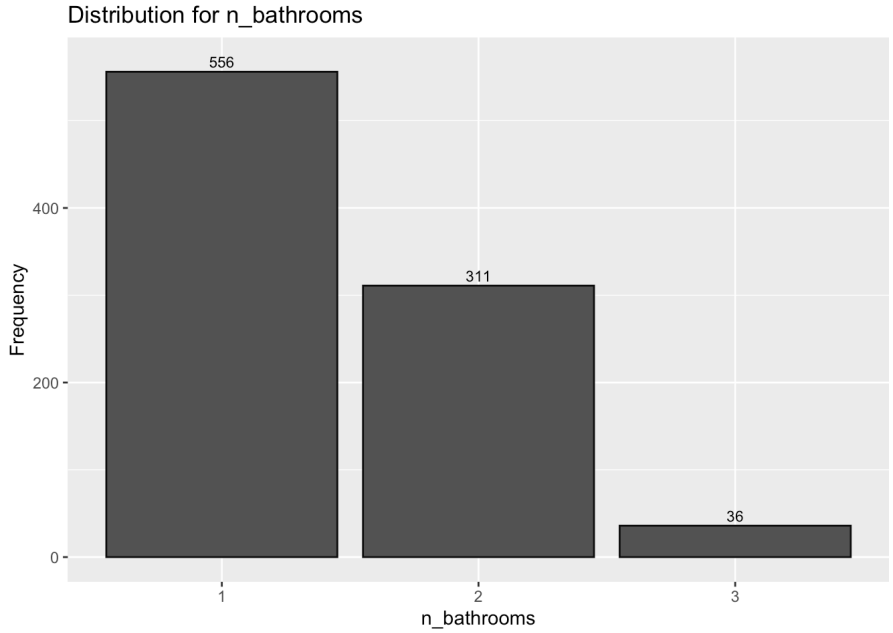

```
# Analysing n_rooms
mysummary(housedf_clean$n_rooms, var_name = "n_rooms")
```

```
##   2   3   4   5
## 134 354 254 161
```



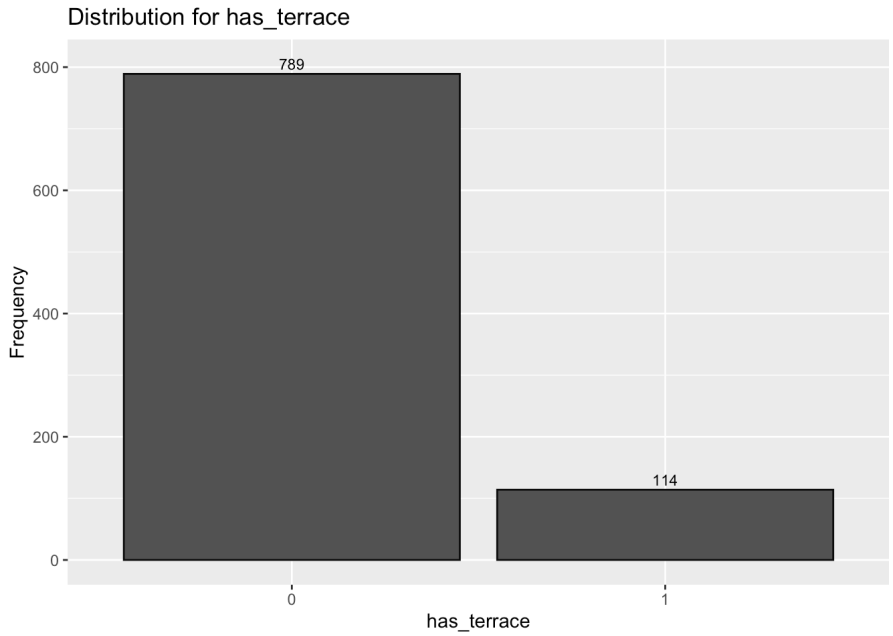

```
# Analysing n_bathrooms
mysummary(housedf_clean$n_bathrooms, var_name = "n_bathrooms")
```

```
##   1   2   3
## 556 311  36
```



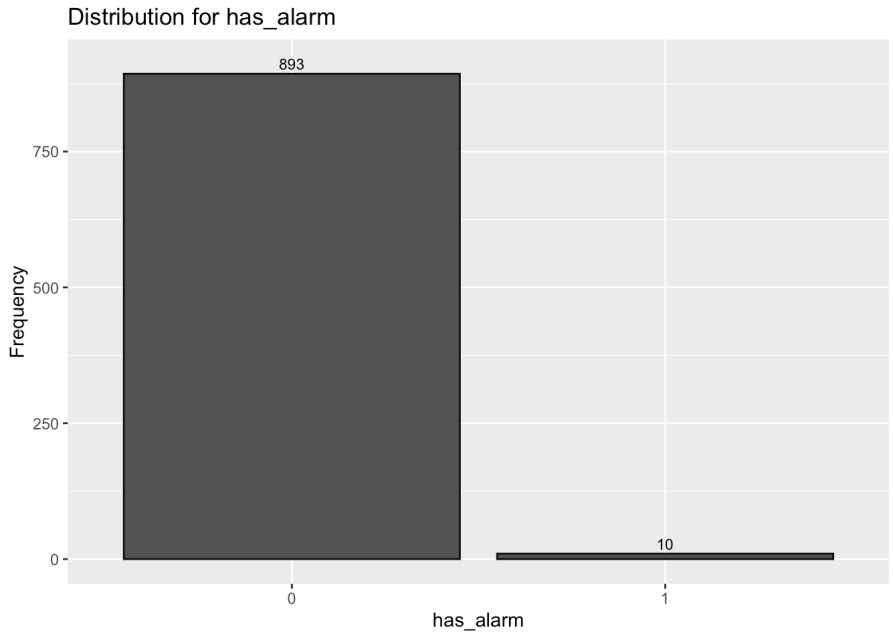

```
# Analysing has_terrace
mysummary(housedf_clean$has_terrace, var_name = "has_terrace")
```

```
##   0   1
## 789 114
```



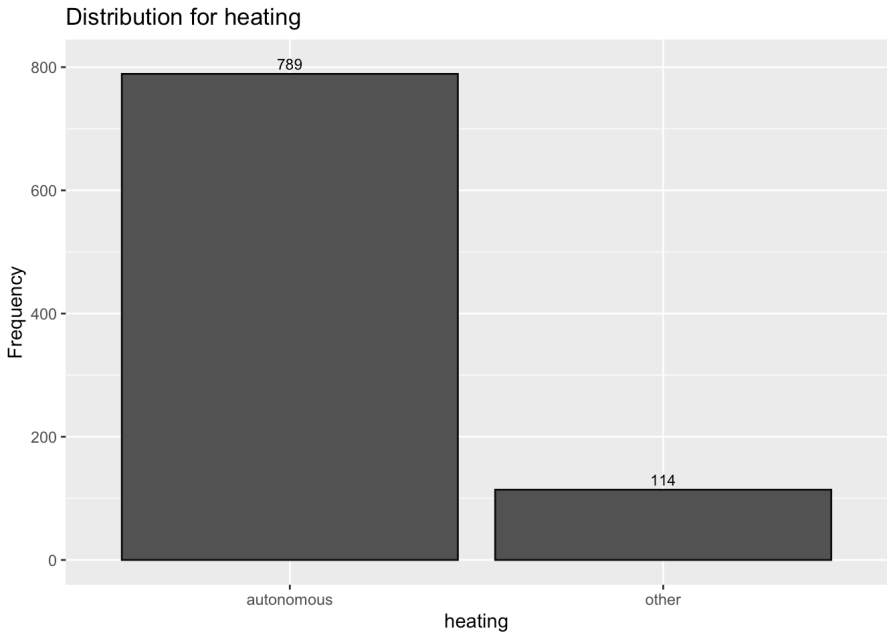

```
# Analysing has_alarm
mysummary(housedf_clean$has_alarm, var_name = "has_alarm")
```

```
##   0   1
## 893  10
```



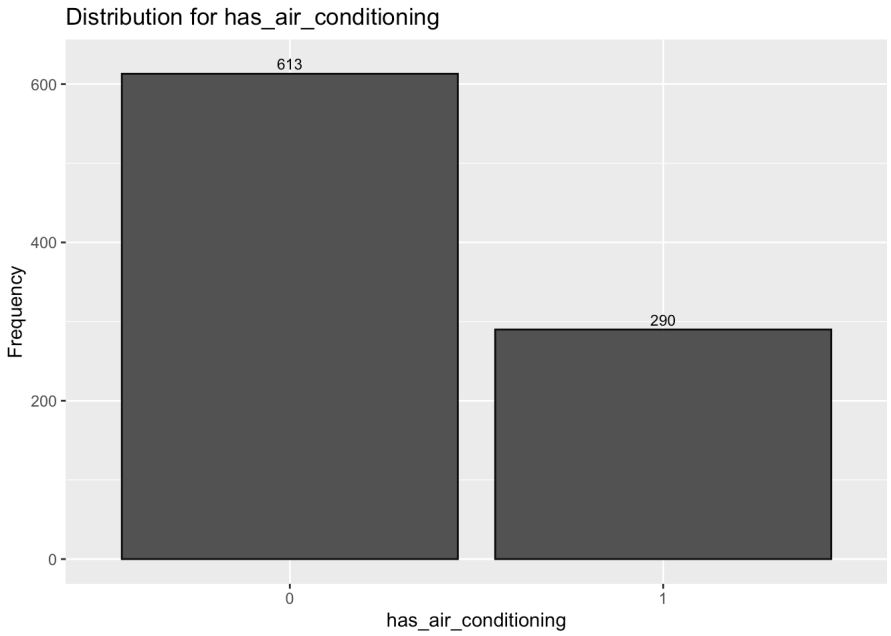

```
# Analysing heating
mysummary(housedf_clean$heating, var_name = "heating")
```

```
## autonomous      other
##        789        114
```



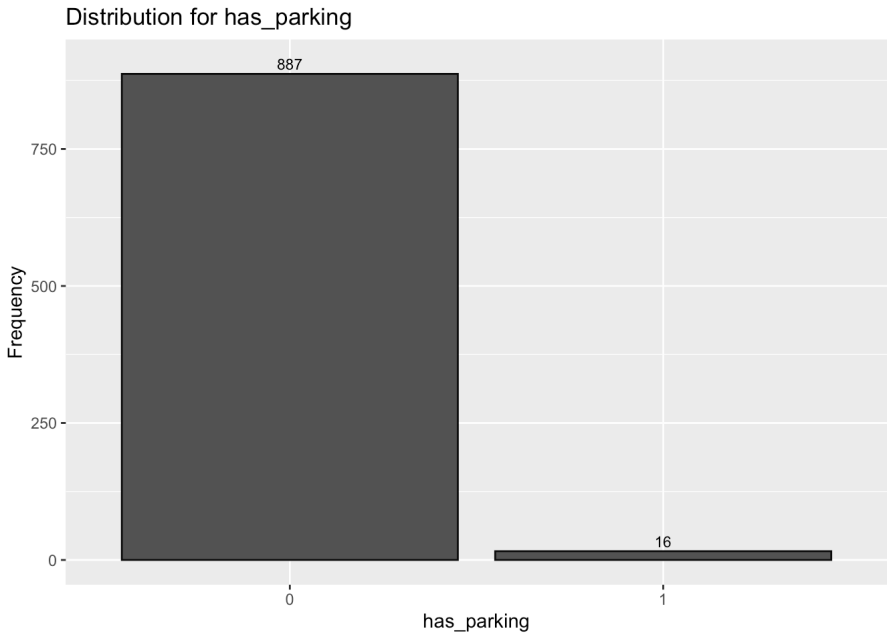

```
# Analysing has_air_conditioning
mysummary(housedf_clean$has_air_conditioning, var_name = "has_air_conditioning")
```

```
##   0   1
## 613 290
```



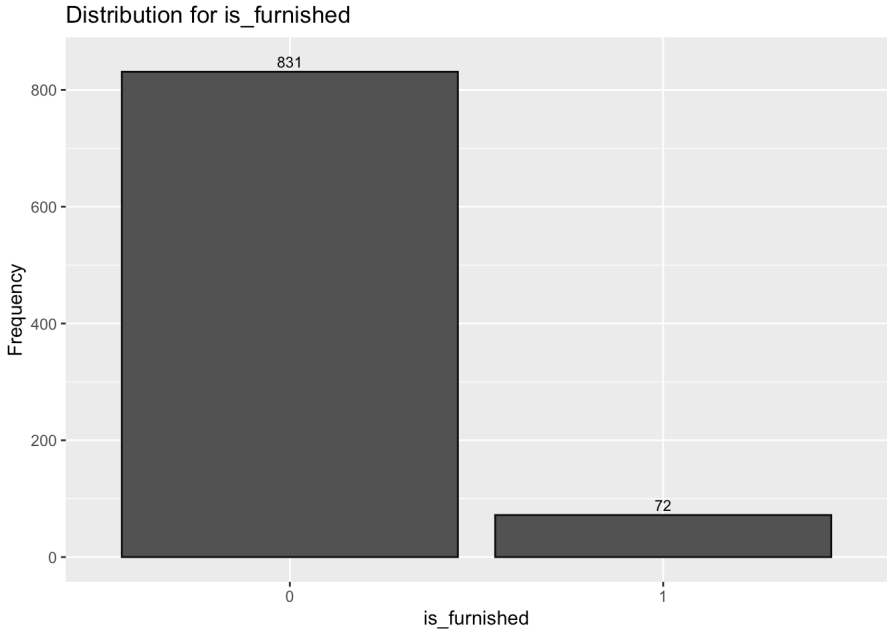

```
# Analysing has_parking
mysummary(housedf_clean$has_parking, var_name = "has_parking")
```

```
##   0   1
## 887  16
```




```
# Analysing is_furnished
mysummary(housedf_clean$is_furnished, var_name = "is_furnished")
```

```
##   0   1
## 831  72
```

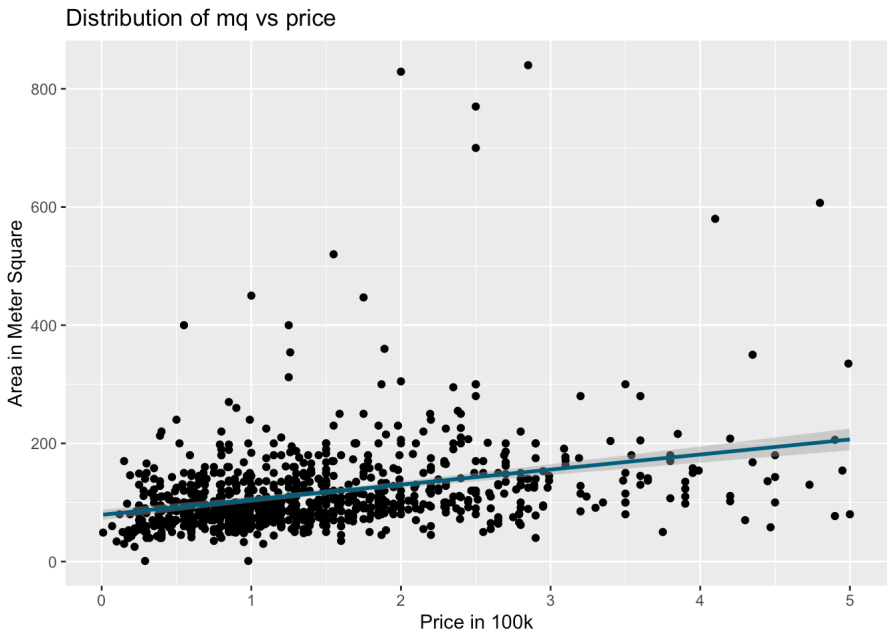



```
# Checking correlation between numerical columns

# Checking correlation between the price and mq
cor.test(housedf_clean$price, housedf_clean$mq)
```

```
##
##  Pearson's product-moment correlation
##
## data:  housedf_clean$price and housedf_clean$mq
## t = 10.036, df = 901, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2571621 0.3745726
## sample estimates:
##       cor
## 0.3170817
```

```
# Scatter plots for dependency between price and mq
ggplot(data = housedf_clean) + aes(x=price/100000, y=mq) + geom_point() +
  labs(title = "Distribution of mq vs price", x="Price in 100k", y="Area in Meter Square") +
  geom_smooth(method = "lm", color = "#006080")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Separate floor from rest of categorical variables for using fisher.test
# as the frequency of floor6, floor7 is 3 and 5
remaining.categorical <- categorical_vars[!categorical_vars == "floor"]

# Checking dependence among floor and remaining categorical variables using fisher.exact
# Perform the fisher.exact test of independence for floor and other categorical columns
for (j in 1:length(remaining.categorical)) {
    fisher.result <- fisher.test(table(housedf_clean$floor, housedf_clean[,remaining.categorical[j]]), simulate.
p.value = T)
    if(fisher.result$p.value<0.05){
      cat("\n")
      print(paste("Dependance between floor and", remaining.categorical[j], "is significant with a p-value of", r
ound(fisher.result$p.value,4)))
    }
}
```

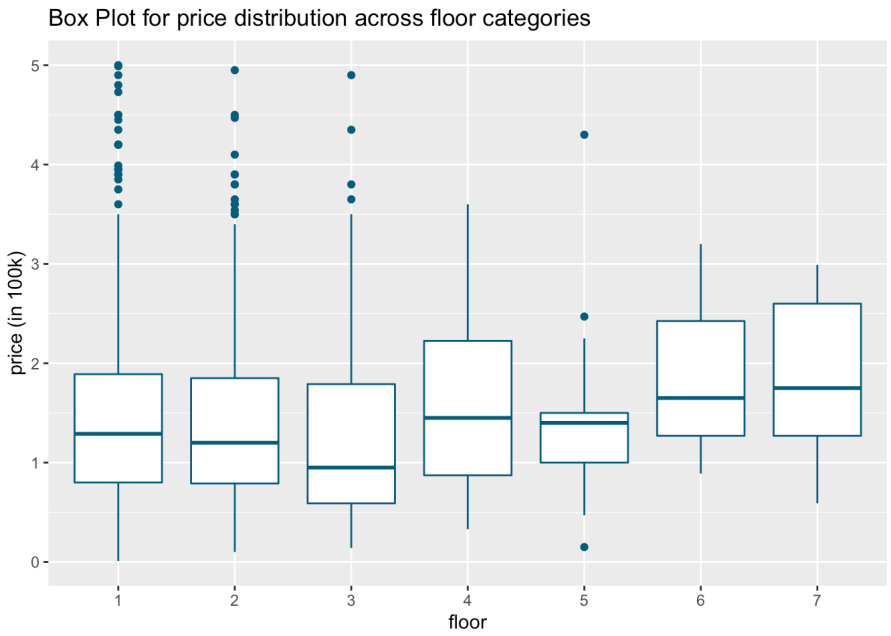
```
##
## [1] "Dependance between floor and heating is significant with a p-value of 5e-04"
##
## [1] "Dependance between floor and has_air_conditioning is significant with a p-value of 0.034"
```

```
# Checking dependence among remaining categorical variables using chi-sq test
# Perform the chi-squared test of independence for each pair of columns
for (i in 1:(length(remaining.categorical)-1)) {
  for (j in (i+1):length(remaining.categorical)) {
    chisq.result <- chisq.test(housedf_clean[,remaining.categorical[i]], housedf_clean[,remaining.categorical[
j]], simulate.p.value = T)
    #print(paste("Chi-Square Test between",columns[i], "and", columns[j], "gives p-value of", round(chisq.result$
p.value,3)))
    if(chisq.result$p.value<0.05){
      cat("\n")
      print(paste("Dependance between",remaining.categorical[i], "and", remaining.categorical[j], "is significant
with a p-value of", round(chisq.result$p.value,4)))
      }
  }
}
```

```
##
## [1] "Dependance between n_rooms and n_bathrooms is significant with a p-value of 5e-04"
##
## [1] "Dependance between n_bathrooms and has_terrace is significant with a p-value of 0.003"
##
## [1] "Dependance between n_bathrooms and has_alarm is significant with a p-value of 0.04"
##
## [1] "Dependance between n_bathrooms and heating is significant with a p-value of 0.0325"
##
## [1] "Dependance between has_terrace and has_air_conditioning is significant with a p-value of 5e-04"
##
## [1] "Dependance between has_terrace and has_parking is significant with a p-value of 0.0085"
##
## [1] "Dependance between has_terrace and is_furnished is significant with a p-value of 0.0205"
##
## [1] "Dependance between has_alarm and has_air_conditioning is significant with a p-value of 0.0045"
##
## [1] "Dependance between has_air_conditioning and has_parking is significant with a p-value of 0.0015"
##
## [1] "Dependance between has_air_conditioning and is_furnished is significant with a p-value of 5e-04"
```

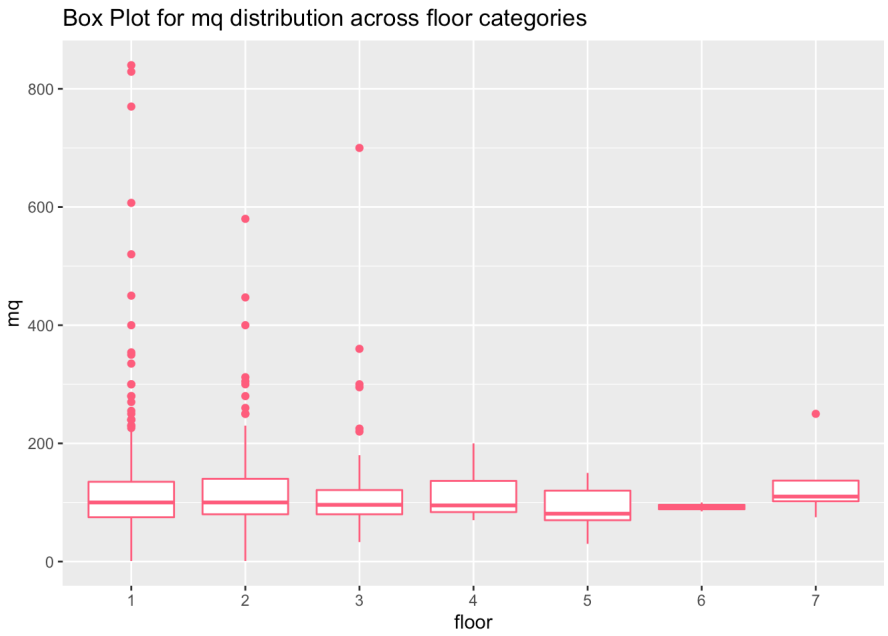
```
# Defining a custom function compare.category() to check the following
# for all (numerical variable, categorical variables) pairs in the data:
# 1) Aggregate mean for numerical variable across different categories of the categorical variable
# 2) ANOVA testing for difference in average value of numerical variable across different categories
# 3) Box Plot indicating the spread of numerical variable across different categories

compare.category <- function(num_var, cat_var){

  # Creating custom formula for categorical variable ~ numerical variable
  compare.formula <- as.formula(paste0(num_var," ~ ",cat_var))

  # Aggregate mean for numerical variable vs categorical variable
  print(paste("Aggregate Mean of",num_var,"between categories of",cat_var))
  print(aggregate(compare.formula,  data = housedf_clean, FUN="mean"))
  cat("\n\n")

  # ANOVA test to check if there's significant difference in average of the numerical variable of different categ
ories
  print(paste("ANOVA Test for",num_var,"~",cat_var))
  print(summary.lm(aov(compare.formula, data=housedf_clean)))

  # Auxiliary code for aesthetics and scaling
  x_var <- housedf_clean[,cat_var]
  y_var <- housedf_clean[,num_var]
  box_col <- "#FF6080"
  y_label <- num_var
  if(num_var == "price"){
    y_var <- y_var / 100000
    y_label <- paste(num_var,"(in 100k)")
    box_col <- "#006080"
  }

  print(paste("Box Plot for",num_var,"vs",cat_var))
  # Box plot for price vs categorical
  print(ggplot(data = housedf_clean) + aes(x = x_var, y = y_var) +
    geom_boxplot(notch = F, color = box_col) +
    labs(title = paste("Box Plot for",num_var,"distribution across",cat_var,"categories"), x = cat_var, y = y_lab
el))
}

# Aggregate Mean, ANOVA Test, and Boxplot for price vs floor
compare.category(num_var = "price", cat_var = "floor")
```

```
## [1] "Aggregate Mean of price between categories of floor"
##   floor    price
## 1     1 146601.9
## 2     2 142478.3
## 3     3 130980.4
## 4     4 156803.6
## 5     5 152384.6
## 6     6 191333.3
## 7     7 184000.0
##
##
## [1] "ANOVA Test for price ~ floor"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -145602  -66703  -20602   42522  359020
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   146602       4408  33.255   <2e-16 ***
## floor2         -4124       7027  -0.587    0.557
## floor3        -15622      10085  -1.549    0.122
## floor4         10202      18271   0.558    0.577
## floor5          5783      26394   0.219    0.827
## floor6         44731      54350   0.823    0.411
## floor7         37398      42192   0.886    0.376
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93830 on 896 degrees of freedom
## Multiple R-squared:  0.005307,   Adjusted R-squared:  -0.001354
## F-statistic: 0.7968 on 6 and 896 DF,  p-value: 0.5725
##
## [1] "Box Plot for price vs floor"
```



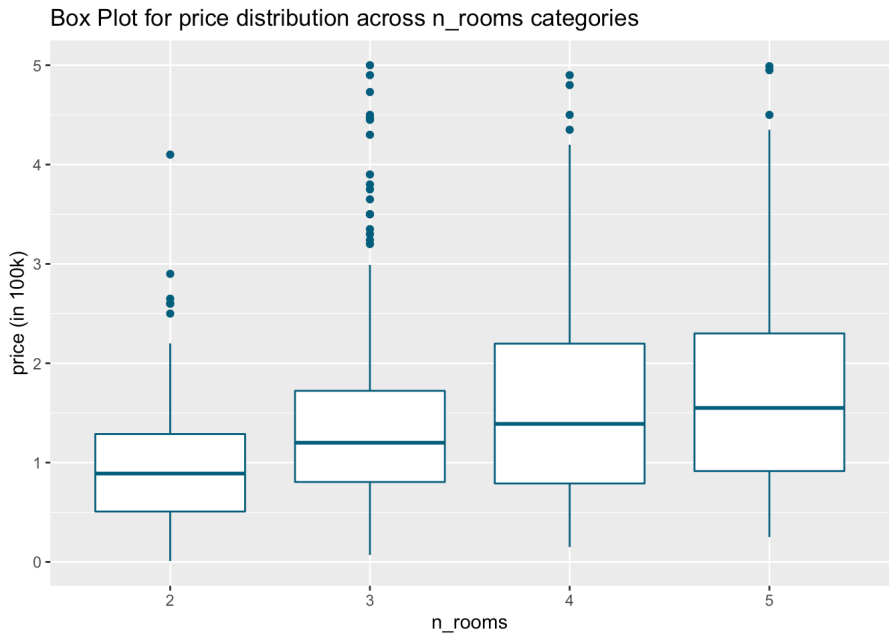

```
# Aggregate Mean, ANOVA Test, and Boxplot for mq vs floor
compare.category(num_var = "mq", cat_var = "floor")
```

```
## [1] "Aggregate Mean of mq between categories of floor"
##   floor        mq
## 1     1 117.36424
## 2     2 116.41156
## 3     3 111.80374
## 4     4 113.14286
## 5     5  91.69231
## 6     6  92.33333
## 7     7 134.80000
##
##
## [1] "ANOVA Test for mq ~ floor"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -116.36  -38.36  -17.36   18.59  722.64
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 117.3642     3.5497  33.063   <2e-16 ***
## floor2       -0.9527     5.6583  -0.168    0.866
## floor3       -5.5605     8.1208  -0.685    0.494
## floor4       -4.2214    14.7126  -0.287    0.774
## floor5      -25.6719    21.2529  -1.208    0.227
## floor6      -25.0309    43.7642  -0.572    0.567
## floor7       17.4358    33.9739   0.513    0.608
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.55 on 896 degrees of freedom
## Multiple R-squared:  0.002755,   Adjusted R-squared:  -0.003923
## F-statistic: 0.4125 on 6 and 896 DF,  p-value: 0.871
##
## [1] "Box Plot for mq vs floor"
```



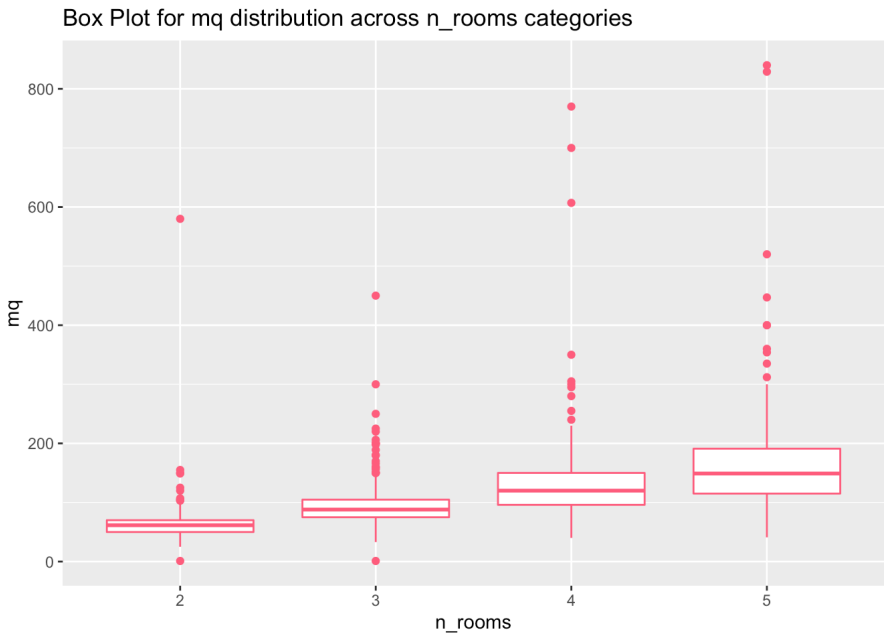

```
# Aggregate Mean, ANOVA Test, and Boxplot for price vs n_rooms
compare.category(num_var = "price", cat_var = "n_rooms")
```

```
## [1] "Aggregate Mean of price between categories of n_rooms"
##   n_rooms     price
## 1       2  96179.75
## 2       3 138795.45
## 3       4 157038.19
## 4       5 175591.93
##
##
## [1] "ANOVA Test for price ~ n_rooms"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -150592  -64488  -17038   42962  361205
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    96180       7843  12.263  < 2e-16 ***
## n_rooms3       42616       9208   4.628 4.24e-06 ***
## n_rooms4       60858       9694   6.278 5.32e-10 ***
## n_rooms5       79412      10616   7.480 1.77e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 90790 on 899 degrees of freedom
## Multiple R-squared:  0.06556,    Adjusted R-squared:  0.06244
## F-statistic: 21.02 on 3 and 899 DF,  p-value: 3.614e-13
##
## [1] "Box Plot for price vs n_rooms"
```



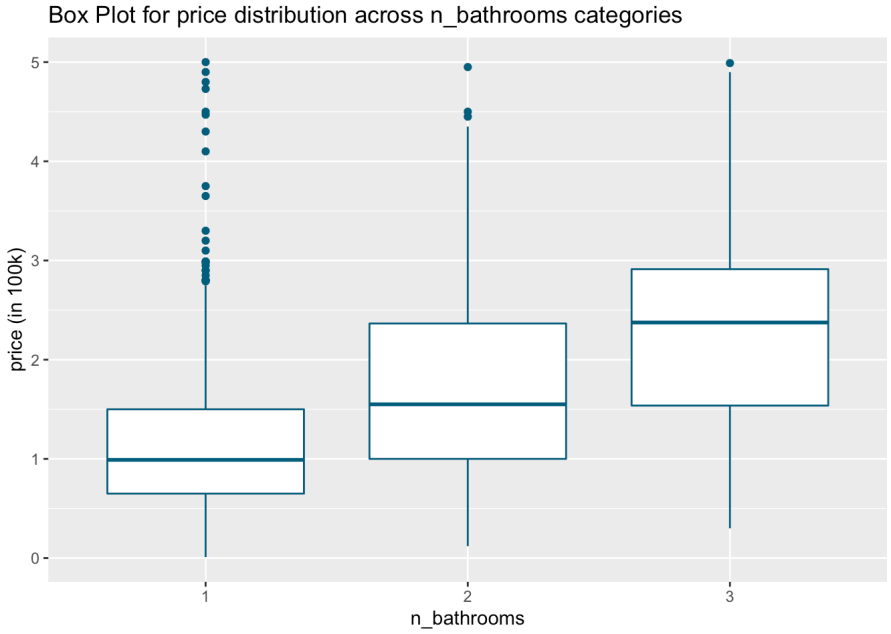

```
# Aggregate Mean, ANOVA Test, and Boxplot for mq vs n_rooms
compare.category(num_var = "mq", cat_var = "n_rooms")
```

```
## [1] "Aggregate Mean of mq between categories of n_rooms"
##   n_rooms        mq
## 1       2  68.49254
## 2       3  95.63559
## 3       4 134.70866
## 4       5 170.28571
##
##
## [1] "ANOVA Test for mq ~ n_rooms"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -129.29  -28.56  -10.49   10.33  669.71
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   68.493      5.845  11.718  < 2e-16 ***
## n_rooms3      27.143      6.863   3.955 8.26e-05 ***
## n_rooms4      66.216      7.224   9.166  < 2e-16 ***
## n_rooms5     101.793      7.912  12.865  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 67.66 on 899 degrees of freedom
## Multiple R-squared:  0.1974, Adjusted R-squared:  0.1948
## F-statistic: 73.72 on 3 and 899 DF,  p-value: < 2.2e-16
##
## [1] "Box Plot for mq vs n_rooms"
```



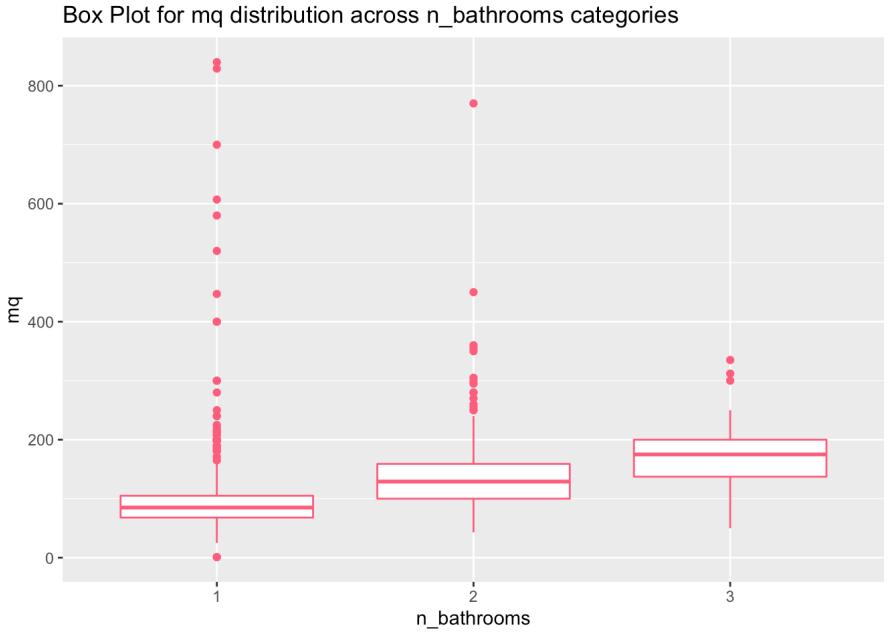

```
# Aggregate Mean, ANOVA Test, and Boxplot for price vs n_bathrooms
compare.category(num_var = "price", cat_var = "n_bathrooms")
```

```
## [1] "Aggregate Mean of price between categories of n_bathrooms"
##   n_bathrooms    price
## 1           1 118788.2
## 2           2 178284.4
## 3           3 241305.6
##
##
## [1] "ANOVA Test for price ~ n_bathrooms"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -211306  -58838  -19284   41212  381212
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    118788       3705  32.059  < 2e-16 ***
## n_bathrooms2    59496       6187   9.617  < 2e-16 ***
## n_bathrooms3   122517      15026   8.154 1.18e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 87370 on 900 degrees of freedom
## Multiple R-squared:  0.1336, Adjusted R-squared:  0.1317
## F-statistic: 69.42 on 2 and 900 DF,  p-value: < 2.2e-16
##
## [1] "Box Plot for price vs n_bathrooms"
```



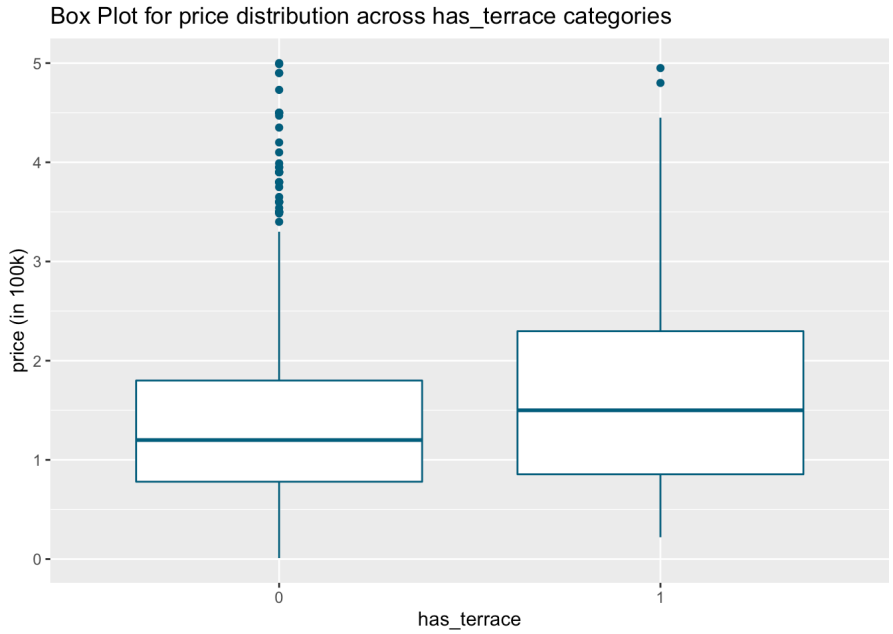

```
# Aggregate Mean, ANOVA Test, and Boxplot for mq vs n_bathrooms
compare.category(num_var = "mq", cat_var = "n_bathrooms")
```

```
## [1] "Aggregate Mean of mq between categories of n_bathrooms"
##   n_bathrooms        mq
## 1           1  99.49281
## 2           2 138.23473
## 3           3 176.55556
##
##
## [1] "ANOVA Test for mq ~ n_bathrooms"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -126.56  -34.49  -13.23   10.64  740.51
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    99.493      3.062  32.494  < 2e-16 ***
## n_bathrooms2   38.742      5.112   7.578 8.71e-14 ***
## n_bathrooms3   77.063     12.417   6.206 8.26e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 72.2 on 900 degrees of freedom
## Multiple R-squared:  0.08526,    Adjusted R-squared:  0.08323
## F-statistic: 41.94 on 2 and 900 DF,  p-value: < 2.2e-16
##
## [1] "Box Plot for mq vs n_bathrooms"
```



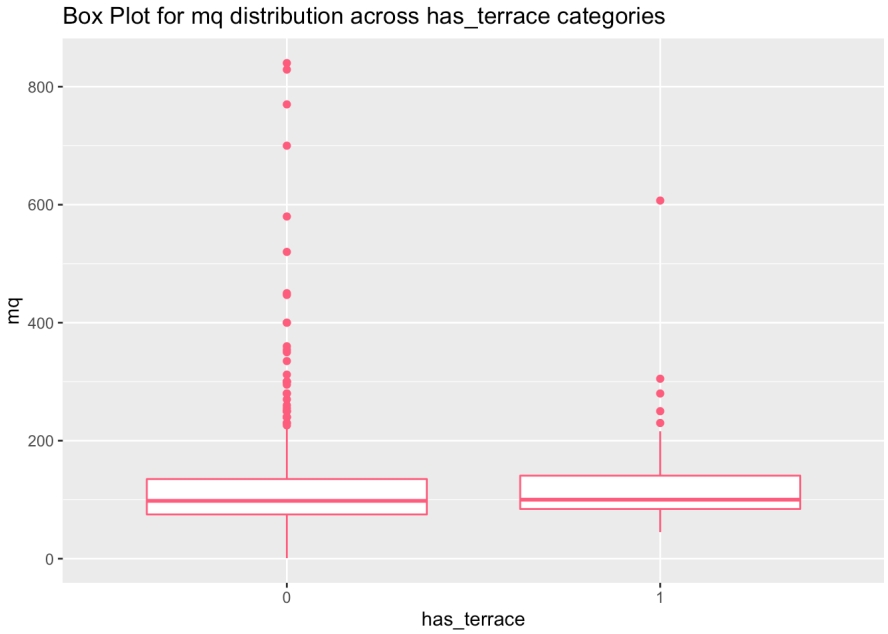

```
# Aggregate Mean, ANOVA Test, and Boxplot for price vs has_terrace
compare.category(num_var = "price", cat_var = "has_terrace")
```

```
## [1] "Aggregate Mean of price between categories of has_terrace"
##   has_terrace    price
## 1           0 140104.9
## 2           1 172253.9
##
##
## [1] "ANOVA Test for price ~ has_terrace"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -150254  -65105  -20105   44895  359895
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    140105       3318  42.224  < 2e-16 ***
## has_terrace1    32149       9339   3.443 0.000603 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93200 on 901 degrees of freedom
## Multiple R-squared:  0.01298,    Adjusted R-squared:  0.01189
## F-statistic: 11.85 on 1 and 901 DF,  p-value: 0.0006028
##
## [1] "Box Plot for price vs has_terrace"
```



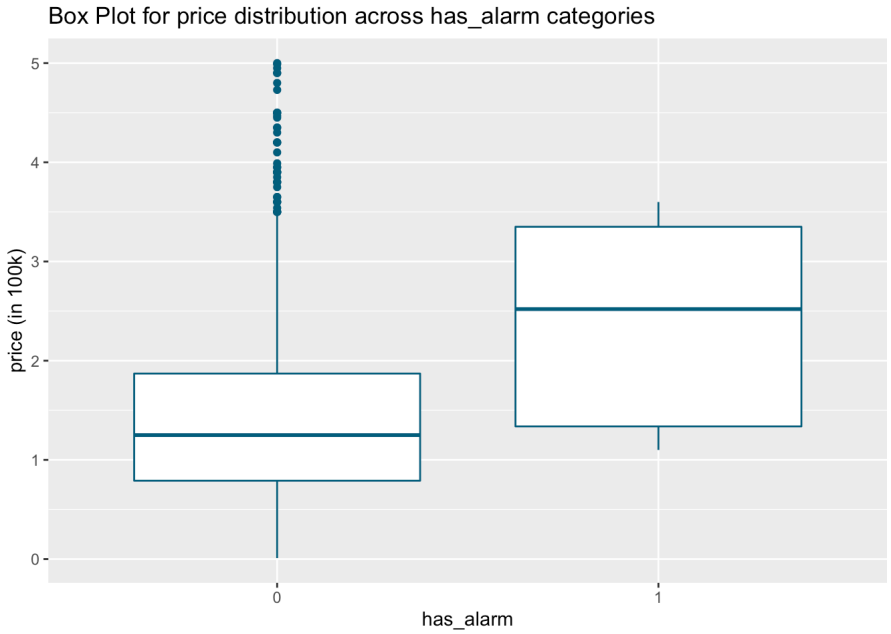

```
# Aggregate Mean, ANOVA Test, and Boxplot for mq vs n_rooms
compare.category(num_var = "mq", cat_var = "has_terrace")
```

```
## [1] "Aggregate Mean of mq between categories of has_terrace"
##   has_terrace       mq
## 1           0 115.2129
## 2           1 120.7193
##
##
## [1] "ANOVA Test for mq ~ has_terrace"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -114.21  -39.72  -17.72   19.79  724.79
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   115.213      2.685  42.907   <2e-16 ***
## has_terrace1    5.506      7.557   0.729    0.466
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.42 on 901 degrees of freedom
## Multiple R-squared:  0.0005889,  Adjusted R-squared:  -0.0005203
## F-statistic: 0.5309 on 1 and 901 DF,  p-value: 0.4664
##
## [1] "Box Plot for mq vs has_terrace"
```



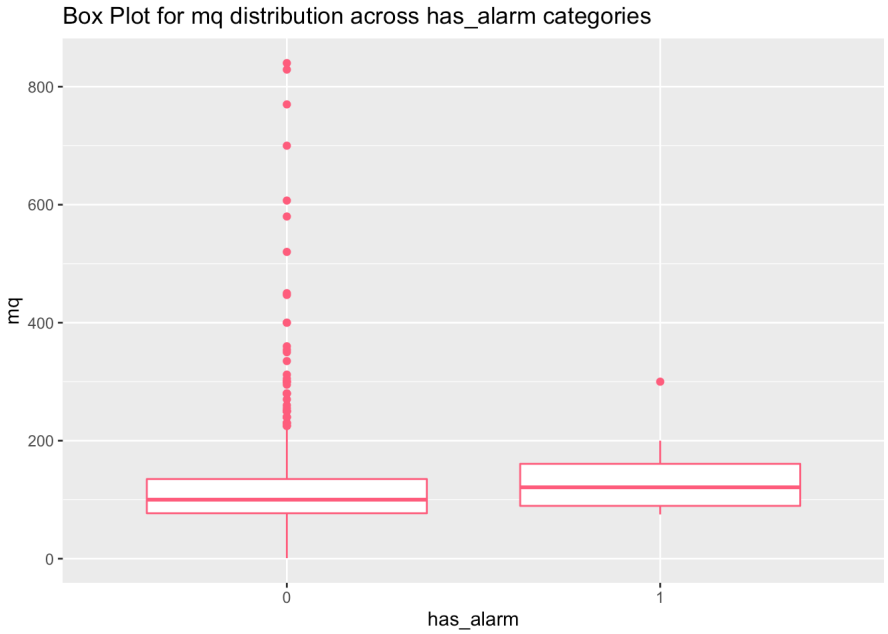

```
# Aggregate Mean, ANOVA Test, and Boxplot for price vs has_alarm
compare.category(num_var = "price", cat_var = "has_alarm")
```

```
## [1] "Aggregate Mean of price between categories of has_alarm"
##   has_alarm  price
## 1         0 143134
## 2         1 236100
##
##
## [1] "ANOVA Test for price ~ has_alarm"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -142134  -65134  -18134   43866  356866
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   143134       3122  45.840  < 2e-16 ***
## has_alarm1     92966      29672   3.133  0.00179 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93310 on 901 degrees of freedom
## Multiple R-squared:  0.01078,    Adjusted R-squared:  0.00968
## F-statistic: 9.817 on 1 and 901 DF,  p-value: 0.001785
##
## [1] "Box Plot for price vs has_alarm"
```



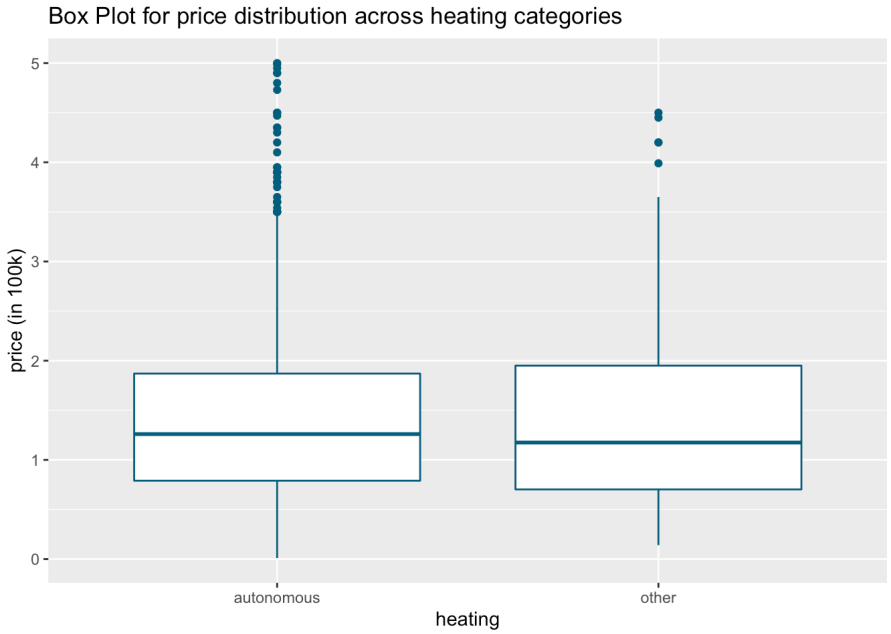

```
# Aggregate Mean, ANOVA Test, and Boxplot for mq vs has_alarm
compare.category(num_var = "mq", cat_var = "has_alarm")
```

```
## [1] "Aggregate Mean of mq between categories of has_alarm"
##   has_alarm       mq
## 1         0 115.6439
## 2         1 139.5000
##
##
## [1] "ANOVA Test for mq ~ has_alarm"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -114.64  -38.64  -15.64   19.36  724.36
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  115.644      2.523  45.830   <2e-16 ***
## has_alarm1    23.856     23.978   0.995     0.32
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.4 on 901 degrees of freedom
## Multiple R-squared:  0.001097,   Adjusted R-squared:  -1.126e-05
## F-statistic: 0.9898 on 1 and 901 DF,  p-value: 0.32
##
## [1] "Box Plot for mq vs has_alarm"
```



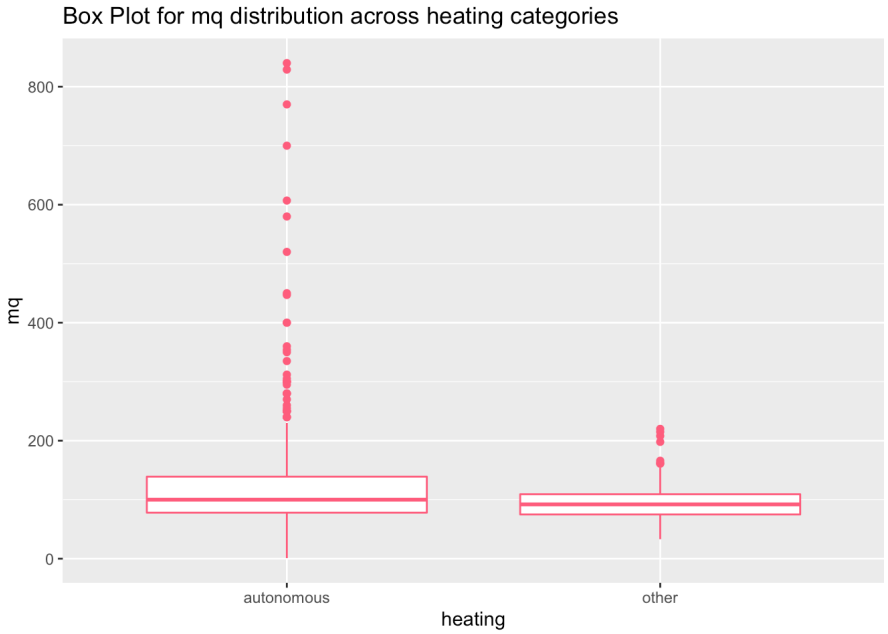

```
# Aggregate Mean, ANOVA Test, and Boxplot for price vs heating
compare.category(num_var = "price", cat_var = "heating")
```

```
## [1] "Aggregate Mean of price between categories of heating"
##      heating    price
## 1 autonomous 143514.2
## 2      other 148657.9
##
##
## [1] "ANOVA Test for price ~ heating"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -142514  -64514  -19658   43486  356486
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    143514       3339  42.976   <2e-16 ***
## heatingother     5144       9398   0.547    0.584
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93800 on 901 degrees of freedom
## Multiple R-squared:  0.0003323,  Adjusted R-squared:  -0.0007772
## F-statistic: 0.2995 on 1 and 901 DF,  p-value: 0.5843
##
## [1] "Box Plot for price vs heating"
```



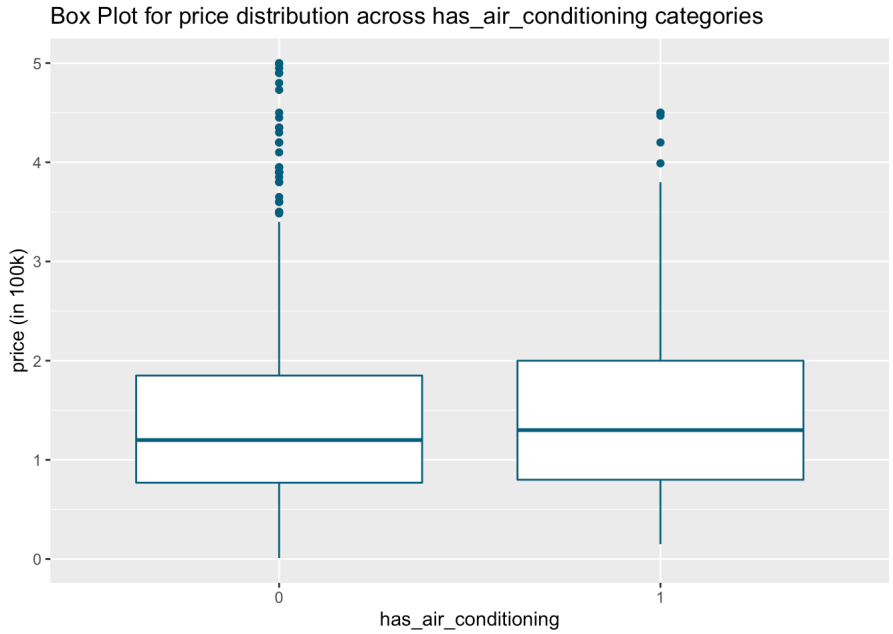

```
# Aggregate Mean, ANOVA Test, and Boxplot for mq vs heating
compare.category(num_var = "mq", cat_var = "heating")
```

```
## [1] "Aggregate Mean of mq between categories of heating"
##      heating        mq
## 1 autonomous 118.56654
## 2      other  97.50877
##
##
## [1] "ANOVA Test for mq ~ heating"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -117.57  -38.57  -18.57   18.43  721.43
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   118.567      2.674  44.335  < 2e-16 ***
## heatingother  -21.058      7.527  -2.798  0.00526 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.12 on 901 degrees of freedom
## Multiple R-squared:  0.008612,   Adjusted R-squared:  0.007512
## F-statistic: 7.827 on 1 and 901 DF,  p-value: 0.005257
##
## [1] "Box Plot for mq vs heating"
```



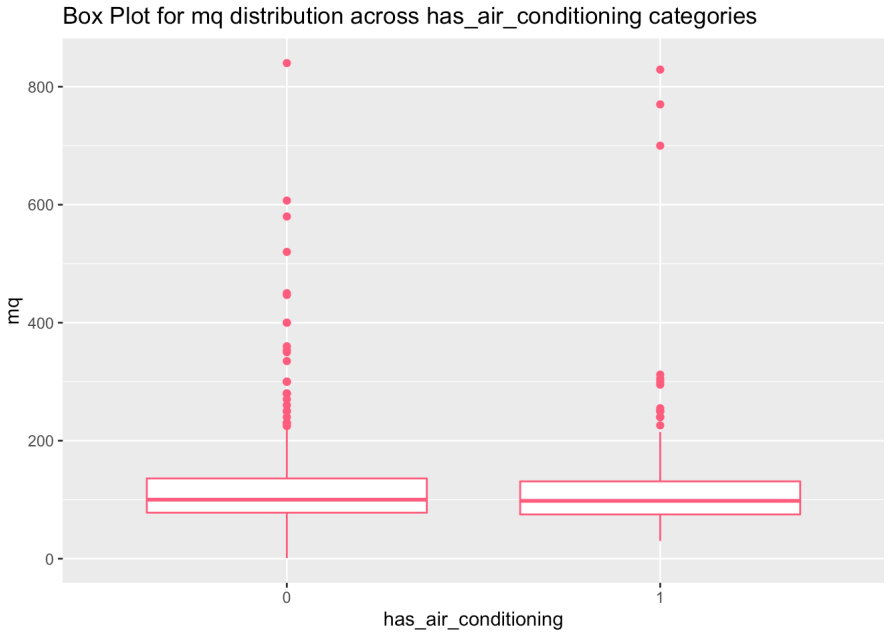

```
# Aggregate Mean, ANOVA Test, and Boxplot for price vs has_air_conditioning
compare.category(num_var = "price", cat_var = "has_air_conditioning")
```

```
## [1] "Aggregate Mean of price between categories of has_air_conditioning"
##   has_air_conditioning    price
## 1                    0 141823.5
## 2                    1 149109.8
##
##
## [1] "ANOVA Test for price ~ has_air_conditioning"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -140824  -66824  -21824   45176  358176
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             141824       3787   37.45   <2e-16 ***
## has_air_conditioning1     7286       6682    1.09    0.276
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93750 on 901 degrees of freedom
## Multiple R-squared:  0.001318,   Adjusted R-squared:  0.0002096
## F-statistic: 1.189 on 1 and 901 DF,  p-value: 0.2758
##
## [1] "Box Plot for price vs has_air_conditioning"
```



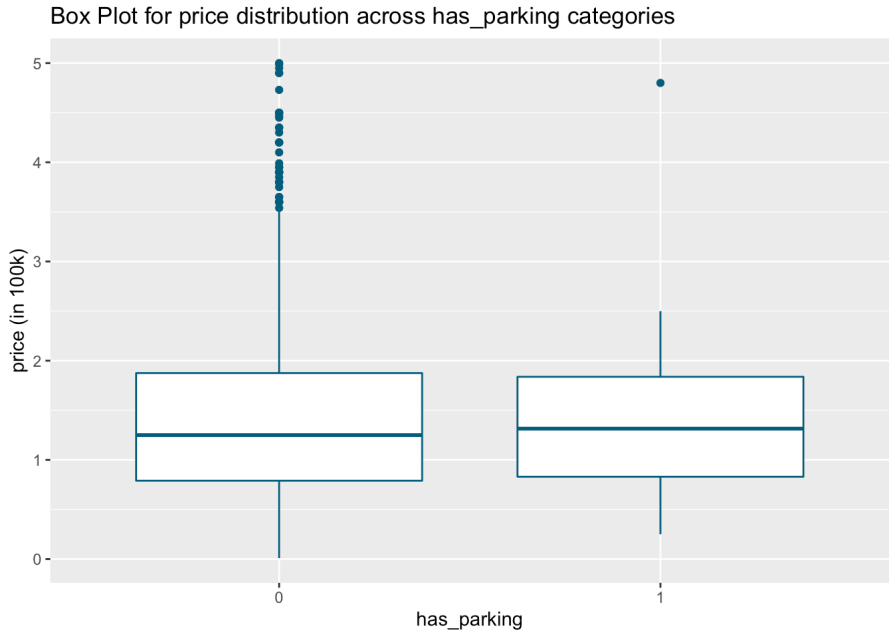

```
# Aggregate Mean, ANOVA Test, and Boxplot for mq vs has_air_conditioning
compare.category(num_var = "mq", cat_var = "has_air_conditioning")
```

```
## [1] "Aggregate Mean of mq between categories of has_air_conditioning"
##   has_air_conditioning       mq
## 1                    0 116.1811
## 2                    1 115.3310
##
##
## [1] "ANOVA Test for mq ~ has_air_conditioning"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -115.18  -38.18  -16.18   18.82  723.82
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            116.181      3.047  38.127   <2e-16 ***
## has_air_conditioning1   -0.850      5.377  -0.158    0.874
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.45 on 901 degrees of freedom
## Multiple R-squared:  2.774e-05,  Adjusted R-squared:  -0.001082
## F-statistic: 0.02499 on 1 and 901 DF,  p-value: 0.8744
##
## [1] "Box Plot for mq vs has_air_conditioning"
```



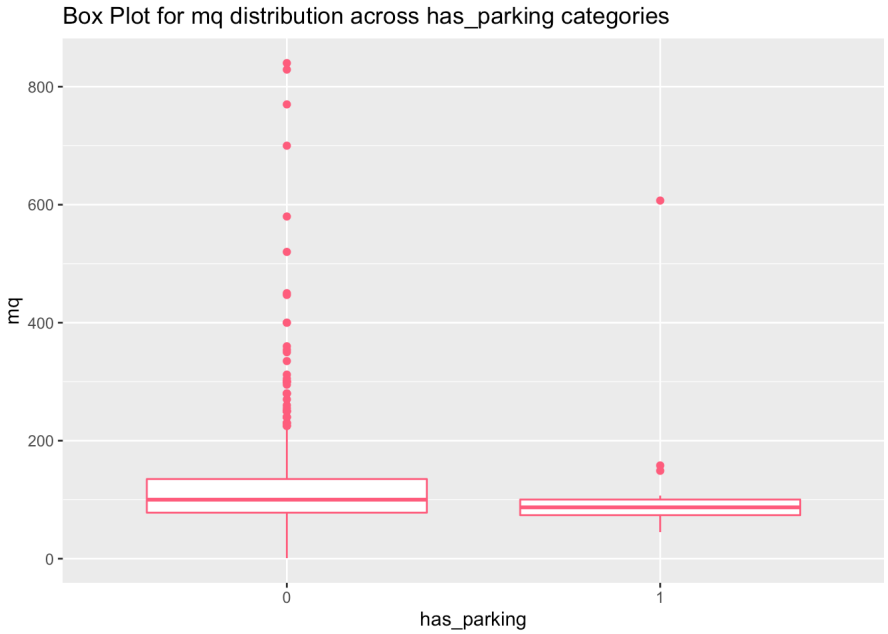

```
# Aggregate Mean, ANOVA Test, and Boxplot for price vs has_parking
compare.category(num_var = "price", cat_var = "has_parking")
```

```
## [1] "Aggregate Mean of price between categories of has_parking"
##   has_parking    price
## 1           0 144110.1
## 2           1 147125.0
##
##
## [1] "ANOVA Test for price ~ has_parking"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -143110  -65110  -19110   43390  355890
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    144110       3150  45.749   <2e-16 ***
## has_parking1     3015      23664   0.127    0.899
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93810 on 901 degrees of freedom
## Multiple R-squared:  1.801e-05,  Adjusted R-squared:  -0.001092
## F-statistic: 0.01623 on 1 and 901 DF,  p-value: 0.8987
##
## [1] "Box Plot for price vs has_parking"
```



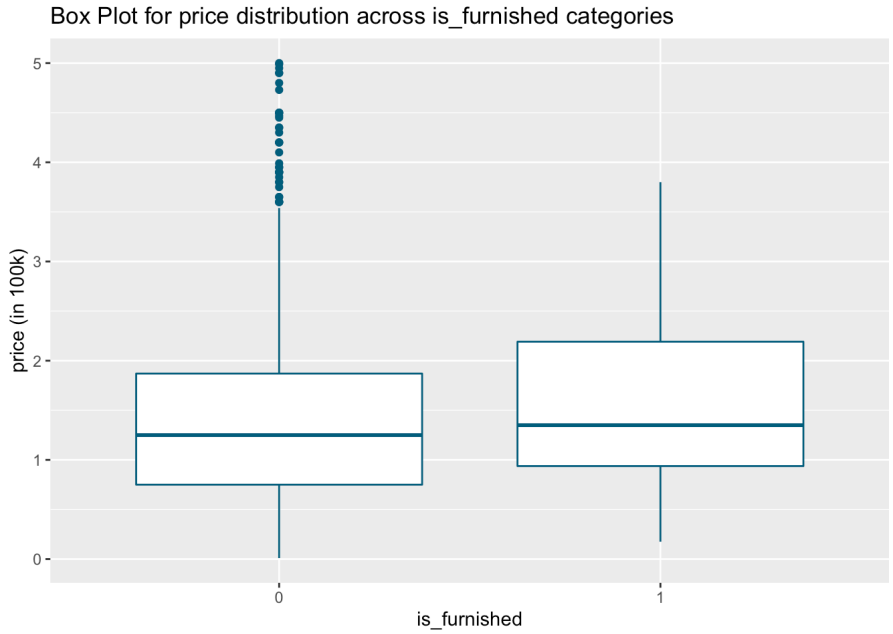

```
# Aggregate Mean, ANOVA Test, and Boxplot for mq vs has_parking
compare.category(num_var = "mq", cat_var = "has_parking")
```

```
## [1] "Aggregate Mean of mq between categories of has_parking"
##   has_parking       mq
## 1           0 115.7813
## 2           1 122.9375
##
##
## [1] "ANOVA Test for mq ~ has_parking"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -114.78  -38.78  -15.78   19.22  724.22
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   115.781      2.533  45.709   <2e-16 ***
## has_parking1    7.156     19.029   0.376    0.707
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.44 on 901 degrees of freedom
## Multiple R-squared:  0.0001569,  Adjusted R-squared:  -0.0009528
## F-statistic: 0.1414 on 1 and 901 DF,  p-value: 0.707
##
## [1] "Box Plot for mq vs has_parking"
```



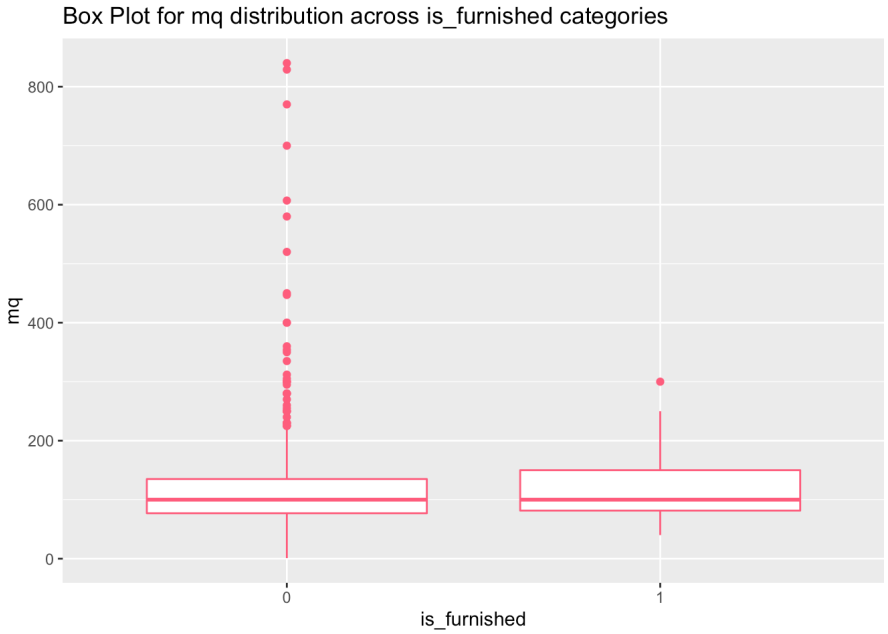

```
# Aggregate Mean, ANOVA Test, and Boxplot for price vs is_furnished
compare.category(num_var = "price", cat_var = "is_furnished")
```

```
## [1] "Aggregate Mean of price between categories of is_furnished"
##   is_furnished    price
## 1            0 143157.3
## 2            1 155777.8
##
##
## [1] "ANOVA Test for price ~ is_furnished"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -142157  -66778  -20778   43843  356843
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     143157       3252  44.018   <2e-16 ***
## is_furnished1    12620      11518   1.096    0.273
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93750 on 901 degrees of freedom
## Multiple R-squared:  0.001331,   Adjusted R-squared:  0.0002224
## F-statistic: 1.201 on 1 and 901 DF,  p-value: 0.2735
##
## [1] "Box Plot for price vs is_furnished"
```




```
# Aggregate Mean, ANOVA Test, and Boxplot for mq vs is_furnished
compare.category(num_var = "mq", cat_var = "is_furnished")
```

```
## [1] "Aggregate Mean of mq between categories of is_furnished"
##   is_furnished       mq
## 1            0 115.7665
## 2            1 117.5417
##
##
## [1] "ANOVA Test for mq ~ is_furnished"
##
## Call:
## aov(formula = compare.formula, data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -114.77  -38.27  -17.54   19.23  724.23
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    115.767      2.617  44.234   <2e-16 ***
## is_furnished1    1.775      9.268   0.192    0.848
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 75.44 on 901 degrees of freedom
## Multiple R-squared:  4.071e-05,  Adjusted R-squared:  -0.001069
## F-statistic: 0.03668 on 1 and 901 DF,  p-value: 0.8482
##
## [1] "Box Plot for mq vs is_furnished"
```



2.3 Additional insights and issues

Some insights and potential issues from the above analysis are:

1. The skewed distribution of *price* and *mq* may suggest the presence of extreme or abnormal values i.e outliers that could be influencing the overall trend.
2. The significant dependence between certain pairs of categorical variables, as indicated by the Chi-Square test, may indicate a relationship between these variables, and could potentially allow for predictions to be made about one variable based on the values of the other.
3. The significant effect of the number of rooms and bathrooms on both the price and size of a house indicates the importance of these factors in influencing the value and area of the property.
4. The significant differences in price between houses with and without certain features (such as terrace or alarm) indicates that these features are highly valued by buyers and could be used as selling points. Meanwhile, the significant differences in area for different heating groups indicate that heating may determinine property area.
5. The lack of significant differences or meaningful trends in price or size across groups of *floor*, *has_parking*, *has_air_conditioning* and *is_furnished* indicates that these variables are not as influential on the overall price or size of a property.

3. Modelling

3.1 Explain your analysis plan

Given the research question regarding property price, an analysis plan would include the following steps:

1. **Model Selection**: Since the dependent variable is numerical, and the explanatory variables are a mix of categorical and numerical, we will use ANCOVA or multiple linear regression model for our research question.
2. **Variable Selection**: We'll start with including variables that have a significant effect on the property price such as *mq*, *n_bathrooms*, *has_terrace*, *has_alarm* and *heating*. An alternate way that we're following in our modelling would be to build a maximal model that incorporates all the variables in our dataset as it allows us to determine maximum potential performance for our model.
3. **Variable Augmentation**: Next, we'll add additional variables to the model through transformation and interactions, as these new variables would capture non-linear relationships or higher order interactions that the original variables did not capture. This would also help with reduce the skewness of non-normally distributed variables like *mq*.
4. **Fitting the model and evaluating the performance**: We'll fit the model using the selected variables and evaluate the performance through metrics such as R-squared, significant columns (p-value < 0.05), residuals and diagnostic plots.
5. **Updating the model**: Once we check the performance of the model, we'll update the model to remove non-significant variables while retaining the significant ones. We use **step**() function for this as it updates the model in one go and then evaluate the performance of the significant model. We'll also check multicollinearity between variables using *vif*() method (For more information on calculating variable inflation factor, see the following resource: [@vif]).
6. **Interpreting model coefficients**: Finally, we'll analyse the coefficients to understand the impact of each variable on the property price.

3.2 Build a model for property price

```
# Building Maximal Model
price.max.model <- lm(formula = price ~ ., data = housedf_clean)

# Summary of the model
summary(price.max.model)
```

```
##
## Call:
## lm(formula = price ~ ., data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -183023  -56510  -14271   40666  389396
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.436e+04  1.005e+04   6.401 2.51e-10 ***
## id                      4.831e-02  4.360e-02   1.108  0.26812
## mq                      2.816e+02  4.204e+01   6.699 3.74e-11 ***
## floor2                 -3.613e+03  6.331e+03  -0.571  0.56832
## floor3                 -1.364e+04  9.142e+03  -1.492  0.13593
## floor4                  8.035e+03  1.657e+04   0.485  0.62791
## floor5                  1.802e+04  2.386e+04   0.755  0.45038
## floor6                  4.822e+04  4.916e+04   0.981  0.32696
## floor7                  2.999e+04  3.803e+04   0.789  0.43061
## n_rooms3                2.514e+04  8.726e+03   2.882  0.00405 **
## n_rooms4                1.889e+04  9.778e+03   1.932  0.05365 .
## n_rooms5                1.791e+04  1.120e+04   1.598  0.11031
## n_bathrooms2            4.581e+04  6.554e+03   6.989 5.46e-12 ***
## n_bathrooms3            1.007e+05  1.517e+04   6.639 5.51e-11 ***
## has_terrace1            2.071e+04  8.734e+03   2.372  0.01791 *
## has_alarm1              6.488e+04  2.710e+04   2.394  0.01686 *
## heatingother            1.679e+04  8.696e+03   1.931  0.05382 .
## has_air_conditioning1 -1.776e+03  6.284e+03  -0.283  0.77760
## has_parking1            5.230e+03  2.160e+04   0.242  0.80876
## is_furnished1           7.953e+03  1.057e+04   0.752  0.45207
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 84060 on 883 degrees of freedom
## Multiple R-squared:  0.2132, Adjusted R-squared:  0.1963
## F-statistic: 12.59 on 19 and 883 DF,  p-value: < 2.2e-16
```

```
# Updating the model using transformation of mq variable
price.max.model <- update(price.max.model, . ~ . + log(mq))

# Summary of the updated model
summary(price.max.model)
```

```
##
## Call:
## lm(formula = price ~ id + mq + floor + n_rooms + n_bathrooms +
##     has_terrace + has_alarm + heating + has_air_conditioning +
##     has_parking + is_furnished + log(mq), data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -184113  -57482  -14697   40618  389728
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.914e+03  4.281e+04   0.091  0.92718
## id                      4.585e-02  4.360e-02   1.051  0.29335
## mq                      2.031e+02  6.842e+01   2.969  0.00307 **
## floor2                 -4.068e+03  6.334e+03  -0.642  0.52090
## floor3                 -1.397e+04  9.139e+03  -1.529  0.12668
## floor4                  7.140e+03  1.657e+04   0.431  0.66672
## floor5                  1.764e+04  2.385e+04   0.740  0.45967
## floor6                  4.711e+04  4.914e+04   0.959  0.33800
## floor7                  2.840e+04  3.802e+04   0.747  0.45534
## n_rooms3                2.152e+04  9.070e+03   2.373  0.01784 *
## n_rooms4                1.362e+04  1.042e+04   1.307  0.19157
## n_rooms5                1.246e+04  1.181e+04   1.056  0.29139
## n_bathrooms2            4.380e+04  6.695e+03   6.542 1.03e-10 ***
## n_bathrooms3            9.772e+04  1.530e+04   6.388 2.72e-10 ***
## has_terrace1            2.027e+04  8.733e+03   2.322  0.02048 *
## has_alarm1              6.448e+04  2.708e+04   2.381  0.01749 *
## heatingother            1.653e+04  8.693e+03   1.902  0.05755 .
## has_air_conditioning1 -1.528e+03  6.283e+03  -0.243  0.80794
## has_parking1            6.357e+03  2.161e+04   0.294  0.76865
## is_furnished1           7.574e+03  1.057e+04   0.717  0.47377
## log(mq)                 1.617e+04  1.114e+04   1.452  0.14674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 84010 on 882 degrees of freedom
## Multiple R-squared:  0.2151, Adjusted R-squared:  0.1973
## F-statistic: 12.08 on 20 and 882 DF,  p-value: < 2.2e-16
```

```
# Updating the model using step() method
price.model <- step(price.max.model)
```

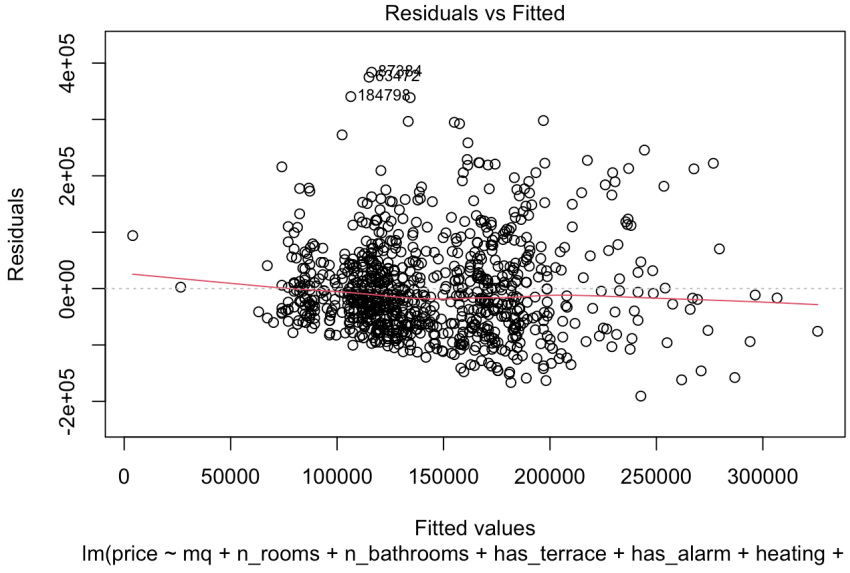
```
## Start:  AIC=20498.36
## price ~ id + mq + floor + n_rooms + n_bathrooms + has_terrace +
##     has_alarm + heating + has_air_conditioning + has_parking +
##     is_furnished + log(mq)
##
##                        Df  Sum of Sq        RSS   AIC
## - floor                 6 3.7057e+10 6.2614e+12 20492
## - has_air_conditioning  1 4.1726e+08 6.2248e+12 20496
## - has_parking           1 6.1094e+08 6.2250e+12 20496
## - is_furnished          1 3.6246e+09 6.2280e+12 20497
## - id                    1 7.8016e+09 6.2322e+12 20498
## <none>                               6.2244e+12 20498
## - log(mq)               1 1.4887e+10 6.2393e+12 20498
## - n_rooms               3 4.3513e+10 6.2679e+12 20499
## - heating               1 2.5519e+10 6.2499e+12 20500
## - has_terrace           1 3.8036e+10 6.2624e+12 20502
## - has_alarm             1 3.9996e+10 6.2644e+12 20502
## - mq                    1 6.2214e+10 6.2866e+12 20505
## - n_bathrooms           2 4.6995e+11 6.6943e+12 20560
##
## Step:  AIC=20491.72
## price ~ id + mq + n_rooms + n_bathrooms + has_terrace + has_alarm +
##     heating + has_air_conditioning + has_parking + is_furnished +
##     log(mq)
##
##                        Df  Sum of Sq        RSS   AIC
## - has_parking           1 2.2217e+08 6.2617e+12 20490
## - has_air_conditioning  1 2.5584e+08 6.2617e+12 20490
## - is_furnished          1 5.5116e+09 6.2670e+12 20490
## - id                    1 8.9765e+09 6.2704e+12 20491
## <none>                               6.2614e+12 20492
## - log(mq)               1 1.5108e+10 6.2766e+12 20492
## - n_rooms               3 4.6168e+10 6.3076e+12 20492
## - heating               1 2.7413e+10 6.2889e+12 20494
## - has_alarm             1 3.8856e+10 6.3003e+12 20495
## - has_terrace           1 4.0446e+10 6.3019e+12 20496
```

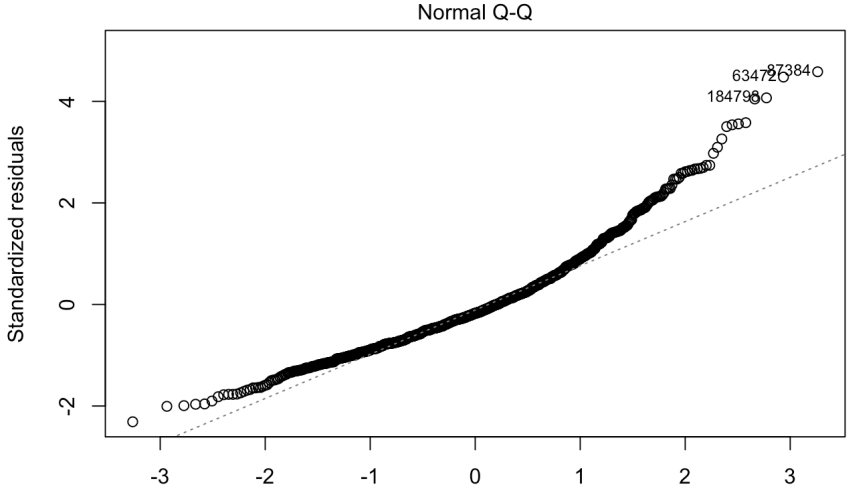
```
## - mq                    1 6.2864e+10 6.3243e+12 20499
## - n_bathrooms           2 4.6637e+11 6.7278e+12 20553
##
## Step:  AIC=20489.75
## price ~ id + mq + n_rooms + n_bathrooms + has_terrace + has_alarm +
##     heating + has_air_conditioning + is_furnished + log(mq)
##
##                        Df  Sum of Sq        RSS   AIC
## - has_air_conditioning  1 2.0815e+08 6.2619e+12 20488
## - is_furnished          1 5.4513e+09 6.2671e+12 20488
## - id                    1 9.0811e+09 6.2708e+12 20489
## <none>                               6.2617e+12 20490
## - log(mq)               1 1.4991e+10 6.2767e+12 20490
## - n_rooms               3 4.6114e+10 6.3078e+12 20490
## - heating               1 2.7683e+10 6.2894e+12 20492
## - has_alarm             1 3.8741e+10 6.3004e+12 20493
## - has_terrace           1 4.1304e+10 6.3030e+12 20494
## - mq                    1 6.3308e+10 6.3250e+12 20497
## - n_bathrooms           2 4.6661e+11 6.7283e+12 20551
##
## Step:  AIC=20487.78
## price ~ id + mq + n_rooms + n_bathrooms + has_terrace + has_alarm +
##     heating + is_furnished + log(mq)
##
##                Df  Sum of Sq        RSS   AIC
## - is_furnished  1 5.2622e+09 6.2671e+12 20486
## - id            1 9.1197e+09 6.2710e+12 20487
## <none>                       6.2619e+12 20488
## - log(mq)       1 1.5125e+10 6.2770e+12 20488
## - n_rooms       3 4.5974e+10 6.3079e+12 20488
## - heating       1 2.7520e+10 6.2894e+12 20490
## - has_alarm     1 3.8587e+10 6.3005e+12 20491
## - has_terrace   1 4.1516e+10 6.3034e+12 20492
## - mq            1 6.3196e+10 6.3251e+12 20495
## - n_bathrooms   2 4.6654e+11 6.7284e+12 20549
##
## Step:  AIC=20486.54
## price ~ id + mq + n_rooms + n_bathrooms + has_terrace + has_alarm +
##     heating + log(mq)
##
##               Df  Sum of Sq        RSS   AIC
## - id           1 8.8659e+09 6.2760e+12 20486
## <none>                      6.2671e+12 20486
## - log(mq)      1 1.5465e+10 6.2826e+12 20487
## - n_rooms      3 4.7527e+10 6.3147e+12 20487
## - heating      1 2.6078e+10 6.2932e+12 20488
## - has_alarm    1 3.8945e+10 6.3061e+12 20490
## - has_terrace  1 4.4244e+10 6.3114e+12 20491
## - mq           1 6.2420e+10 6.3296e+12 20494
## - n_bathrooms  2 4.6413e+11 6.7313e+12 20547
##
## Step:  AIC=20485.82
## price ~ mq + n_rooms + n_bathrooms + has_terrace + has_alarm +
##     heating + log(mq)
##
##               Df  Sum of Sq        RSS   AIC
## <none>                      6.2760e+12 20486
## - log(mq)      1 1.6402e+10 6.2924e+12 20486
## - n_rooms      3 4.8658e+10 6.3247e+12 20487
## - heating      1 2.6919e+10 6.3029e+12 20488
## - has_alarm    1 3.9978e+10 6.3160e+12 20490
## - has_terrace  1 4.4969e+10 6.3210e+12 20490
## - mq           1 5.9495e+10 6.3355e+12 20492
## - n_bathrooms  2 4.5917e+11 6.7352e+12 20546
```

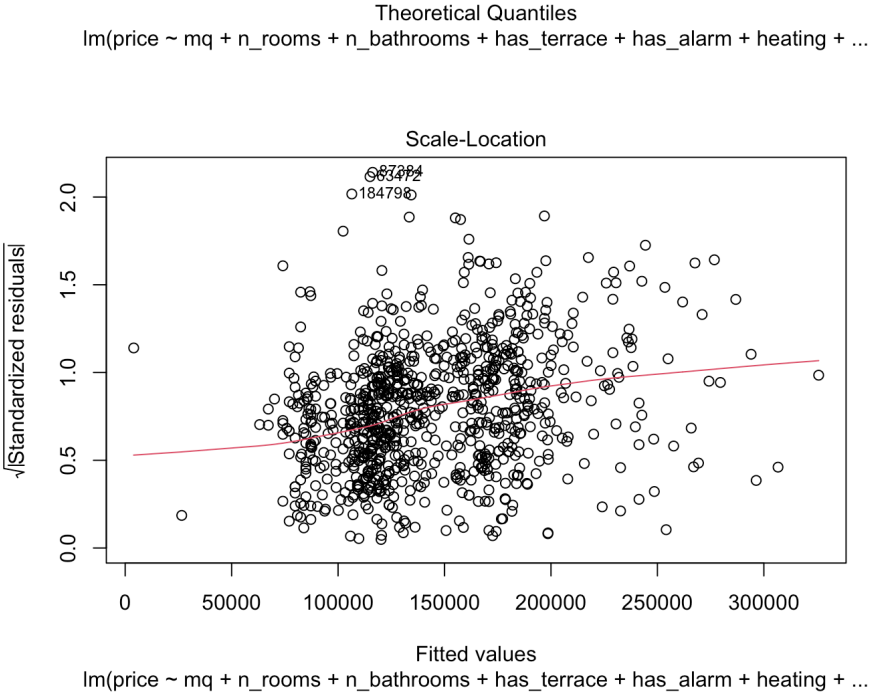
```
# Summary of the model
summary(price.model)
```

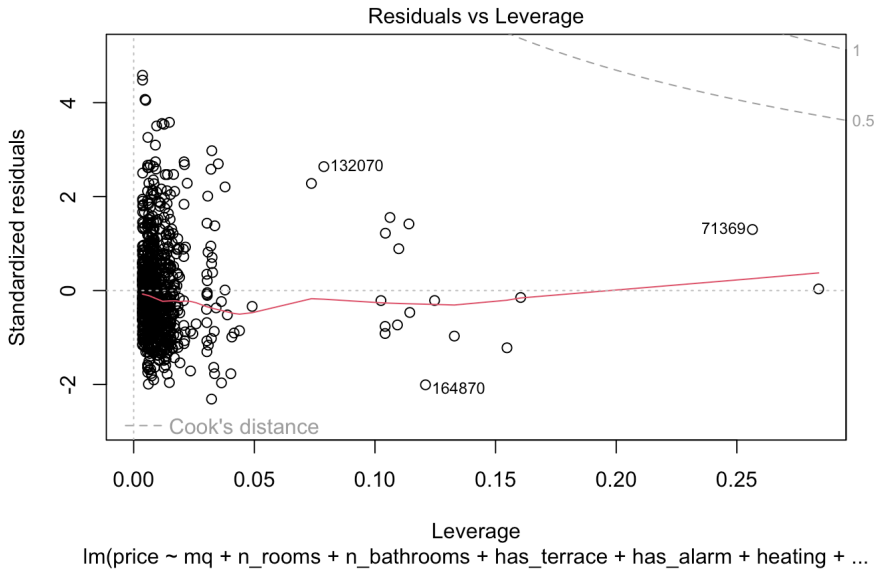
```
##
## Call:
## lm(formula = price ~ mq + n_rooms + n_bathrooms + has_terrace +
##     has_alarm + heating + log(mq), data = housedf_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -190684  -58003  -14541   39935  383769
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3807.50   42428.86   0.090  0.92852
## mq             197.58      67.95   2.908  0.00373 **
## n_rooms3     22546.44    8996.05   2.506  0.01238 *
## n_rooms4     14134.88   10338.57   1.367  0.17191
## n_rooms5     12803.13   11731.20   1.091  0.27540
## n_bathrooms2 43308.89    6659.28   6.504 1.31e-10 ***
## n_bathrooms3 95665.78   15228.22   6.282 5.21e-10 ***
## has_terrace1 21427.56    8475.73   2.528  0.01164 *
## has_alarm1   63979.13   26840.19   2.384  0.01735 *
## heatingother 16574.03    8473.47   1.956  0.05078 .
## log(mq)      16903.30   11070.92   1.527  0.12716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83880 on 892 degrees of freedom
## Multiple R-squared:  0.2086, Adjusted R-squared:  0.1997
## F-statistic: 23.51 on 10 and 892 DF,  p-value: < 2.2e-16
```

```
# Diagnostic plots for the final model
plot(price.model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(price ~ mq + n_rooms + n_bathrooms + has_terrace + has_alarm + heating + ...
87384
634722
184798
4e+05
2e+05
0e+00
-2e+05
0
50000
100000
150000
200000
250000
300000


Normal Q-Q
Standardized residuals
87384
634722
184798
4
2
0
-2
-3
-2
-1
0
1
2
3






```
# Checking multi-collinearity in the model parameters
# No multi-collinearity as the value of VIF in the output is almost equal to 1
vif(price.model)
```

```
##                GVIF Df GVIF^(1/(2*Df))
## mq          3.365315  1        1.834480
## n_rooms     1.650749  3        1.087127
## n_bathrooms 1.340342  2        1.075979
## has_terrace 1.017021  1        1.008475
## has_alarm   1.012550  1        1.006255
## heating     1.016480  1        1.008206
## log(mq)     4.132373  1        2.032824
```

The linear regression equation from our significant model is

$$price = 197.58 * mq + 22546.44 * n_rooms3 + 14134.88 * n_rooms4 + 12803.13 * n_rooms5 + 43308.89 * n_bathrooms2 + 95665.78 * n_bathrooms3 + 1$$

3.3 Critique model using relevant diagnostics

The above is a linear regression model to predict the price of the property based on factors listed in the coefficients. We are using step function to remove insignificant dependent variables from our model. On the basis of the final model summary, the following observations can be made:

1. In our final model, the independent variables influencing price in a significant way are *mq, n_rooms, n_bathrooms, has_terrace, has_alarm & heating*. This is indicated by the p-value of the corresponding coefficients being less than 0.05. The variable *log(mq)* has a p-value of 0.1276 > 0.05, implying the relationship between log(mq) and price is not statistically significant at 0.05 level. However, this doesn't rule out any affect of the variable on the response variable price.
2. The R-squared value that represents the goodness of fit is 0.2086, which means the model explains about 21% of the variance in the response variable, price.
3. The residuals which represent the difference between the observed values for price and the predicted values from the model, have a large overall range (from -190684 to 383769) suggesting that the model is not fitting the data accurately.
4. The overall p-value for the model is < 2.2e-16, indicating the model is statistically significant.
5. The *Residual vs Fitted plot* and the upward slope in *Scale-Location plot* indicate heteroscedasticity i.e. the variance of the residuals is not constant. The distribution of residuals is relatively concentrated around the lower fitted values, and gets increasingly spread out till around 200000, after which the distribution of observations decreases.
6. The banana-shaped curve on the *normal Q-Q plot* indicates the residuals are not normally distributed and the distribution is skewed.
7. The *Residuals vs Leverage* plot shows some outliers but no highly influential case beyond the cook's distance.
8. There is no evidence of high multicollinearity between the independent variables as the value of Variance Inflation Factor (VIF) for each variable nears 1.

Potential weaknesses of the model include

1. The relatively low R-squared value at 0.2086, indicating the model explains only 21% of the variance in price, which may not be enough to make accurate predictions. This could also be an indication of other factors influencing the price of the property that were not originally mentioned in the provided data.
2. The residuals have a large range, suggesting that the model is not accurately fitting the data. This may be caused by outliers, nonlinear relationships, or other factors that the model is unable to capture.
3. The presence of heteroscedasticity and influential outliers could cause problems with the model fit and may lead to biased / inaccurate parameter estimates.
4. As indicated by the banana-curve on the QQ plot, the residuals have a skewed distribution i.e the errors are not normally distributed. This could cause problems for models requiring normally distributed errors.
5. Based on the above mentioned weaknesses, it appears that the linear regression model may not be a good choice for modeling the data, despite being a statistically significant model for predicting the price of the property.

3.4 Suggest improvements to your model

Based on the above weakness, the following improvements are recommended:

1. To improve the model's fit and increase the R-squared value, we could consider adding more variables to the model, through transformation or interactions. This would help capture additional relationships in the data and increase the model's explanatory power.
2. To address the issue of residuals having a large range, it may be necessary to perform further data cleaning to remove outliers and unusual values (such as *mq* being 1), and merging categories with lower frequency, for e.g. merging the observation with *floors* 4,5, and 6 into a single category.
3. To address the concerns regarding the skewed distribution for dependent variable *price* and independent variables such as *mq*, we could try further transforming them.
4. To address influential outliers and heteroscedasticity, we could try removing those outliers and use models that are less susceptible to outliers and non-constant variances.
5. Since the model fit of the linear regression model is low, the use of non-linear model such as polynomial regression etc. could potentially improve model fit and accuracy.

4. Extension work

4.1 Model the likelihood of a property being furnished (using the is_furnished variable provided).

An analysis plan for modelling the likelihood of property being furnished is:

1. **Model Selection**: Since the dependent variable is binary, and the explanatory variables are a mix of categorical and numerical, we will use logistic regression.
2. **Building Maximal Model**: We'll build a maximal model by including all the variables in our model as it allows us to determine the model's maximum potential performance.
3. **Evaluating Model significance**: Once we fit the model, we check for the presence of non-significant variables (p-value < 0.05) in our model. If no non-significant variables are present, our model is significant.
4. **Model Updation**: Next, we'll update the model to remove non-significant variables while retaining the significant ones. We use **step**() function for this as it updates the model in one go and then evaluate the performance of the significant model.

5. **Performance Evaluation**: From the model, we'll calculate the chi-squared statistic and the predictor degree of freedom, which will give us the p-value, from which we can check if the model has a good fit or not (refer [@GLM]). Next, we'll predict the likelihood of property being furnished, and generate the confusion matrix (refer [@ConfMatrix]) then evaluate model performance through metrics of accuracy, precision and recall.
6. **Interpreting model coefficients**: Finally, we'll analyse the odds ratio to understand the effect of a unit change in the dependent variable on the odds of property being furnished.

In the final model below, the following observations are made:

1. The model has a relatively high residual deviance and a p-value of 0.99 (>0.05) generated from the chi-square statistic, indicating model is not good fit to the data.
2. The model has a very high accuracy of 92%, and the precision and recall are NaN (0/0) and 0 respectively.
3. The variable influencing is_furnished significantly is *has_air_conditioning=1* (p-value < 0.05), while *heating=other* and *has_terrace*=1 have p-value>0.05.
4. From the odds ratio, we observe *has_terrace*=1 and *has_air_conditioning*=1 increase the likelihood of property being furnished by factors of 1.66 and 2.77 respectively, while *heating=other* decreases the likelihood by a factor of 0.36.

The issues in the model are:

1. The model does not fit the data very well as evident from the high residual deviance and chi-sq p-value. This indicates that the variables included in the model lack explanatory power in predicting *is_furnished*.
2. The precision and recall values indicate the model is not able to accurately classify the positive cases.
3. The p-value for the variables "heating=other" and "has_terrace=1" is greater than 0.05, indicating that they may not have a strong effect on *is_furnished*.

```
# Model building
furnished.max.model <- glm(formula = is_furnished ~ ., data = housedf_clean, family = "binomial")

# Summary of the model
summary(furnished.max.model)
```

```
##
## Call:
## glm(formula = is_furnished ~ ., family = "binomial", data = housedf_clean)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3262  -0.4169  -0.3425  -0.2425   3.1676
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.382e+00  5.709e-01  -5.924 3.14e-09 ***
## id                    -1.068e-06  2.001e-06  -0.533   0.5938
## price                  1.240e-06  1.475e-06   0.841   0.4003
## mq                    -6.599e-04  1.876e-03  -0.352   0.7250
## floor2                -4.699e-01  2.969e-01  -1.583   0.1135
## floor3                -1.007e+00  5.433e-01  -1.853   0.0639 .
## floor4                 4.797e-04  7.729e-01   0.001   0.9995
## floor5                 4.257e-01  8.231e-01   0.517   0.6050
## floor6                 1.297e+00  1.401e+00   0.926   0.3544
## floor7                -1.350e+01  6.271e+02  -0.022   0.9828
## n_rooms3               8.818e-01  5.098e-01   1.730   0.0837 .
## n_rooms4               9.132e-01  5.454e-01   1.674   0.0941 .
## n_rooms5               9.228e-01  6.001e-01   1.538   0.1241
## n_bathrooms2          -3.725e-02  2.965e-01  -0.126   0.9000
## n_bathrooms3          -1.412e+00  1.066e+00  -1.324   0.1855
## has_terrace1           3.825e-01  3.315e-01   1.154   0.2486
## has_alarm1            -2.238e-01  1.113e+00  -0.201   0.8407
## heatingother          -1.048e+00  5.420e-01  -1.933   0.0532 .
## has_air_conditioning1  1.057e+00  2.627e-01   4.022 5.78e-05 ***
## has_parking1          -7.095e-01  1.071e+00  -0.663   0.5076
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 502.28  on 902  degrees of freedom
## Residual deviance: 459.98  on 883  degrees of freedom
## AIC: 499.98
##
## Number of Fisher Scoring iterations: 14
```

```
# Updating model using step() to remove non-significant terms
furnished.model <- step(furnished.max.model)
```

```
## Start:  AIC=499.98
## is_furnished ~ id + price + mq + floor + n_rooms + n_bathrooms +
##     has_terrace + has_alarm + heating + has_air_conditioning +
##     has_parking
##
##                        Df Deviance    AIC
## - floor                 6   468.41 496.41
## - n_rooms               3   463.85 497.85
## - has_alarm             1   460.02 498.02
## - mq                    1   460.11 498.11
## - id                    1   460.26 498.26
## - has_parking           1   460.50 498.50
## - n_bathrooms           2   462.58 498.58
## - price                 1   460.67 498.67
## - has_terrace           1   461.25 499.25
## <none>                      459.98 499.98
## - heating               1   464.79 502.79
## - has_air_conditioning  1   476.07 514.07
##
## Step:  AIC=496.41
## is_furnished ~ id + price + mq + n_rooms + n_bathrooms + has_terrace +
##     has_alarm + heating + has_air_conditioning + has_parking
##
##                        Df Deviance    AIC
## - n_rooms               3   472.03 494.03
## - has_alarm             1   468.45 494.45
## - mq                    1   468.60 494.60
## - id                    1   468.64 494.64
## - has_parking           1   468.95 494.95
## - n_bathrooms           2   471.31 495.31
## - price                 1   469.49 495.49
## - has_terrace           1   470.26 496.26
## <none>                      468.41 496.41
## - heating               1   473.70 499.70
## - has_air_conditioning  1   484.48 510.48
##
## Step:  AIC=494.03
## is_furnished ~ id + price + mq + n_bathrooms + has_terrace +
##     has_alarm + heating + has_air_conditioning + has_parking
##
##                        Df Deviance    AIC
## - mq                    1   472.04 492.04
## - has_alarm             1   472.06 492.06
## - id                    1   472.24 492.24
## - has_parking           1   472.59 492.59
## - n_bathrooms           2   474.90 492.90
## - price                 1   473.37 493.37
## - has_terrace           1   473.92 493.92
## <none>                      472.03 494.03
## - heating               1   477.31 497.31
## - has_air_conditioning  1   488.38 508.38
##
## Step:  AIC=492.04
## is_furnished ~ id + price + n_bathrooms + has_terrace + has_alarm +
##     heating + has_air_conditioning + has_parking
##
##                        Df Deviance    AIC
## - has_alarm             1   472.07 490.07
## - id                    1   472.25 490.25
## - has_parking           1   472.60 490.60
## - n_bathrooms           2   474.97 490.97
## - price                 1   473.40 491.40
## - has_terrace           1   473.95 491.95
## <none>                      472.04 492.04
## - heating               1   477.31 495.31
## - has_air_conditioning  1   488.44 506.44
##
## Step:  AIC=490.07
## is_furnished ~ id + price + n_bathrooms + has_terrace + heating +
##     has_air_conditioning + has_parking
##
##                        Df Deviance    AIC
## - id                    1   472.28 488.28
## - has_parking           1   472.63 488.63
## - n_bathrooms           2   475.02 489.02
## - price                 1   473.40 489.40
## - has_terrace           1   474.00 490.00
## <none>                      472.07 490.07
## - heating               1   477.36 493.36
## - has_air_conditioning  1   488.52 504.52
```

```
##
## Step:  AIC=488.28
## is_furnished ~ price + n_bathrooms + has_terrace + heating +
##     has_air_conditioning + has_parking
##
##                         Df Deviance    AIC
## - has_parking            1   472.86 486.86
## - n_bathrooms            2   475.17 487.17
## - price                  1   473.54 487.54
## - has_terrace            1   474.23 488.23
## <none>                       472.28 488.28
## - heating                1   477.65 491.65
## - has_air_conditioning   1   488.75 502.75
##
## Step:  AIC=486.86
## is_furnished ~ price + n_bathrooms + has_terrace + heating +
##     has_air_conditioning
##
##                         Df Deviance    AIC
## - n_bathrooms            2   475.69 485.69
## - price                  1   474.11 486.11
## - has_terrace            1   474.68 486.68
## <none>                       472.86 486.86
## - heating                1   478.39 490.39
## - has_air_conditioning   1   488.96 500.96
##
## Step:  AIC=485.69
## is_furnished ~ price + has_terrace + heating + has_air_conditioning
##
##                         Df Deviance    AIC
## - price                  1   476.47 484.47
## <none>                       475.69 485.69
## - has_terrace            1   477.78 485.78
## - heating                1   480.88 488.88
## - has_air_conditioning   1   491.67 499.67
##
## Step:  AIC=484.47
## is_furnished ~ has_terrace + heating + has_air_conditioning
##
##                         Df Deviance    AIC
## <none>                       476.47 484.47
## - has_terrace            1   478.90 484.90
## - heating                1   481.43 487.43
## - has_air_conditioning   1   492.41 498.41
```

```
# Summary of the updated model
summary(furnished.model)
```

```
##
## Call:
## glm(formula = is_furnished ~ has_terrace + heating + has_air_conditioning,
##     family = "binomial", data = housedf_clean)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6807  -0.4239  -0.3320  -0.3320   2.7994
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -2.8708     0.1880 -15.271  < 2e-16 ***
## has_terrace1            0.5067     0.3137   1.615   0.1063
## heatingother           -1.0274     0.5285  -1.944   0.0519 .
## has_air_conditioning1   1.0199     0.2549   4.001 6.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 502.28  on 902  degrees of freedom
## Residual deviance: 476.47  on 899  degrees of freedom
## AIC: 484.47
##
## Number of Fisher Scoring iterations: 6
```

```
#plot(furnished.model)

# Predict the probability of property being furnished
housedf_clean$pfurnished <- predict(furnished.model, type = "response")

# Analysing pfurnished values
summary(housedf_clean$pfurnished)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01987 0.05361 0.05361 0.07973 0.13577 0.20682
```

```
# Calculating Chi-Square statistic for furnished model
chisq.furnished <- furnished.model$null.deviance - furnished.model$deviance
chisq.furnished
```

```
## [1] 25.81885
```

```
# Calculating predictor values degree of freedom
df.furnished <- furnished.model$df.null - furnished.model$df.residual
df.furnished
```

```
## [1] 3
```

```
# Calculating p-value from chi-square statistics
pchisq(chisq.furnished, df = df.furnished)
```

```
## [1] 0.9999896
```

```
# Transforming pfurnished to binary factors (1 if pfurnished > 0.5, 0 otherwise)
predictions_factor <- as.factor(ifelse(housedf_clean$pfurnished > 0.5, 1, 0))

# Call the confusionMatrix() function to generate the confusion matrix
confusion_matrix <- confusionMatrix(data = predictions_factor, reference = housedf_clean$is_furnished, positive =
"1")
```

```
## Warning in confusionMatrix.default(data = predictions_factor, reference =
## housedf_clean$is_furnished, : Levels are not in the same order for reference and
## data. Refactoring data to match.
```

```
# Display the confusion matrix
confusion_matrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 831  72
##          1   0   0
##
##                Accuracy : 0.9203
##                  95% CI : (0.9006, 0.9371)
##     No Information Rate : 0.9203
##     P-Value [Acc > NIR] : 0.5313
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.00000
##             Specificity : 1.00000
##          Pos Pred Value :     NaN
##          Neg Pred Value : 0.92027
##              Prevalence : 0.07973
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.00000
##       Balanced Accuracy : 0.50000
##
##        'Positive' Class : 1
##
```

```
# Odd ratios
exp(coef(furnished.model))
```

```
##           (Intercept)          has_terrace1          heatingother
##            0.05665217            1.65977015            0.35793041
## has_air_conditioning1
##            2.77299760
```

References

[@r-charts] r-charts (2020). “Pie Chart Labels Outside with ggplot2.” Retrieved from https://r-charts.com/part-whole/pie-chart-labels-outside-ggplot2/ (https://r-charts.com/part-whole/pie-chart-labels-outside-ggplot2/)

[@vif] VIF (2020). “vif: Variance Inflation Factors.” Retrieved from https://www.rdocumentation.org/packages/VIF/versions/1.0/topics/vif (https://www.rdocumentation.org/packages/VIF/versions/1.0/topics/vif)

[@GLM] Statology (2020). Interpret GLM Output in R. Retrieved from **https://www.statology.org/interpret-glm-output-in-r/** (https://www.statology.org/interpret-glm-output-in-r/).

[@ConfMatrix] R Development Core Team (2020). caret: Classification and Regression Training. Retrieved from **https://CRAN.R-project.org/package=caret** (https://CRAN.R-project.org/package=caret)