Final Step3 - Analysis of Airbnb Rental Prices

Ghanta, Madhavi

2023-05-31

Problem Statement

Airbnb is an online marketplace that connects people who want to rent out their homes with people who are looking for accommodations in that locale.It currently covers more than 100,000 cities and 220 countries worldwide. It largely does not own dwellings or real estate of its own; instead, it collects fees by acting as a broker between those with dwellings to rent and those looking to book lodging. The company has been criticized for a direct correlation between increases in the number of its listings and increases in nearby rent prices and creating nuisances for those living near leased properties. The problem here I am addressing is how the the prices of Chicago AirBnB rentals affect the prices of the nearby neighborhood rent prices. Data science algorithm will help here to predict the prices of Chicago AirBnB rentals and also help to understand the correlation between the prices of Chicago AirBnB rentals and neighborhood rent prices.

Research Questions

- What are the Airbnb rental prices for different areas in Chicago?
- What is the correlation between the Airbnb rental prices and Chicago neighborhood rental prices?
- What are the average rental prices by the neighborhood?
- What are the average rental prices for Airbnb by the neighborhood?
- What type of houses are most rented on Airbnb?
- What is the monthly rent from the Airbnb properties?
- What are the rental property options by neighborhood?
- How much profit does Airbnb make monthly?

Approach

Approach involves analyzing data to discover correlations, patterns and create machine learning model to predict how AirBnB rentals prices affects the nearby housing rental prices in Chicago based of various factors i.e. neighborhood, zip code, Airbnb prices, number of reviews, housing rental area, housing rental units etc. 1. The approach is to start with finding the most important predictors for the regression model. 2. Once the predictors are decided then I will look into the R2 , Adjusted R2 statistics, p-value. 3. I will then calculate the betas for the predictors in the regression model. It will tell me how the 1 standard deviation change in predictor will impact dependent (response) variable. 4. I will then calculate confidence intervals which indicate that the estimates how the model are likely to be representative of the true population values. 5. I will then perform an analysis of variance on all models to compare performance of different models. 6. I will then calculate standardized residuals, the leverage, cooks distance, and covariance rations 7. At last I will check if the regression model unbiased and then will select the unbiased model for the prediction of the Airbnb prices

How your approach addresses (fully or partially) the problem.

Approach focus on to give enough data inputs to be able to address the problem completely. The approach will help to predict direct correlation between increases in the number of its listings and increases in nearby rent prices. It will help uncover various data patterns to answer multiple research questions. It will help understand cause and effect relationship between Airbnb prices and nearby housing rental prices. It also intends to develop a model to predict Airbnb prices based on given variables.

Packages

Load the readxl package

Set the working directory to the root of your DSC 520 directory

```
setwd("C:/Users/mghan/Documents/dsc520/FinalProject")

#Above data set contains information across US cities
#Filtering the data based on city==Chicago as we are focusing on Chicago
library(readr)
airbnb_chicago_df <- readr::read_csv('airbnb-listings.csv')
```

```
## Rows: 6357 Columns: 74
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (28): listing_url, last_scraped, name, description, neighborhood_overvie...
## dbl (38): id, scrape_id, host_id, host_listings_count, host_total_listings_c...
## lgl  (8): host_is_superhost, host_has_profile_pic, host_identity_verified, n...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(airbnb_chicago_df)
```

```
## spc_tbl_ [6,357 x 74] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id                                          : num [1:6357] 2384 7126 10945 12068 12140 ...
##  $ listing_url                                 : chr [1:6357] "https://www.airbnb.com/rooms/2384" "ht
##  $ scrape_id                                   : num [1:6357] 2.02e+13 2.02e+13 2.02e+13 2.02e+13 2.0
##  $ last_scraped                                : chr [1:6357] "7/11/2021" "7/11/2021" "7/11/2021" "7/
##  $ name                                        : chr [1:6357] "Hyde Park - Walk to University of Chi
##  $ description                                 : chr [1:6357] "If you have been fully vaccinated, you
##  $ neighborhood_overview                       : chr [1:6357] "The apartment is less than one block f
##  $ picture_url                                 : chr [1:6357] "https://a0.muscache.com/pictures/acf6
##  $ host_id                                     : num [1:6357] 2613 17928 33004 40731 46734 ...
##  $ host_url                                    : chr [1:6357] "https://www.airbnb.com/users/show/261
##  $ host_name                                   : chr [1:6357] "Rebecca" "Sarah" "At Home Inn" "Domini
##  $ host_since                                  : chr [1:6357] "8/29/2008" "5/19/2009" "8/21/2009" "9/
##  $ host_location                               : chr [1:6357] "Chicago, Illinois, United States" "Chi
##  $ host_about                                  : chr [1:6357] "My 2 bdrm apartment is a 2nd floor wal
##  $ host_response_time                          : chr [1:6357] "within an hour" "within an hour" "with
##  $ host_response_rate                          : chr [1:6357] "100%" "100%" "100%" "92%" ...
```

```
##  $ host_acceptance_rate                    : chr [1:6357] "93%" "96%" "92%" "94%" ...
##  $ host_is_superhost                       : logi [1:6357] TRUE TRUE TRUE FALSE FALSE FALSE ...
##  $ host_thumbnail_url                      : chr [1:6357] "https://a0.muscache.com/im/pictures/us
##  $ host_picture_url                        : chr [1:6357] "https://a0.muscache.com/im/pictures/us
##  $ host_neighbourhood                      : chr [1:6357] "Hyde Park" "Ukrainian Village" "Old To
##  $ host_listings_count                     : num [1:6357] 1 2 10 3 1 2 8 8 3 2 ...
##  $ host_total_listings_count               : num [1:6357] 1 2 10 3 1 2 8 8 3 2 ...
##  $ host_verifications                      : chr [1:6357] "['email', 'phone', 'reviews', 'manual_
##  $ host_has_profile_pic                    : logi [1:6357] TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ host_identity_verified                  : logi [1:6357] TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ neighbourhood                           : chr [1:6357] "Chicago, Illinois, United States" "Chi
##  $ neighbourhood_cleansed                  : chr [1:6357] "Hyde Park" "West Town" "Lincoln Park"
##  $ neighbourhood_group_cleansed            : logi [1:6357] NA NA NA NA NA NA ...
##  $ latitude                                : num [1:6357] 41.8 41.9 41.9 41.9 41.9 ...
##  $ longitude                               : num [1:6357] -87.6 -87.7 -87.6 -87.6 -87.6 ...
##  $ property_type                           : chr [1:6357] "Private room in condominium" "Entire a
##  $ room_type                               : chr [1:6357] "Private room" "Entire home/apt" "Entir
##  $ accommodates                            : num [1:6357] 1 2 4 2 2 4 3 6 6 2 ...
##  $ bathrooms                               : logi [1:6357] NA NA NA NA NA NA ...
##  $ bathrooms_text                          : chr [1:6357] "1 shared bath" "1 bath" "1 bath" "1 ba
##  $ bedrooms                                : num [1:6357] 1 1 2 1 1 2 1 2 3 1 ...
##  $ beds                                    : num [1:6357] 1 1 2 1 2 2 1 0 3 1 ...
##  $ amenities                               : chr [1:6357] "[\"Hot water kettle\", \"Wine glasses\
##  $ price                                   : num [1:6357] 85 65 143 99 329 105 133 95 214 110 ..
##  $ minimum_nights                          : num [1:6357] 1 2 4 7 2 121 32 32 2 3 ...
##  $ maximum_nights                          : num [1:6357] 90 60 180 180 7 ...
##  $ minimum_minimum_nights                  : num [1:6357] 2 2 4 7 2 121 32 11 2 3 ...
##  $ maximum_minimum_nights                  : num [1:6357] 4 2 4 7 2 121 56 32 2 3 ...
##  $ minimum_maximum_nights                  : num [1:6357] 90 1125 180 180 7 ...
##  $ maximum_maximum_nights                  : num [1:6357] 90 1125 180 180 7 ...
##  $ minimum_nights_avg_ntm                  : num [1:6357] 2 2 4 7 2 121 37.5 31.5 2 3 ...
##  $ maximum_nights_avg_ntm                  : num [1:6357] 90 1125 180 180 7 ...
##  $ calendar_updated                        : logi [1:6357] NA NA NA NA NA NA ...
##  $ has_availability                        : logi [1:6357] TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ availability_30                         : num [1:6357] 10 1 0 0 19 0 0 6 2 0 ...
##  $ availability_60                         : num [1:6357] 33 11 7 3 47 0 0 10 14 0 ...
##  $ availability_90                         : num [1:6357] 63 32 17 3 77 0 0 25 38 0 ...
##  $ availability_365                        : num [1:6357] 338 59 102 242 165 273 252 238 175 0 .
##  $ calendar_last_scraped                   : chr [1:6357] "7/11/2021" "7/11/2021" "7/11/2021" "7/
##  $ number_of_reviews                       : num [1:6357] 185 401 28 11 7 9 37 47 97 250 ...
##  $ number_of_reviews_ltm                   : num [1:6357] 7 17 11 2 3 0 0 0 46 0 ...
##  $ number_of_reviews_l30d                  : num [1:6357] 2 0 1 0 1 0 0 0 4 0 ...
##  $ first_review                            : chr [1:6357] "4/30/2015" "8/10/2011" "4/28/2014" "2/
##  $ last_review                             : chr [1:6357] "6/21/2021" "5/22/2021" "6/21/2021" "5/
##  $ review_scores_rating                    : num [1:6357] 4.99 4.66 4.5 4.73 5 4.67 4.27 4.36 4.6
##  $ review_scores_accuracy                  : num [1:6357] 4.98 4.83 4.64 4.73 5 4.78 3.96 4.58 4
##  $ review_scores_cleanliness               : num [1:6357] 4.99 4.52 4.68 4.73 5 4.78 3.92 4.33 4
##  $ review_scores_checkin                   : num [1:6357] 4.98 4.89 4.68 4.64 5 5 4.08 4.12 4.87
##  $ review_scores_communication             : num [1:6357] 4.98 4.85 4.64 4.73 5 5 3.96 4.15 4.83
##  $ review_scores_location                  : num [1:6357] 4.95 4.87 4.96 4.73 5 4.89 4.88 4.91 4
##  $ review_scores_value                     : num [1:6357] 4.94 4.72 4.54 4.73 5 4.89 4.08 4.36 4
##  $ license                                 : chr [1:6357] "R17000015609" "R18000034991" "2120297
##  $ instant_bookable                        : logi [1:6357] FALSE FALSE TRUE FALSE FALSE FALSE ..
##  $ calculated_host_listings_count          : num [1:6357] 1 1 10 1 1 1 4 4 1 2 ...
```

```
##  $ calculated_host_listings_count_entire_homes : num [1:6357] 0 1 10 1 0 1 4 4 1 1 ...
##  $ calculated_host_listings_count_private_rooms: num [1:6357] 1 0 0 0 1 0 0 0 0 1 ...
##  $ calculated_host_listings_count_shared_rooms : num [1:6357] 0 0 0 0 0 0 0 0 0 0 ...
##  $ reviews_per_month                           : num [1:6357] 2.45 3.32 0.32 0.14 0.1 0.1 0.33 1.54 (
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   listing_url = col_character(),
##   ..   scrape_id = col_double(),
##   ..   last_scraped = col_character(),
##   ..   name = col_character(),
##   ..   description = col_character(),
##   ..   neighborhood_overview = col_character(),
##   ..   picture_url = col_character(),
##   ..   host_id = col_double(),
##   ..   host_url = col_character(),
##   ..   host_name = col_character(),
##   ..   host_since = col_character(),
##   ..   host_location = col_character(),
##   ..   host_about = col_character(),
##   ..   host_response_time = col_character(),
##   ..   host_response_rate = col_character(),
##   ..   host_acceptance_rate = col_character(),
##   ..   host_is_superhost = col_logical(),
##   ..   host_thumbnail_url = col_character(),
##   ..   host_picture_url = col_character(),
##   ..   host_neighbourhood = col_character(),
##   ..   host_listings_count = col_double(),
##   ..   host_total_listings_count = col_double(),
##   ..   host_verifications = col_character(),
##   ..   host_has_profile_pic = col_logical(),
##   ..   host_identity_verified = col_logical(),
##   ..   neighbourhood = col_character(),
##   ..   neighbourhood_cleansed = col_character(),
##   ..   neighbourhood_group_cleansed = col_logical(),
##   ..   latitude = col_double(),
##   ..   longitude = col_double(),
##   ..   property_type = col_character(),
##   ..   room_type = col_character(),
##   ..   accommodates = col_double(),
##   ..   bathrooms = col_logical(),
##   ..   bathrooms_text = col_character(),
##   ..   bedrooms = col_double(),
##   ..   beds = col_double(),
##   ..   amenities = col_character(),
##   ..   price = col_double(),
##   ..   minimum_nights = col_double(),
##   ..   maximum_nights = col_double(),
##   ..   minimum_minimum_nights = col_double(),
##   ..   maximum_minimum_nights = col_double(),
##   ..   minimum_maximum_nights = col_double(),
##   ..   maximum_maximum_nights = col_double(),
##   ..   minimum_nights_avg_ntm = col_double(),
##   ..   maximum_nights_avg_ntm = col_double(),
```

```
##   ..   calendar_updated = col_logical(),
##   ..   has_availability = col_logical(),
##   ..   availability_30 = col_double(),
##   ..   availability_60 = col_double(),
##   ..   availability_90 = col_double(),
##   ..   availability_365 = col_double(),
##   ..   calendar_last_scraped = col_character(),
##   ..   number_of_reviews = col_double(),
##   ..   number_of_reviews_ltm = col_double(),
##   ..   number_of_reviews_l30d = col_double(),
##   ..   first_review = col_character(),
##   ..   last_review = col_character(),
##   ..   review_scores_rating = col_double(),
##   ..   review_scores_accuracy = col_double(),
##   ..   review_scores_cleanliness = col_double(),
##   ..   review_scores_checkin = col_double(),
##   ..   review_scores_communication = col_double(),
##   ..   review_scores_location = col_double(),
##   ..   review_scores_value = col_double(),
##   ..   license = col_character(),
##   ..   instant_bookable = col_logical(),
##   ..   calculated_host_listings_count = col_double(),
##   ..   calculated_host_listings_count_entire_homes = col_double(),
##   ..   calculated_host_listings_count_private_rooms = col_double(),
##   ..   calculated_host_listings_count_shared_rooms = col_double(),
##   ..   reviews_per_month = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(airbnb_chicago_df)
```

```
##        id           listing_url          scrape_id        last_scraped
##  Min.   :    2384   Length:6357        Min.   :2.02e+13   Length:6357
##  1st Qu.:20834660   Class :character   1st Qu.:2.02e+13   Class :character
##  Median :35194693   Mode  :character   Median :2.02e+13   Mode  :character
##  Mean   :31907317                      Mean   :2.02e+13
##  3rd Qu.:44630643                      3rd Qu.:2.02e+13
##  Max.   :50952621                      Max.   :2.02e+13
##
##      name           description        neighborhood_overview picture_url
##  Length:6357        Length:6357        Length:6357           Length:6357
##  Class :character   Class :character   Class :character      Class :character
##  Mode  :character   Mode  :character   Mode  :character      Mode  :character
##
##
##
##
##     host_id            host_url          host_name          host_since
##  Min.   :     2153   Length:6357        Length:6357        Length:6357
##  1st Qu.: 20365985   Class :character   Class :character   Class :character
##  Median : 74157338   Mode  :character   Mode  :character   Mode  :character
##  Mean   :117505299
##  3rd Qu.:186121450
##  Max.   :409973260
```

```
##
##  host_location       host_about        host_response_time host_response_rate
##  Length:6357        Length:6357        Length:6357        Length:6357
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  host_acceptance_rate host_is_superhost host_thumbnail_url host_picture_url
##  Length:6357          Mode :logical     Length:6357        Length:6357
##  Class :character     FALSE:4027        Class :character   Class :character
##  Mode  :character     TRUE :2329        Mode  :character   Mode  :character
##                       NA's :1
##
##
##
##  host_neighbourhood host_listings_count host_total_listings_count
##  Length:6357        Min.   :   0.00     Min.   :   0.00
##  Class :character   1st Qu.:   1.00     1st Qu.:   1.00
##  Mode  :character   Median :   2.00     Median :   2.00
##                     Mean   :  82.11     Mean   :  82.11
##                     3rd Qu.:   9.00     3rd Qu.:   9.00
##                     Max.   :3924.00     Max.   :3924.00
##                     NA's   :1           NA's   :1
##  host_verifications host_has_profile_pic host_identity_verified
##  Length:6357        Mode :logical        Mode :logical
##  Class :character   FALSE:13             FALSE:1140
##  Mode  :character   TRUE :6343           TRUE :5216
##                     NA's :1              NA's :1
##
##
##
##  neighbourhood      neighbourhood_cleansed neighbourhood_group_cleansed
##  Length:6357        Length:6357            Mode:logical
##  Class :character   Class :character       NA's:6357
##  Mode  :character   Mode  :character
##
##
##
##
##     latitude       longitude      property_type       room_type
##  Min.   :41.65   Min.   :-87.85   Length:6357        Length:6357
##  1st Qu.:41.87   1st Qu.:-87.69   Class :character   Class :character
##  Median :41.90   Median :-87.66   Mode  :character   Mode  :character
##  Mean   :41.90   Mean   :-87.66
##  3rd Qu.:41.94   3rd Qu.:-87.63
##  Max.   :42.02   Max.   :-87.54
##
##   accommodates    bathrooms      bathrooms_text        bedrooms
##  Min.   : 0.000   Mode:logical   Length:6357        Min.   : 1.000
##  1st Qu.: 2.000   NA's:6357      Class :character   1st Qu.: 1.000
##  Median : 4.000                  Mode  :character   Median : 1.000
##  Mean   : 4.094                                     Mean   : 1.761
```

```
##  3rd Qu.: 5.000                                      3rd Qu.: 2.000
##  Max.   :16.000                                      Max.   :12.000
##                                                      NA's   :537
##       beds          amenities             price        minimum_nights
##  Min.   : 0.000   Length:6357        Min.   :   0.0   Min.   :  1.000
##  1st Qu.: 1.000   Class :character   1st Qu.:  75.0   1st Qu.:  1.000
##  Median : 2.000   Mode  :character   Median : 120.0   Median :  2.000
##  Mean   : 2.119                      Mean   : 183.2   Mean   :  9.047
##  3rd Qu.: 3.000                      3rd Qu.: 200.0   3rd Qu.:  4.000
##  Max.   :25.000                      Max.   :9999.0   Max.   :500.000
##  NA's   :81
##  maximum_nights   minimum_minimum_nights maximum_minimum_nights
##  Min.   :   1.0   Min.   :  1.00         Min.   :  1.00
##  1st Qu.:  60.0   1st Qu.:  2.00         1st Qu.:  2.00
##  Median :1125.0   Median :  2.00         Median :  3.00
##  Mean   : 696.5   Mean   : 10.13         Mean   : 38.55
##  3rd Qu.:1125.0   3rd Qu.:  4.00         3rd Qu.:  5.00
##  Max.   :1125.0   Max.   :730.00         Max.   :730.00
##
##  minimum_maximum_nights maximum_maximum_nights minimum_nights_avg_ntm
##  Min.   :1.000e+00      Min.   :1.000e+00      Min.   :  1.00
##  1st Qu.:3.650e+02      1st Qu.:1.123e+03      1st Qu.:  2.00
##  Median :1.125e+03      Median :1.125e+03      Median :  2.00
##  Mean   :6.081e+06      Mean   :6.082e+06      Mean   : 37.27
##  3rd Qu.:1.125e+03      3rd Qu.:1.125e+03      3rd Qu.:  5.00
##  Max.   :2.147e+09      Max.   :2.147e+09      Max.   :730.00
##
##  maximum_nights_avg_ntm calendar_updated has_availability availability_30
##  Min.   :1.000e+00      Mode:logical     Mode :logical    Min.   : 0.000
##  1st Qu.:9.200e+02      NA's:6357        FALSE:59         1st Qu.: 0.000
##  Median :1.125e+03                       TRUE :6298       Median : 3.000
##  Mean   :6.082e+06                                        Mean   : 7.401
##  3rd Qu.:1.125e+03                                        3rd Qu.:12.000
##  Max.   :2.147e+09                                        Max.   :30.000
##
##  availability_60 availability_90 availability_365 calendar_last_scraped
##  Min.   : 0.00   Min.   : 0.00   Min.   :  0.0    Length:6357
##  1st Qu.: 0.00   1st Qu.: 2.00   1st Qu.: 31.0    Class :character
##  Median :16.00   Median :35.00   Median :146.0    Mode  :character
##  Mean   :20.37   Mean   :35.65   Mean   :164.9
##  3rd Qu.:35.00   3rd Qu.:60.00   3rd Qu.:305.0
##  Max.   :60.00   Max.   :90.00   Max.   :365.0
##
##  number_of_reviews number_of_reviews_ltm number_of_reviews_l30d
##  Min.   :   0.00   Min.   :  0.000       Min.   :  0.000
##  1st Qu.:   2.00   1st Qu.:  0.000       1st Qu.:  0.000
##  Median :  13.00   Median :  2.000       Median :  0.000
##  Mean   :  43.59   Mean   :  9.198       Mean   :  1.251
##  3rd Qu.:  55.00   3rd Qu.: 11.000       3rd Qu.:  2.000
##  Max.   :1027.00   Max.   :647.000       Max.   :119.000
##
##  first_review       last_review        review_scores_rating
##  Length:6357        Length:6357        Min.   :0.000
##  Class :character   Class :character   1st Qu.:4.670
```

```
##  Mode  :character   Mode  :character   Median :4.850
##                                        Mean   :4.702
##                                        3rd Qu.:4.970
##                                        Max.   :5.000
##                                        NA's   :1083
##  review_scores_accuracy review_scores_cleanliness review_scores_checkin
##  Min.   :1.000          Min.   :1.00              Min.   :1.000
##  1st Qu.:4.760          1st Qu.:4.66              1st Qu.:4.850
##  Median :4.910          Median :4.86              Median :4.950
##  Mean   :4.796          Mean   :4.73              Mean   :4.855
##  3rd Qu.:5.000          3rd Qu.:4.99              3rd Qu.:5.000
##  Max.   :5.000          Max.   :5.00              Max.   :5.000
##  NA's   :1119           NA's   :1119              NA's   :1120
##  review_scores_communication review_scores_location review_scores_value
##  Min.   :1.000               Min.   :1.000          Min.   :1.000
##  1st Qu.:4.860               1st Qu.:4.750          1st Qu.:4.650
##  Median :4.960               Median :4.900          Median :4.810
##  Mean   :4.853               Mean   :4.795          Mean   :4.706
##  3rd Qu.:5.000               3rd Qu.:5.000          3rd Qu.:4.920
##  Max.   :5.000               Max.   :5.000          Max.   :5.000
##  NA's   :1121                NA's   :1120           NA's   :1120
##    license          instant_bookable calculated_host_listings_count
##  Length:6357        Mode :logical    Min.   :  1.00
##  Class :character   FALSE:3923       1st Qu.:  1.00
##  Mode  :character   TRUE :2434       Median :  2.00
##                                      Mean   : 18.21
##                                      3rd Qu.:  9.00
##                                      Max.   :260.00
##
##  calculated_host_listings_count_entire_homes
##  Min.   :  0.00
##  1st Qu.:  1.00
##  Median :  1.00
##  Mean   : 16.46
##  3rd Qu.:  4.00
##  Max.   :260.00
##
##  calculated_host_listings_count_private_rooms
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 0.000
##  Mean   : 1.467
##  3rd Qu.: 1.000
##  Max.   :24.000
##
##  calculated_host_listings_count_shared_rooms reviews_per_month
##  Min.   : 0.0000                             Min.   :  0.010
##  1st Qu.: 0.0000                             1st Qu.:  0.470
##  Median : 0.0000                             Median :  1.480
##  Mean   : 0.1417                             Mean   :  2.665
##  3rd Qu.: 0.0000                             3rd Qu.:  3.300
##  Max.   :16.0000                             Max.   :121.820
##                                              NA's   :1083
```

```
## Load the Affordable rental housing dataset

housing_df=read.csv("C:/Users/mghan/Documents/dsc520/FinalProject/Affordable_Rental_Housing_Developments
glimpse(housing_df)
```

```
## Rows: 488
## Columns: 14
## $ neighbourhood_cleansed <chr> "Edgewater", "Roseland", "Humboldt Park", "Gran~
## $ Community.Area.Number  <int> 77, 49, 23, 38, 42, 36, 36, 8, 24, 18, 14, 38, ~
## $ Property.Type          <chr> "Multifamily", "Senior", "Multifamily", "Multif~
## $ Property.Name          <chr> "Winthrop Apts.", "Victory Center of Roseland",~
## $ Address                <chr> "6214 N. Winthrop Ave.", "10450 S. Michigan Ave~
## $ Zip.Code               <int> 60660, 60628, 60624, 60615, 60637, 60653, 60653~
## $ Phone.Number           <chr> "773-477-7070", "773-468-6400", "773-227-6332",~
## $ Management.Company     <chr> "Hunter Properties", "Pathway Senior Living", "~
## $ Units                  <int> 108, 81, 6, 8, 33, 148, 76, 7, 3, 3, 97, 220, 6~
## $ X.Coordinate           <dbl> 1167689, 1178829, 1155445, 1181237, 1182661, 11~
## $ Y.Coordinate           <dbl> 1941496, 1835494, 1903207, 1871959, 1864419, 18~
## $ Latitude               <dbl> 41.99502, 41.70389, 41.89020, 41.80390, 41.7831~
## $ Longitude              <dbl> -87.65852, -87.62077, -87.70459, -87.61083, -87~
## $ Location               <chr> "(41.9950154575665, -87.6585160357341)", "(41.7~
```

```
## Load the Average rent Chicago neighborhood dataset

avg_rent_df <- read_excel("C:/Users/mghan/Documents/dsc520/FinalProject/Average_rent_Chicago_neighbourho
glimpse(avg_rent_df)
```

```
## Rows: 70
## Columns: 2
## $ neighbourhood_cleansed <chr> "Near North Side", "Lakeview", "West Town", "Lo~
## $ `Average Rent`         <dbl> 2200, 1395, 1600, 2350, 1299, 1500, 1180, 1299,~
```

```
#Merge the airbnb df with rental housing df based on neighbourhood
final_1_df <- left_join(airbnb_chicago_df,housing_df,by="neighbourhood_cleansed")
```

```
## Warning in left_join(airbnb_chicago_df, housing_df, by = "neighbourhood_cleansed"): Detected an unexp
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 82 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
glimpse(final_1_df)
```

```
## Rows: 88,192
## Columns: 87
## $ id                                          <dbl> 2384, 2384, 2384, 2384, 2~
## $ listing_url                                 <chr> "https://www.airbnb.com/r~
## $ scrape_id                                   <dbl> 2.02e+13, 2.02e+13, 2.02e~
## $ last_scraped                                <chr> "7/11/2021", "7/11/2021",~
## $ name                                        <chr> "Hyde Park - Walk to Univ~
## $ description                                 <chr> "If you have been fully v~
```

```
## $ neighborhood_overview                <chr> "The apartment is less th~
## $ picture_url                          <chr> "https://a0.muscache.com/~
## $ host_id                              <dbl> 2613, 2613, 2613, 2613, 2~
## $ host_url                             <chr> "https://www.airbnb.com/u~
## $ host_name                            <chr> "Rebecca", "Rebecca", "Re~
## $ host_since                           <chr> "8/29/2008", "8/29/2008",~
## $ host_location                        <chr> "Chicago, Illinois, Unite~
## $ host_about                           <chr> "My 2 bdrm apartment is a~
## $ host_response_time                   <chr> "within an hour", "within~
## $ host_response_rate                   <chr> "100%", "100%", "100%", "~
## $ host_acceptance_rate                 <chr> "93%", "93%", "93%", "93%~
## $ host_is_superhost                    <lgl> TRUE, TRUE, TRUE, TRUE, T~
## $ host_thumbnail_url                   <chr> "https://a0.muscache.com/~
## $ host_picture_url                     <chr> "https://a0.muscache.com/~
## $ host_neighbourhood                   <chr> "Hyde Park", "Hyde Park",~
## $ host_listings_count                  <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2~
## $ host_total_listings_count            <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2~
## $ host_verifications                   <chr> "['email', 'phone', 'revi~
## $ host_has_profile_pic                 <lgl> TRUE, TRUE, TRUE, TRUE, T~
## $ host_identity_verified               <lgl> TRUE, TRUE, TRUE, TRUE, T~
## $ neighbourhood                        <chr> "Chicago, Illinois, Unite~
## $ neighbourhood_cleansed               <chr> "Hyde Park", "Hyde Park",~
## $ neighbourhood_group_cleansed         <lgl> NA, NA, NA, NA, NA, NA, N~
## $ latitude                             <dbl> 41.78790, 41.78790, 41.78~
## $ longitude                            <dbl> -87.58780, -87.58780, -87~
## $ property_type                        <chr> "Private room in condomin~
## $ room_type                            <chr> "Private room", "Private ~
## $ accommodates                         <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2~
## $ bathrooms                            <lgl> NA, NA, NA, NA, NA, NA, N~
## $ bathrooms_text                       <chr> "1 shared bath", "1 share~
## $ bedrooms                             <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ beds                                 <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ amenities                            <chr> "[\"Hot water kettle\", \~
## $ price                                <dbl> 85, 85, 85, 85, 85, 65, 6~
## $ minimum_nights                       <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2~
## $ maximum_nights                       <dbl> 90, 90, 90, 90, 90, 60, 6~
## $ minimum_minimum_nights               <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2~
## $ maximum_minimum_nights               <dbl> 4, 4, 4, 4, 4, 2, 2, 2, 2~
## $ minimum_maximum_nights               <dbl> 90, 90, 90, 90, 90, 1125,~
## $ maximum_maximum_nights               <dbl> 90, 90, 90, 90, 90, 1125,~
## $ minimum_nights_avg_ntm               <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2~
## $ maximum_nights_avg_ntm               <dbl> 90, 90, 90, 90, 90, 1125,~
## $ calendar_updated                     <lgl> NA, NA, NA, NA, NA, NA, N~
## $ has_availability                     <lgl> TRUE, TRUE, TRUE, TRUE, T~
## $ availability_30                      <dbl> 10, 10, 10, 10, 10, 1, 1,~
## $ availability_60                      <dbl> 33, 33, 33, 33, 33, 11, 1~
## $ availability_90                      <dbl> 63, 63, 63, 63, 63, 32, 3~
## $ availability_365                     <dbl> 338, 338, 338, 338, 338, ~
## $ calendar_last_scraped                <chr> "7/11/2021", "7/11/2021",~
## $ number_of_reviews                    <dbl> 185, 185, 185, 185, 185, ~
## $ number_of_reviews_ltm                <dbl> 7, 7, 7, 7, 7, 17, 17, 17~
## $ number_of_reviews_l30d               <dbl> 2, 2, 2, 2, 2, 0, 0, 0, 0~
## $ first_review                         <chr> "4/30/2015", "4/30/2015",~
## $ last_review                          <chr> "6/21/2021", "6/21/2021",~
```

```
## $ review_scores_rating                            <dbl> 4.99, 4.99, 4.99, 4.99, 4~
## $ review_scores_accuracy                          <dbl> 4.98, 4.98, 4.98, 4.98, 4~
## $ review_scores_cleanliness                       <dbl> 4.99, 4.99, 4.99, 4.99, 4~
## $ review_scores_checkin                           <dbl> 4.98, 4.98, 4.98, 4.98, 4~
## $ review_scores_communication                     <dbl> 4.98, 4.98, 4.98, 4.98, 4~
## $ review_scores_location                          <dbl> 4.95, 4.95, 4.95, 4.95, 4~
## $ review_scores_value                             <dbl> 4.94, 4.94, 4.94, 4.94, 4~
## $ license                                         <chr> "R17000015609", "R1700001~
## $ instant_bookable                                <lgl> FALSE, FALSE, FALSE, FALS~
## $ calculated_host_listings_count                  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ calculated_host_listings_count_entire_homes     <dbl> 0, 0, 0, 0, 0, 1, 1, 1, 1~
## $ calculated_host_listings_count_private_rooms    <dbl> 1, 1, 1, 1, 1, 0, 0, 0, 0~
## $ calculated_host_listings_count_shared_rooms     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ reviews_per_month                               <dbl> 2.45, 2.45, 2.45, 2.45, 2~
## $ Community.Area.Number                           <int> 41, 41, 41, 41, 41, 24, 2~
## $ Property.Type                                   <chr> "ARO", "ARO", "ARO", "ARO~
## $ Property.Name                                   <chr> "City Hyde Park", "Vue53"~
## $ Address                                         <chr> "5105 S. Harper Ave.", "1~
## $ Zip.Code                                        <int> 60615, 60615, 60615, 6061~
## $ Phone.Number                                    <chr> "773-548-5077", "773-355-~
## $ Management.Company                              <chr> "Mac Properties", "Peak C~
## $ Units                                           <int> 36, 27, 2, 10, 36, 3, 52,~
## $ X.Coordinate                                    <dbl> 1187194, 1185905, 1186745~
## $ Y.Coordinate                                    <dbl> 1871413, 1870431, 1870452~
## $ Latitude                                        <dbl> 41.80226, 41.79960, 41.79~
## $ Longitude                                       <dbl> -87.58900, -87.59376, -87~
## $ Location                                        <chr> "(41.8022605698632, -87.5~
```

```
head(final_1_df)
```

```
## # A tibble: 6 x 87
##      id listing_url                       scrape_id last_scraped name  description
##   <dbl> <chr>                                <dbl> <chr>        <chr> <chr>
## 1  2384 https://www.airbnb.com/rooms/2~   2.02e13 7/11/2021    Hyde~ If you hav~
## 2  2384 https://www.airbnb.com/rooms/2~   2.02e13 7/11/2021    Hyde~ If you hav~
## 3  2384 https://www.airbnb.com/rooms/2~   2.02e13 7/11/2021    Hyde~ If you hav~
## 4  2384 https://www.airbnb.com/rooms/2~   2.02e13 7/11/2021    Hyde~ If you hav~
## 5  2384 https://www.airbnb.com/rooms/2~   2.02e13 7/11/2021    Hyde~ If you hav~
## 6  7126 https://www.airbnb.com/rooms/7~   2.02e13 7/11/2021    Tiny~ A very sma~
## # i 81 more variables: neighborhood_overview <chr>, picture_url <chr>,
## #   host_id <dbl>, host_url <chr>, host_name <chr>, host_since <chr>,
## #   host_location <chr>, host_about <chr>, host_response_time <chr>,
## #   host_response_rate <chr>, host_acceptance_rate <chr>,
## #   host_is_superhost <lgl>, host_thumbnail_url <chr>, host_picture_url <chr>,
## #   host_neighbourhood <chr>, host_listings_count <dbl>,
## #   host_total_listings_count <dbl>, host_verifications <chr>, ...
```

```
#Merge the above df with Average rent df based on neighbourhood
final_2_df <-inner_join(x=final_1_df,y=avg_rent_df,
                        by=c("neighbourhood_cleansed"))
glimpse(final_2_df)
```

```
## Rows: 78,313
```

```
## Columns: 88
## $ id                                  <dbl> 2384, 2384, 2384, 2384, 2~
## $ listing_url                         <chr> "https://www.airbnb.com/r~
## $ scrape_id                           <dbl> 2.02e+13, 2.02e+13, 2.02e~
## $ last_scraped                        <chr> "7/11/2021", "7/11/2021",~
## $ name                                <chr> "Hyde Park - Walk to Univ~
## $ description                         <chr> "If you have been fully v~
## $ neighborhood_overview               <chr> "The apartment is less th~
## $ picture_url                         <chr> "https://a0.muscache.com/~
## $ host_id                             <dbl> 2613, 2613, 2613, 2613, 2~
## $ host_url                            <chr> "https://www.airbnb.com/u~
## $ host_name                           <chr> "Rebecca", "Rebecca", "Re~
## $ host_since                          <chr> "8/29/2008", "8/29/2008",~
## $ host_location                       <chr> "Chicago, Illinois, Unite~
## $ host_about                          <chr> "My 2 bdrm apartment is a~
## $ host_response_time                  <chr> "within an hour", "within~
## $ host_response_rate                  <chr> "100%", "100%", "100%", "~
## $ host_acceptance_rate                <chr> "93%", "93%", "93%", "93%~
## $ host_is_superhost                   <lgl> TRUE, TRUE, TRUE, TRUE, T~
## $ host_thumbnail_url                  <chr> "https://a0.muscache.com/~
## $ host_picture_url                    <chr> "https://a0.muscache.com/~
## $ host_neighbourhood                  <chr> "Hyde Park", "Hyde Park",~
## $ host_listings_count                 <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2~
## $ host_total_listings_count           <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2~
## $ host_verifications                  <chr> "['email', 'phone', 'revi~
## $ host_has_profile_pic                <lgl> TRUE, TRUE, TRUE, TRUE, T~
## $ host_identity_verified              <lgl> TRUE, TRUE, TRUE, TRUE, T~
## $ neighbourhood                       <chr> "Chicago, Illinois, Unite~
## $ neighbourhood_cleansed              <chr> "Hyde Park", "Hyde Park",~
## $ neighbourhood_group_cleansed        <lgl> NA, NA, NA, NA, NA, NA, N~
## $ latitude                            <dbl> 41.78790, 41.78790, 41.78~
## $ longitude                           <dbl> -87.58780, -87.58780, -87~
## $ property_type                       <chr> "Private room in condomin~
## $ room_type                           <chr> "Private room", "Private ~
## $ accommodates                        <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2~
## $ bathrooms                           <lgl> NA, NA, NA, NA, NA, NA, N~
## $ bathrooms_text                      <chr> "1 shared bath", "1 share~
## $ bedrooms                            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ beds                                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ amenities                           <chr> "[\"Hot water kettle\", \~
## $ price                               <dbl> 85, 85, 85, 85, 85, 65, 6~
## $ minimum_nights                      <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2~
## $ maximum_nights                      <dbl> 90, 90, 90, 90, 90, 60, 6~
## $ minimum_minimum_nights              <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2~
## $ maximum_minimum_nights              <dbl> 4, 4, 4, 4, 4, 2, 2, 2, 2~
## $ minimum_maximum_nights              <dbl> 90, 90, 90, 90, 90, 1125,~
## $ maximum_maximum_nights              <dbl> 90, 90, 90, 90, 90, 1125,~
## $ minimum_nights_avg_ntm              <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2~
## $ maximum_nights_avg_ntm              <dbl> 90, 90, 90, 90, 90, 1125,~
## $ calendar_updated                    <lgl> NA, NA, NA, NA, NA, NA, N~
## $ has_availability                    <lgl> TRUE, TRUE, TRUE, TRUE, T~
## $ availability_30                     <dbl> 10, 10, 10, 10, 10, 1, 1,~
## $ availability_60                     <dbl> 33, 33, 33, 33, 33, 11, 1~
## $ availability_90                     <dbl> 63, 63, 63, 63, 63, 32, 3~
```

```
## $ availability_365                                <dbl> 338, 338, 338, 338, 338, ~
## $ calendar_last_scraped                           <chr> "7/11/2021", "7/11/2021",~
## $ number_of_reviews                               <dbl> 185, 185, 185, 185, 185, ~
## $ number_of_reviews_ltm                           <dbl> 7, 7, 7, 7, 7, 17, 17, 17~
## $ number_of_reviews_l30d                          <dbl> 2, 2, 2, 2, 2, 0, 0, 0, 0~
## $ first_review                                    <chr> "4/30/2015", "4/30/2015",~
## $ last_review                                     <chr> "6/21/2021", "6/21/2021",~
## $ review_scores_rating                            <dbl> 4.99, 4.99, 4.99, 4.99, 4~
## $ review_scores_accuracy                          <dbl> 4.98, 4.98, 4.98, 4.98, 4~
## $ review_scores_cleanliness                       <dbl> 4.99, 4.99, 4.99, 4.99, 4~
## $ review_scores_checkin                           <dbl> 4.98, 4.98, 4.98, 4.98, 4~
## $ review_scores_communication                     <dbl> 4.98, 4.98, 4.98, 4.98, 4~
## $ review_scores_location                          <dbl> 4.95, 4.95, 4.95, 4.95, 4~
## $ review_scores_value                             <dbl> 4.94, 4.94, 4.94, 4.94, 4~
## $ license                                         <chr> "R17000015609", "R1700001~
## $ instant_bookable                                <lgl> FALSE, FALSE, FALSE, FALS~
## $ calculated_host_listings_count                  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ calculated_host_listings_count_entire_homes     <dbl> 0, 0, 0, 0, 0, 1, 1, 1, 1~
## $ calculated_host_listings_count_private_rooms    <dbl> 1, 1, 1, 1, 1, 0, 0, 0, 0~
## $ calculated_host_listings_count_shared_rooms     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ reviews_per_month                               <dbl> 2.45, 2.45, 2.45, 2.45, 2~
## $ Community.Area.Number                           <int> 41, 41, 41, 41, 41, 24, 2~
## $ Property.Type                                   <chr> "ARO", "ARO", "ARO", "ARO~
## $ Property.Name                                   <chr> "City Hyde Park", "Vue53"~
## $ Address                                         <chr> "5105 S. Harper Ave.", "1~
## $ Zip.Code                                        <int> 60615, 60615, 60615, 6061~
## $ Phone.Number                                    <chr> "773-548-5077", "773-355-~
## $ Management.Company                              <chr> "Mac Properties", "Peak C~
## $ Units                                           <int> 36, 27, 2, 10, 36, 3, 52,~
## $ X.Coordinate                                    <dbl> 1187194, 1185905, 1186745~
## $ Y.Coordinate                                    <dbl> 1871413, 1870431, 1870452~
## $ Latitude                                        <dbl> 41.80226, 41.79960, 41.79~
## $ Longitude                                       <dbl> -87.58900, -87.59376, -87~
## $ Location                                        <chr> "(41.8022605698632, -87.5~
## $ `Average Rent`                                  <dbl> 1450, 1450, 1450, 1450, 1~
```

#By looking at the data we can say that #Airbnb data # 1. Variable id is just an identifier and we can ignore it. # 2. We can factor the field room.type - Private room,Entire home/apt,Hotel # room, Shared room # 3. We can drop the host.id and host.name,neighbourhood.group,name fields # from the dataset # 4. We can drop fields like last.review,number.of.reviews, # reviews.per.month,calculated.host.listings.count

#Average rent Chicago neighborhood data # 5. We can drop Property Name,Phone Number,Management Company,Units,Zip Codes # from the # dataset

#Average rent Chicago neighborhood data # 6. rename the Average Rent to Average_Rent

```
# Apply above transformation to the dataframe

final_df <- subset(final_2_df, select = c("neighbourhood_cleansed", "latitude",
                                          "longitude", "room_type", "price","minimum_nights", "availabi
                                          "Zip.Code","X.Coordinate", "Y.Coordinate", "Latitude","Longitu
glimpse(final_df)
```

```
## Rows: 78,313
```

```
## Columns: 14
## $ neighbourhood_cleansed <chr> "Hyde Park", "Hyde Park", "Hyde Park", "Hyde Pa~
## $ latitude               <dbl> 41.78790, 41.78790, 41.78790, 41.78790, 41.7879~
## $ longitude              <dbl> -87.58780, -87.58780, -87.58780, -87.58780, -87~
## $ room_type              <chr> "Private room", "Private room", "Private room",~
## $ price                  <dbl> 85, 85, 85, 85, 85, 65, 65, 65, 65, 65, 65, 65,~
## $ minimum_nights         <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,~
## $ availability_365       <dbl> 338, 338, 338, 338, 338, 59, 59, 59, 59, 59, 59~
## $ property_type          <chr> "Private room in condominium", "Private room in~
## $ Zip.Code               <int> 60615, 60615, 60615, 60615, 60615, 60642, 60647~
## $ X.Coordinate           <dbl> 1187194, 1185905, 1186745, 1185103, 1187148, 11~
## $ Y.Coordinate           <dbl> 1871413, 1870431, 1870452, 1869464, 1870068, 19~
## $ Latitude               <dbl> 41.80226, 41.79960, 41.79963, 41.79696, 41.7985~
## $ Longitude              <dbl> -87.58900, -87.59376, -87.59068, -87.59674, -87~
## $ `Average Rent`         <dbl> 1450, 1450, 1450, 1450, 1450, 1600, 1600, 1600,~
```

```
#Rename Average Rent to Average_Rent
colnames(final_df)[14] <- "Average_Rent"

# Checking the summary of data set to gauge the value range of each numerical
# variable

summary(final_df)
```

```
##  neighbourhood_cleansed    latitude       longitude       room_type
##  Length:78313           Min.   :41.66   Min.   :-87.84   Length:78313
##  Class :character       1st Qu.:41.88   1st Qu.:-87.69   Class :character
##  Mode  :character       Median :41.90   Median :-87.67   Mode  :character
##                         Mean   :41.90   Mean   :-87.67
##                         3rd Qu.:41.92   3rd Qu.:-87.64
##                         Max.   :42.02   Max.   :-87.54
##
##      price        minimum_nights    availability_365 property_type
##  Min.   :   0.0   Min.   :  1.000   Min.   :  0.0    Length:78313
##  1st Qu.:  79.0   1st Qu.:  1.000   1st Qu.: 19.0    Class :character
##  Median : 122.0   Median :  2.000   Median :134.0    Mode  :character
##  Mean   : 182.1   Mean   :  8.943   Mean   :158.2
##  3rd Qu.: 196.0   3rd Qu.:  4.000   3rd Qu.:302.0
##  Max.   :9999.0   Max.   :365.000   Max.   :365.0
##
##     Zip.Code      X.Coordinate      Y.Coordinate        Latitude
##  Min.   :60601   Min.   :1127329   Min.   :1824810   Min.   :41.67
##  1st Qu.:60612   1st Qu.:1158284   1st Qu.:1901307   1st Qu.:41.88
##  Median :60622   Median :1165062   Median :1908210   Median :41.90
##  Mean   :60655   Mean   :1164788   Mean   :1906400   Mean   :41.90
##  3rd Qu.:60647   3rd Qu.:1170456   3rd Qu.:1912027   3rd Qu.:41.91
##  Max.   :66007   Max.   :1199523   Max.   :1949531   Max.   :42.02
##  NA's   :120     NA's   :135       NA's   :135       NA's   :135
##    Longitude       Average_Rent
##  Min.   :-87.81   Min.   : 675
##  1st Qu.:-87.69   1st Qu.:1299
##  Median :-87.67   Median :1600
##  Mean   :-87.67   Mean   :1605
##  3rd Qu.:-87.65   3rd Qu.:2200
```

```
##  Max.   :-87.54   Max.   :2350
##  NA's   :135
```

7. Range of values prices are varies from 0 to 10000.

It looks like there are outliers in the field.

8. Range of values minimum_nights varies from 1 to 365.

It looks like there are outliers in the field.

9. Range of values for availability_365 varies from 0 to 365.

10. Range of values for Average_Rent varies from 675 to 2350.

```
#Calculate the 30 days price for airbnb property.
final_df$airbnb_30_days_price=final_df$price * 30
summary(final_df)
```

```
##  neighbourhood_cleansed    latitude       longitude       room_type
##  Length:78313           Min.   :41.66   Min.   :-87.84   Length:78313
##  Class :character       1st Qu.:41.88   1st Qu.:-87.69   Class :character
##  Mode  :character       Median :41.90   Median :-87.67   Mode  :character
##                         Mean   :41.90   Mean   :-87.67
##                         3rd Qu.:41.92   3rd Qu.:-87.64
##                         Max.   :42.02   Max.   :-87.54
##
##      price        minimum_nights    availability_365 property_type
##  Min.   :   0.0   Min.   :  1.000   Min.   :  0.0    Length:78313
##  1st Qu.:  79.0   1st Qu.:  1.000   1st Qu.: 19.0    Class :character
##  Median : 122.0   Median :  2.000   Median :134.0    Mode  :character
##  Mean   : 182.1   Mean   :  8.943   Mean   :158.2
##  3rd Qu.: 196.0   3rd Qu.:  4.000   3rd Qu.:302.0
##  Max.   :9999.0   Max.   :365.000   Max.   :365.0
##
##     Zip.Code      X.Coordinate      Y.Coordinate        Latitude
##  Min.   :60601   Min.   :1127329   Min.   :1824810   Min.   :41.67
##  1st Qu.:60612   1st Qu.:1158284   1st Qu.:1901307   1st Qu.:41.88
##  Median :60622   Median :1165062   Median :1908210   Median :41.90
##  Mean   :60655   Mean   :1164788   Mean   :1906400   Mean   :41.90
##  3rd Qu.:60647   3rd Qu.:1170456   3rd Qu.:1912027   3rd Qu.:41.91
##  Max.   :66007   Max.   :1199523   Max.   :1949531   Max.   :42.02
##  NA's   :120     NA's   :135       NA's   :135       NA's   :135
##    Longitude       Average_Rent  airbnb_30_days_price
##  Min.   :-87.81   Min.   : 675   Min.   :     0
##  1st Qu.:-87.69   1st Qu.:1299   1st Qu.:  2370
##  Median :-87.67   Median :1600   Median :  3660
```

```
##  Mean   :-87.67   Mean   :1605   Mean   :  5463
##  3rd Qu.:-87.65   3rd Qu.:2200   3rd Qu.:  5880
##  Max.   :-87.54   Max.   :2350   Max.   :299970
##  NA's   :135
```

```
#Check missing values
apply(final_df, 2, function(x) any(is.na(x)))
```

```
## neighbourhood_cleansed               latitude              longitude
##                  FALSE                  FALSE                  FALSE
##              room_type                  price         minimum_nights
##                  FALSE                  FALSE                  FALSE
##       availability_365          property_type               Zip.Code
##                  FALSE                  FALSE                   TRUE
##           X.Coordinate           Y.Coordinate               Latitude
##                   TRUE                   TRUE                   TRUE
##              Longitude           Average_Rent   airbnb_30_days_price
##                   TRUE                  FALSE                  FALSE
```

#It looks like there are some missing values for #X.Coordinate ,Y.Coordinate, Latitude, Longitude, Zip.Code

```
## 2.What does the final data set look like?

glimpse(final_df)
```

```
## Rows: 78,313
## Columns: 15
## $ neighbourhood_cleansed <chr> "Hyde Park", "Hyde Park", "Hyde Park", "Hyde Pa~
## $ latitude               <dbl> 41.78790, 41.78790, 41.78790, 41.78790, 41.7879~
## $ longitude              <dbl> -87.58780, -87.58780, -87.58780, -87.58780, -87~
## $ room_type              <chr> "Private room", "Private room", "Private room",~
## $ price                  <dbl> 85, 85, 85, 85, 85, 65, 65, 65, 65, 65, 65, 65,~
## $ minimum_nights         <dbl> 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,~
## $ availability_365       <dbl> 338, 338, 338, 338, 338, 59, 59, 59, 59, 59, 59~
## $ property_type          <chr> "Private room in condominium", "Private room in~
## $ Zip.Code               <int> 60615, 60615, 60615, 60615, 60615, 60642, 60647~
## $ X.Coordinate           <dbl> 1187194, 1185905, 1186745, 1185103, 1187148, 11~
## $ Y.Coordinate           <dbl> 1871413, 1870431, 1870452, 1869464, 1870068, 19~
## $ Latitude               <dbl> 41.80226, 41.79960, 41.79963, 41.79696, 41.7985~
## $ Longitude              <dbl> -87.58900, -87.59376, -87.59068, -87.59674, -87~
## $ Average_Rent           <dbl> 1450, 1450, 1450, 1450, 1450, 1600, 1600, 1600,~
## $ airbnb_30_days_price   <dbl> 2550, 2550, 2550, 2550, 2550, 1950, 1950, 1950,~
```

3. Questions for future steps.

a) Need to learn how to visualize more than two variables.

b) Need to learn application of variable scaling and techniques.

c) Need to learn how lm() function takes care of variable scaling.

d) Need to learn correlation between different variables.

4.What information is not self-evident?

To uncover new information in the data that is not self-evident -

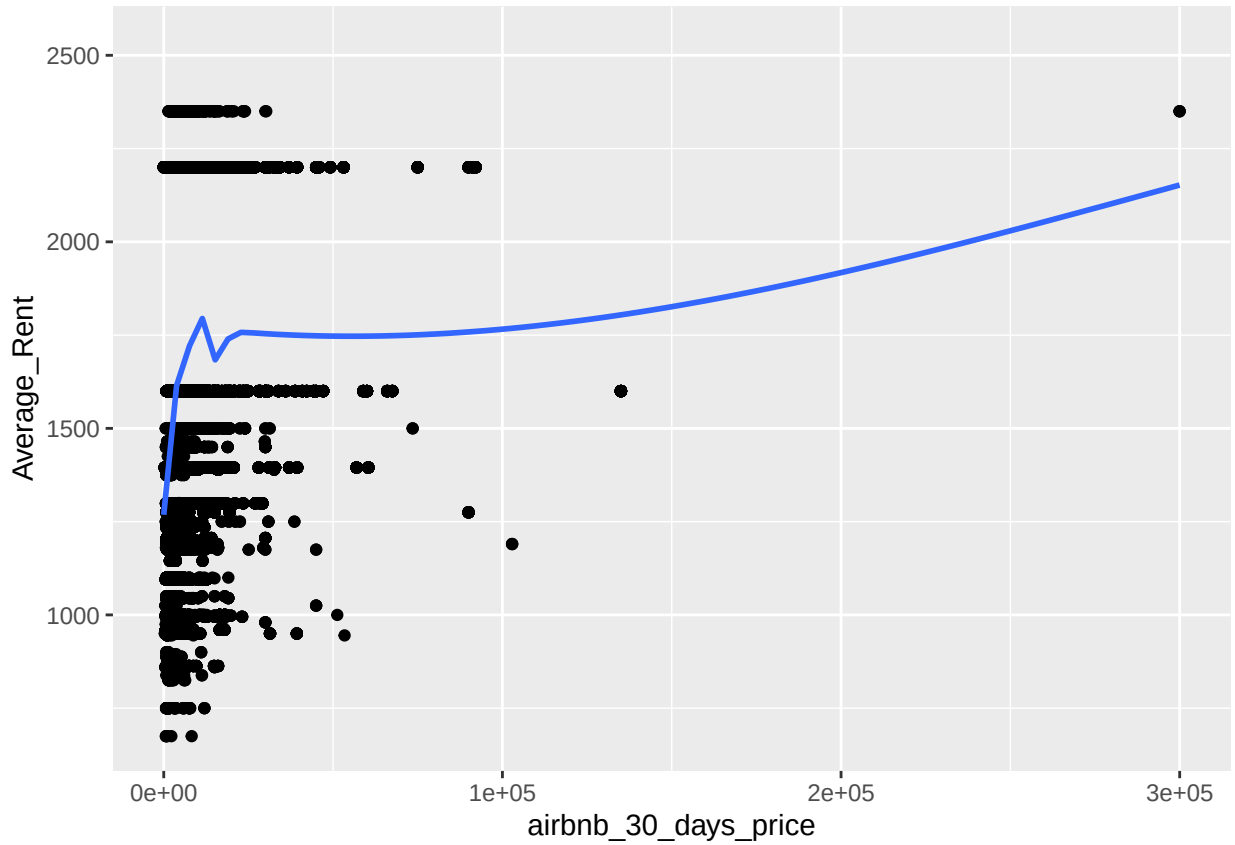
1. visualize data to uncover patterns and trends
2. correlation among variables
3. Check data distribution of variables
4. detect outliers and influencial cases

5.What are different ways you could look at this data?

```
# Checking relation between airbnb_30_days_price and Average_Rent using

ggplot()
```

```
library(ggplot2)
ggplot(data = final_df, aes(x = airbnb_30_days_price,
                            y = Average_Rent)) + geom_point() +
  geom_smooth(fill=NA)
```

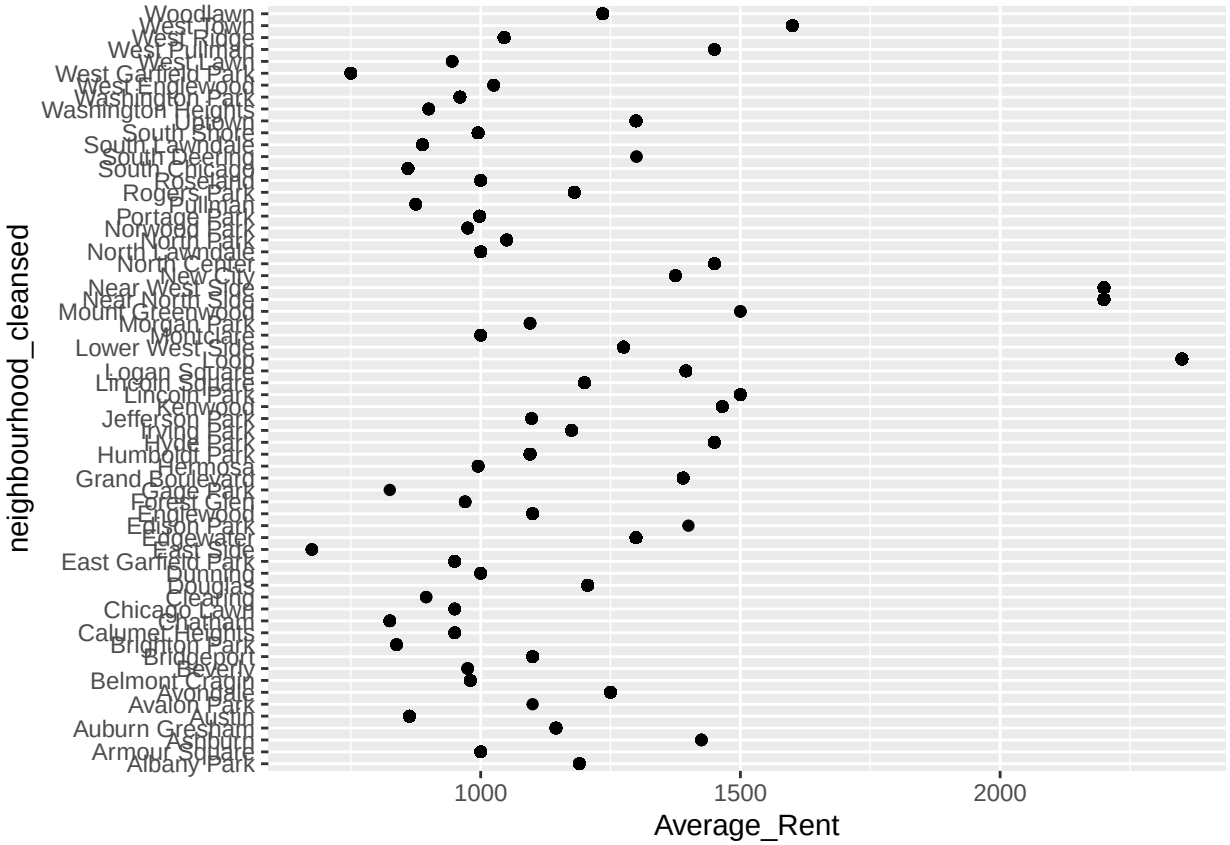
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
# Checking relation between neighbourhood_cleansed and Average_Rent using

ggplot()
```

```
library(ggplot2)
ggplot(data = final_df, aes(y = neighbourhood_cleansed, x = Average_Rent))+ geom_point() + geom_smooth(
```

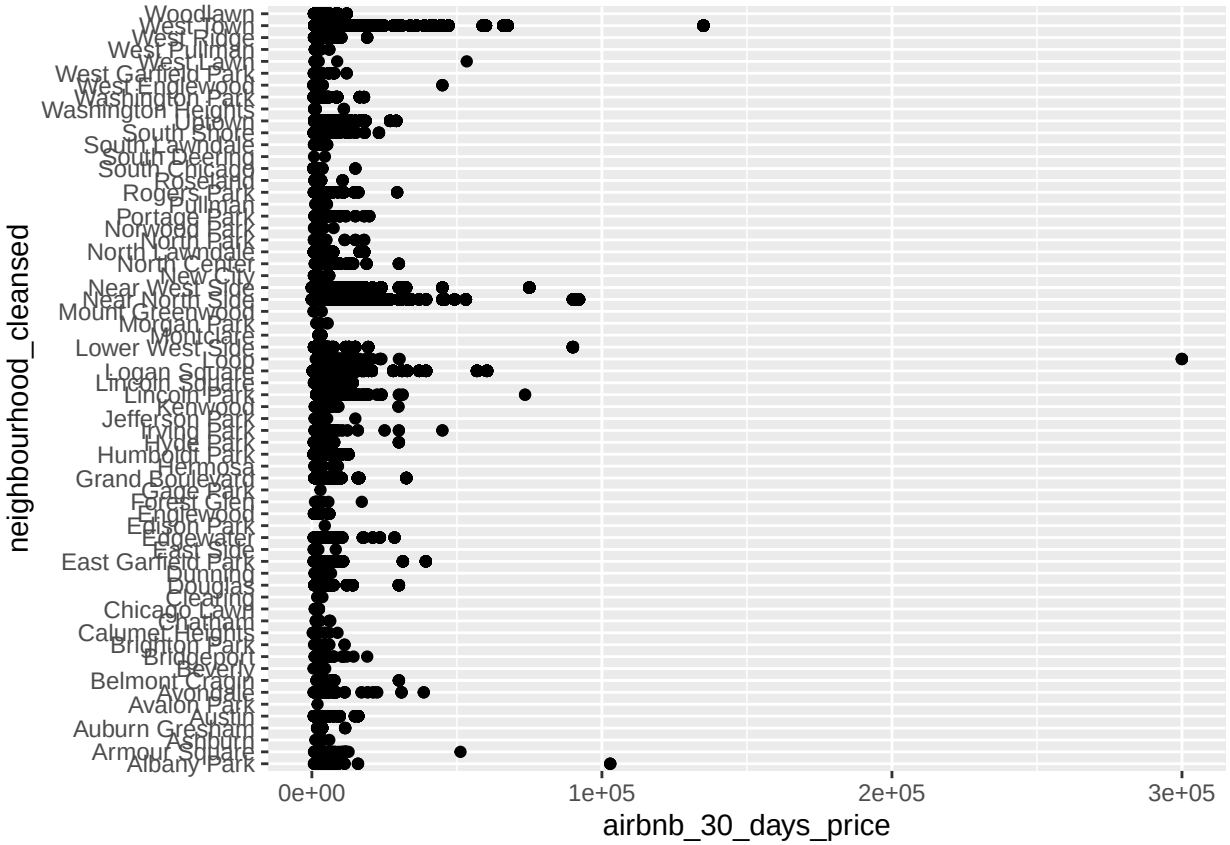
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
# Checking relation between neighbourhood_cleansed and airbnb_30_days_price using

ggplot()
```

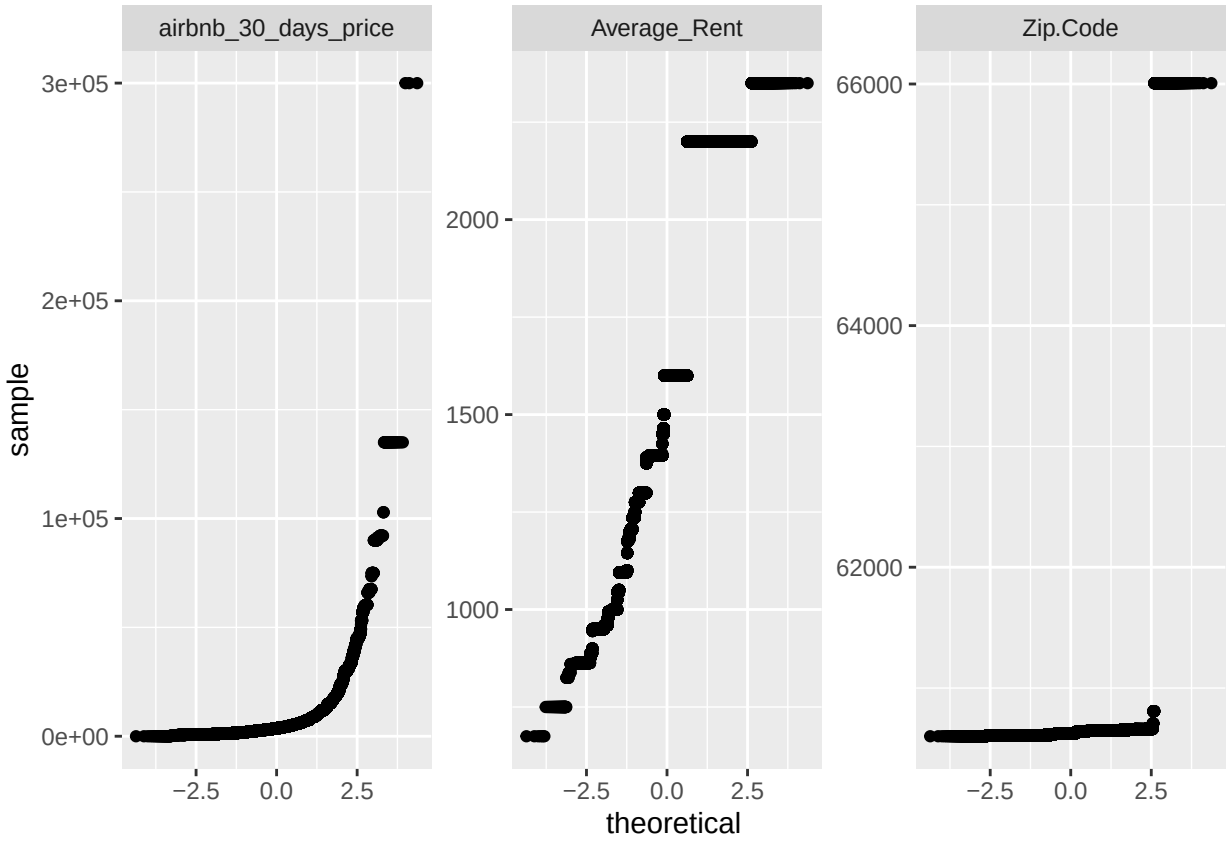
```
library(ggplot2)
ggplot(data = final_df, aes(y = neighbourhood_cleansed,
                            x = airbnb_30_days_price)) + geom_point() + geom_smooth(fill=NA)
```

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

## Warning: Computation failed in `stat_smooth()`
## Caused by error in `gam.reparam()`:
## ! NA/NaN/Inf in foreign function call (arg 3)
```

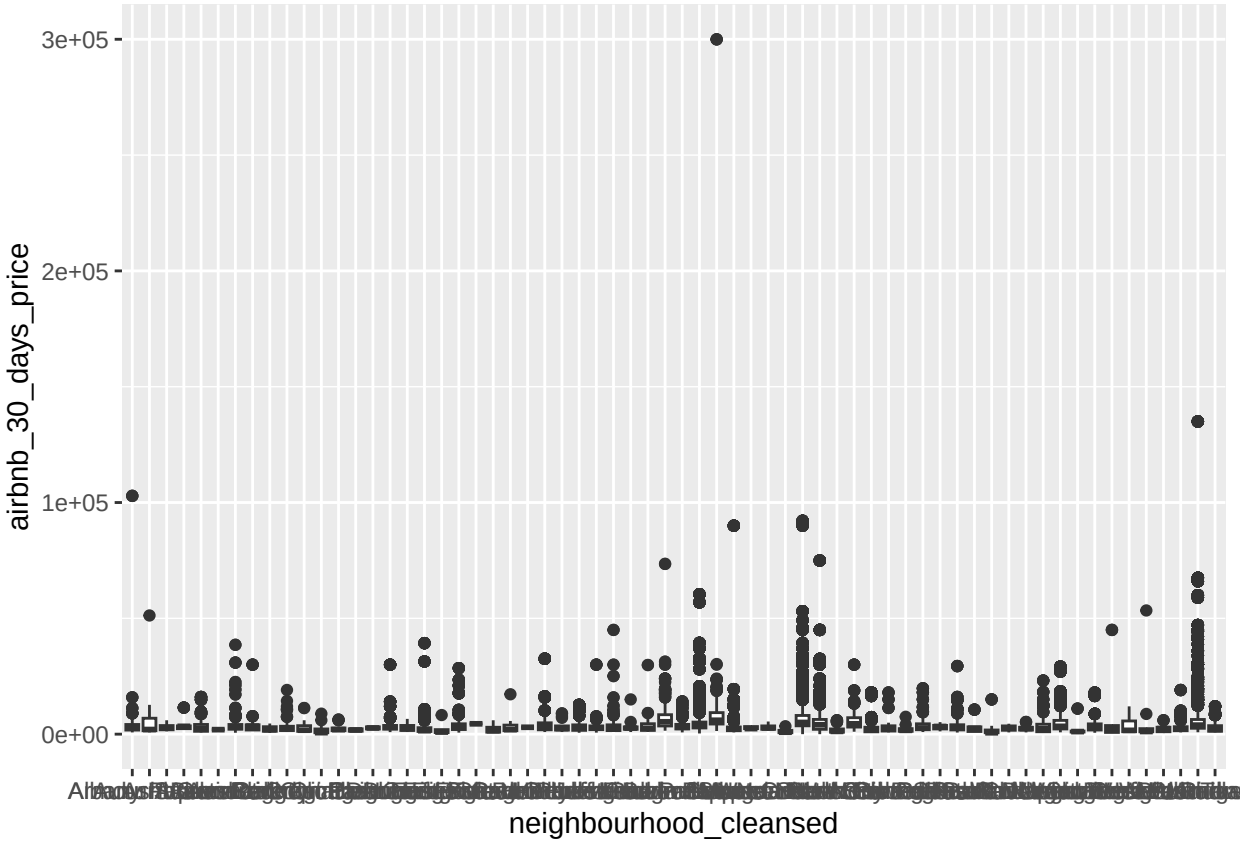



```
#We can see that there is relationship between neighbourhood and prices
# Checking if data distribution of numeric variables is normal
# combining pipe operator between dplyr transformation and ggplot
final_df %>% select(airbnb_30_days_price, Zip.Code, Average_Rent) %>%
  gather()%>%
 ggplot(., aes(sample = value)) +
  stat_qq() +
  facet_wrap(vars(key), scales ='free_y')
```

```
## Warning: Removed 120 rows containing non-finite values (‘stat_qq()‘).
```

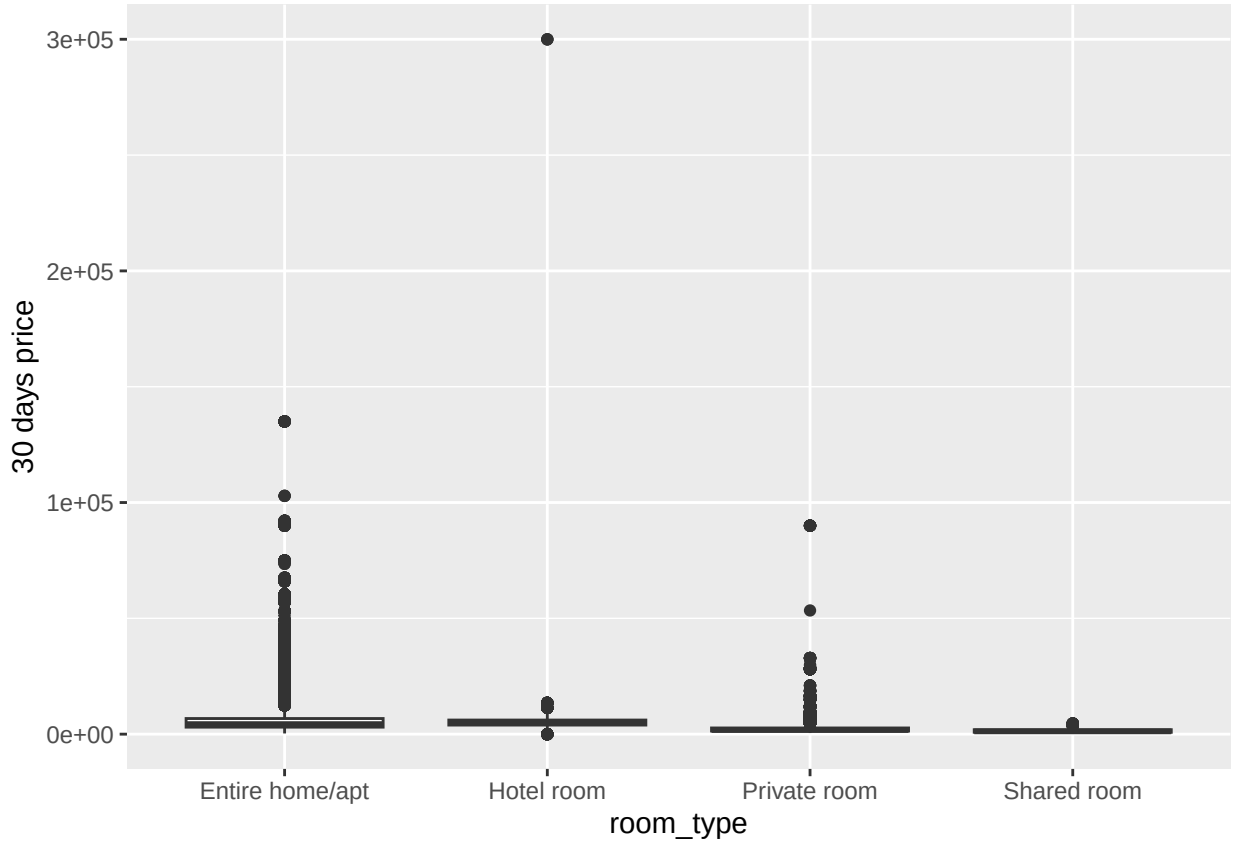



```
#None of the variables looks normally distributed
ggplot(data = final_df, aes(x = neighbourhood_cleansed , y = airbnb_30_days_price)) +
  geom_boxplot() + ylab("airbnb_30_days_price")
```



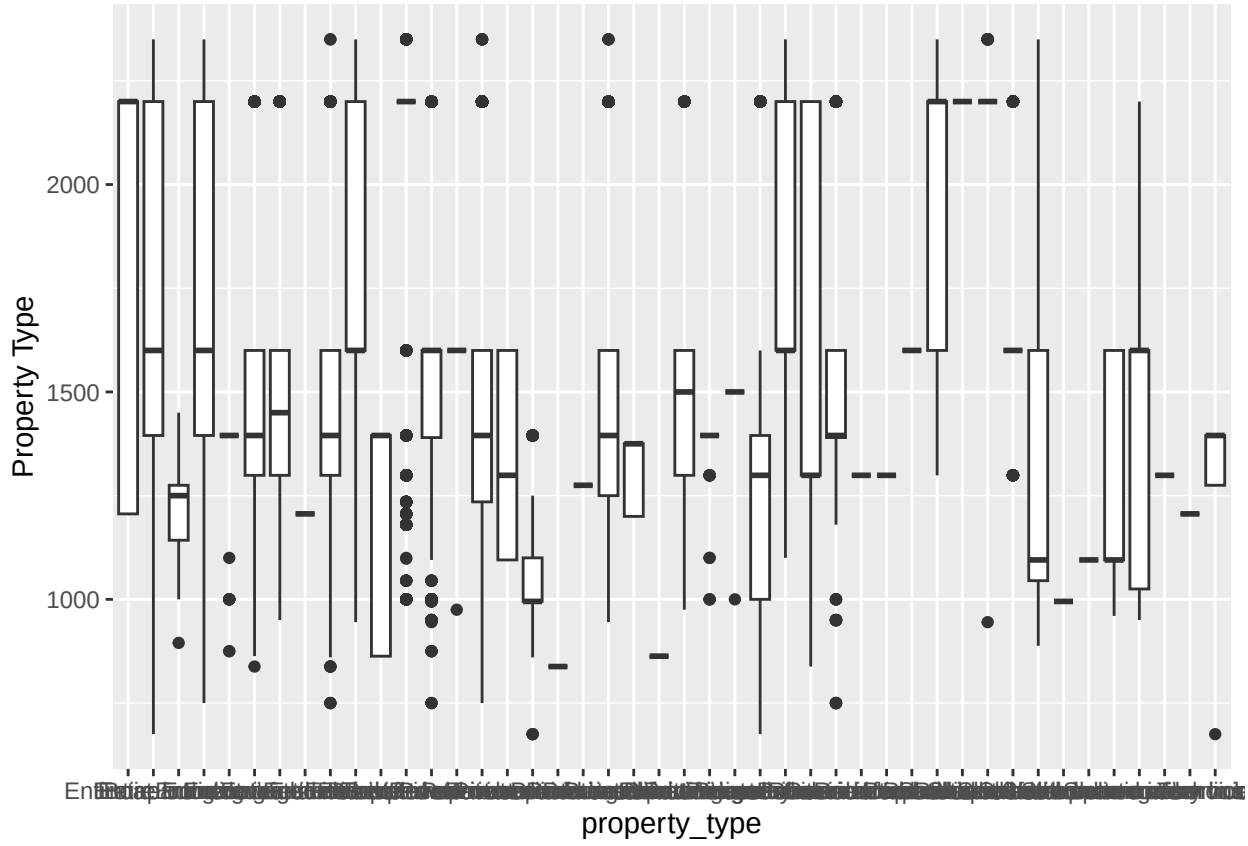
We can see that there are so many outliers for many neighbourhoods # thus data is not normally distributed

```
ggplot(data = final_df, aes(x = room_type , y = airbnb_30_days_price)) +
  geom_boxplot() + ylab("30 days price")
```



We can see that there are so many outliers for room_type thus data is not normally distributed

```
ggplot(data = final_df, aes(x = property_type , y = Average_Rent)) +
  geom_boxplot() + ylab("Property Type")
```



We can see that there are so many outliers for Property_Type thus data is not normally distributed

```
# 6.How do you plan to slice and dice the data?

unique(final_df[c("Zip.Code")])
```

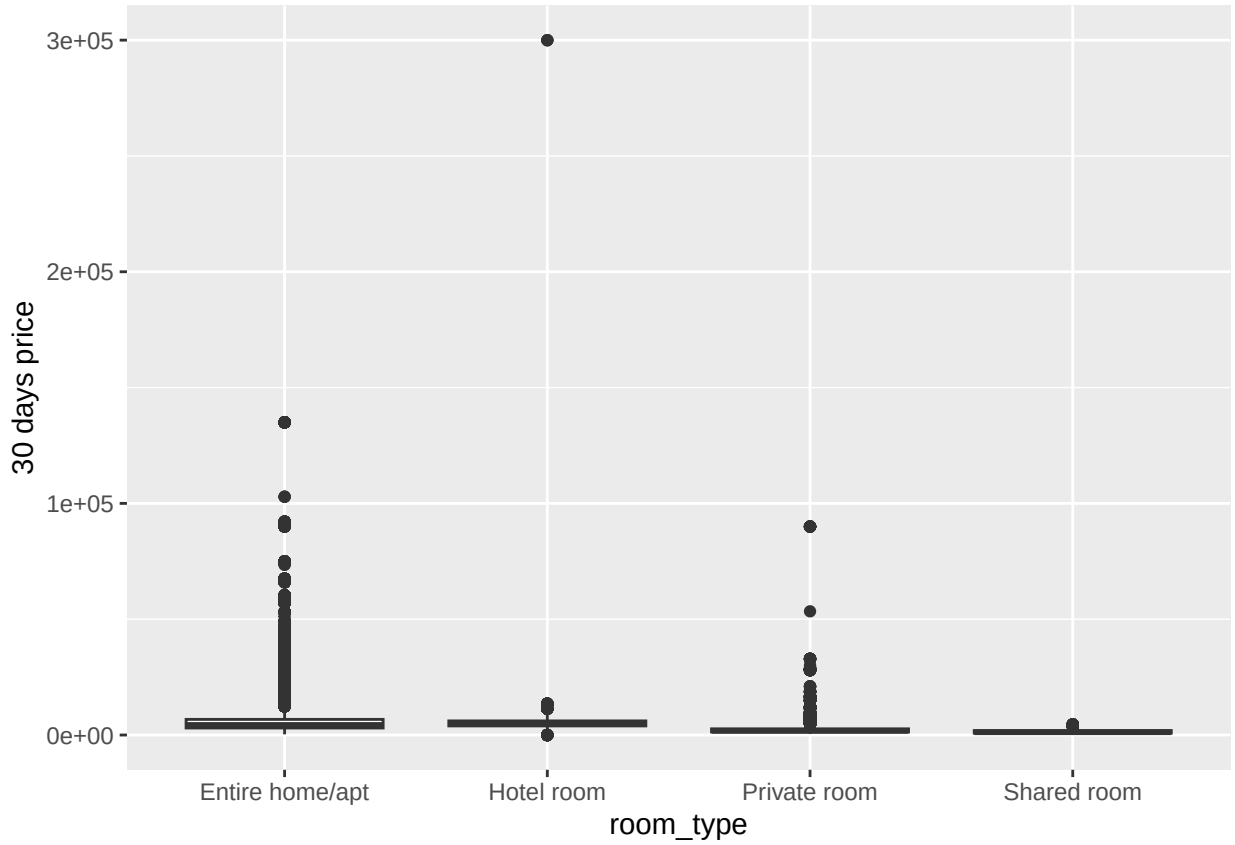
```
## # A tibble: 49 x 1
##    Zip.Code
##       <int>
##  1    60615
##  2    60642
##  3    60647
##  4    60622
##  5    60654
##  6    60614
##  7    60610
##  8    60612
##  9    60640
## 10    60613
## # i 39 more rows
```

```
unique(final_df[c("neighbourhood_cleansed")])
```

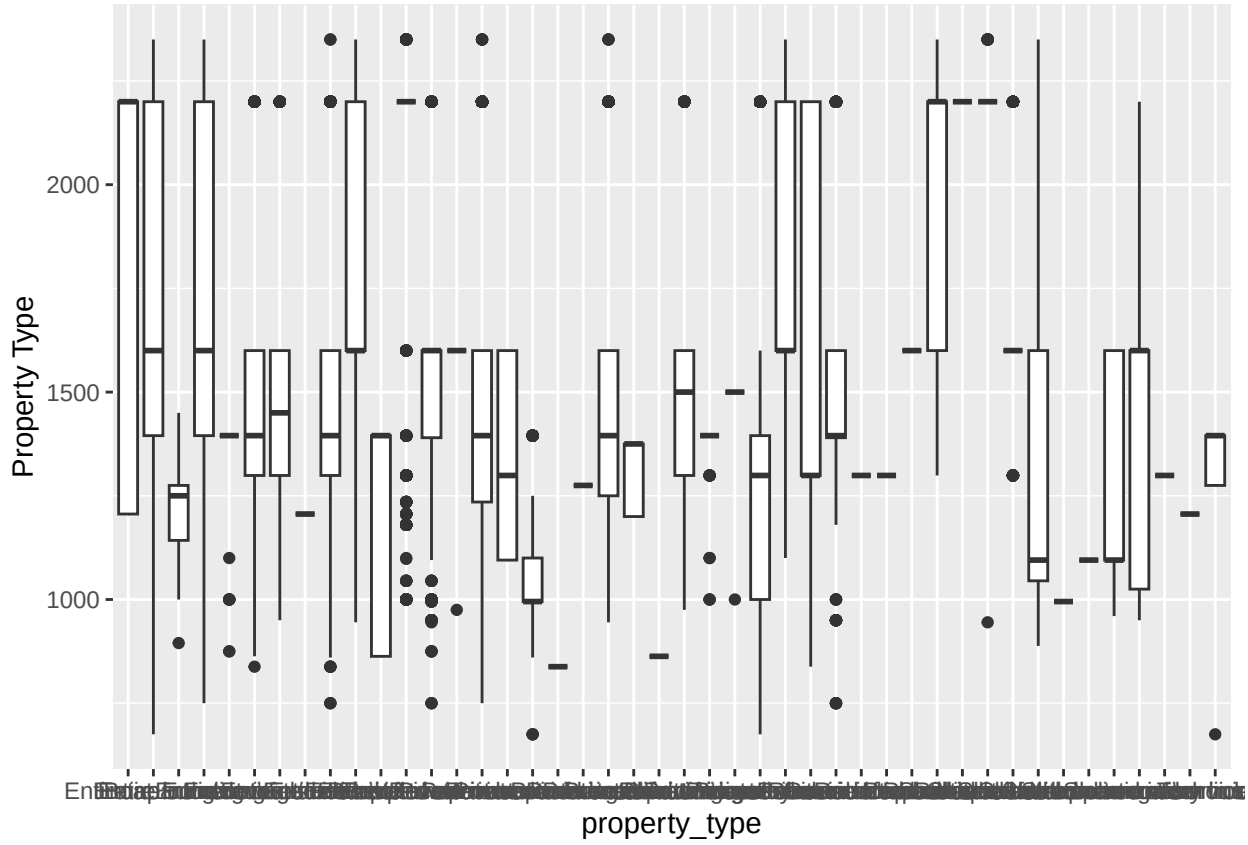
```
## # A tibble: 64 x 1
##    neighbourhood_cleansed
##    <chr>
##  1 Hyde Park
##  2 West Town
##  3 Lincoln Park
##  4 Near North Side
##  5 Logan Square
##  6 Uptown
##  7 North Center
##  8 Albany Park
##  9 Pullman
## 10 West Ridge
## # i 54 more rows
```

#I think need to slice the datasets by zip codes or neighbourhood to analyze # the data in more granular level # We can see that there are so many outliers for many neighbourhoods # thus data is not normally distributed

```
ggplot(data = final_df, aes(x = room_type , y = airbnb_30_days_price)) +
  geom_boxplot() + ylab("30 days price")
```



```
# We can see that there are so many outliers for room_type
# thus data is not normally distributed
ggplot(data = final_df, aes(x = property_type , y = Average_Rent)) +
  geom_boxplot() + ylab("Property Type")
```




```
# We can see that there are so many outliers for Property_Type
# thus data is not normally distributed

# 6.How do you plan to slice and dice the data?
unique(final_df[c("Zip.Code")])
```

```
## # A tibble: 49 x 1
##    Zip.Code
##       <int>
##  1    60615
##  2    60642
##  3    60647
##  4    60622
##  5    60654
##  6    60614
##  7    60610
##  8    60612
##  9    60640
## 10    60613
## # i 39 more rows
```

```r
unique(final_df[c("neighbourhood_cleansed")])
```

```
## # A tibble: 64 x 1
##    neighbourhood_cleansed
##    <chr>
##  1 Hyde Park
##  2 West Town
##  3 Lincoln Park
##  4 Near North Side
##  5 Logan Square
##  6 Uptown
##  7 North Center
##  8 Albany Park
##  9 Pullman
## 10 West Ridge
## # i 54 more rows
```

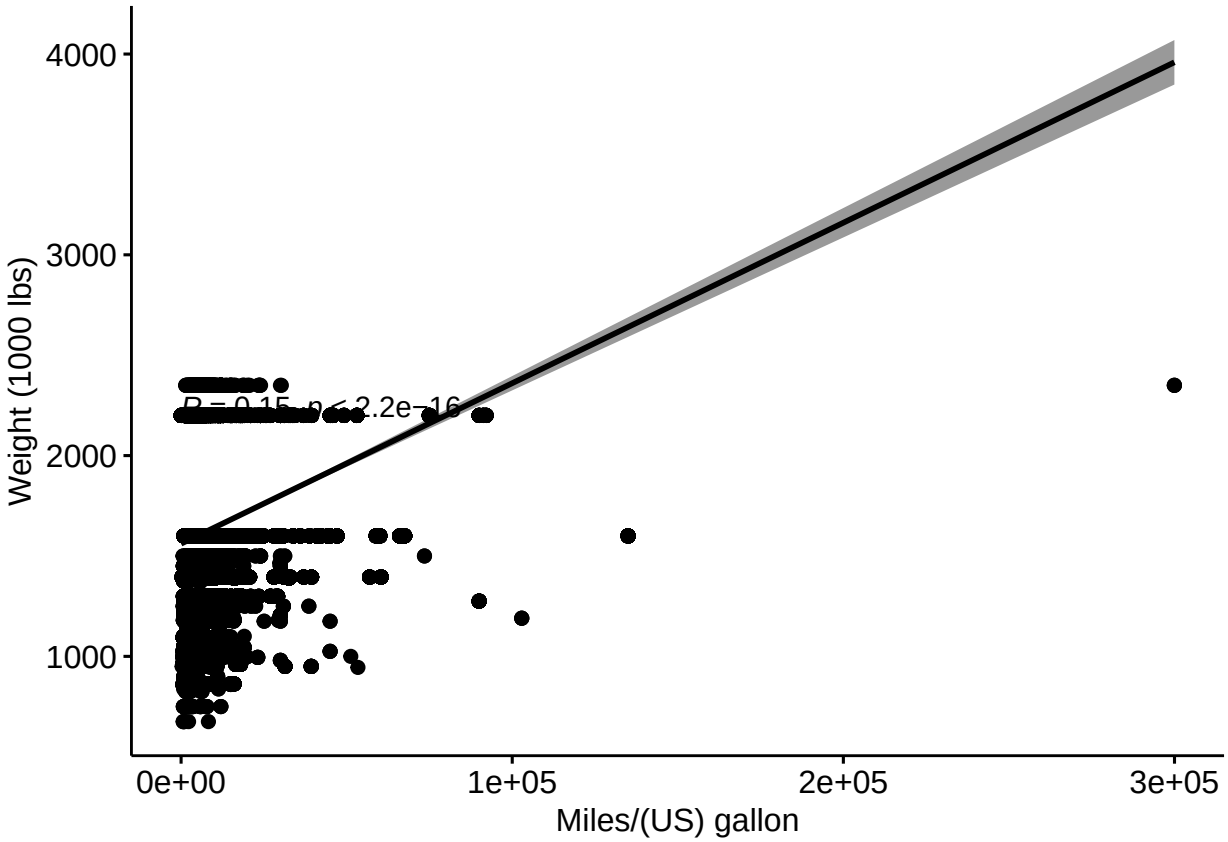
```r
#I think need to slice the datasets by zip codes or neighbourhood to analyze
# the data in more granular level

# 7.How could you summarize your data to answer key questions?
library("ggpubr")
```

```
## Warning: package 'ggpubr' was built under R version 4.2.3
```

```
##
## Attaching package: 'ggpubr'
```

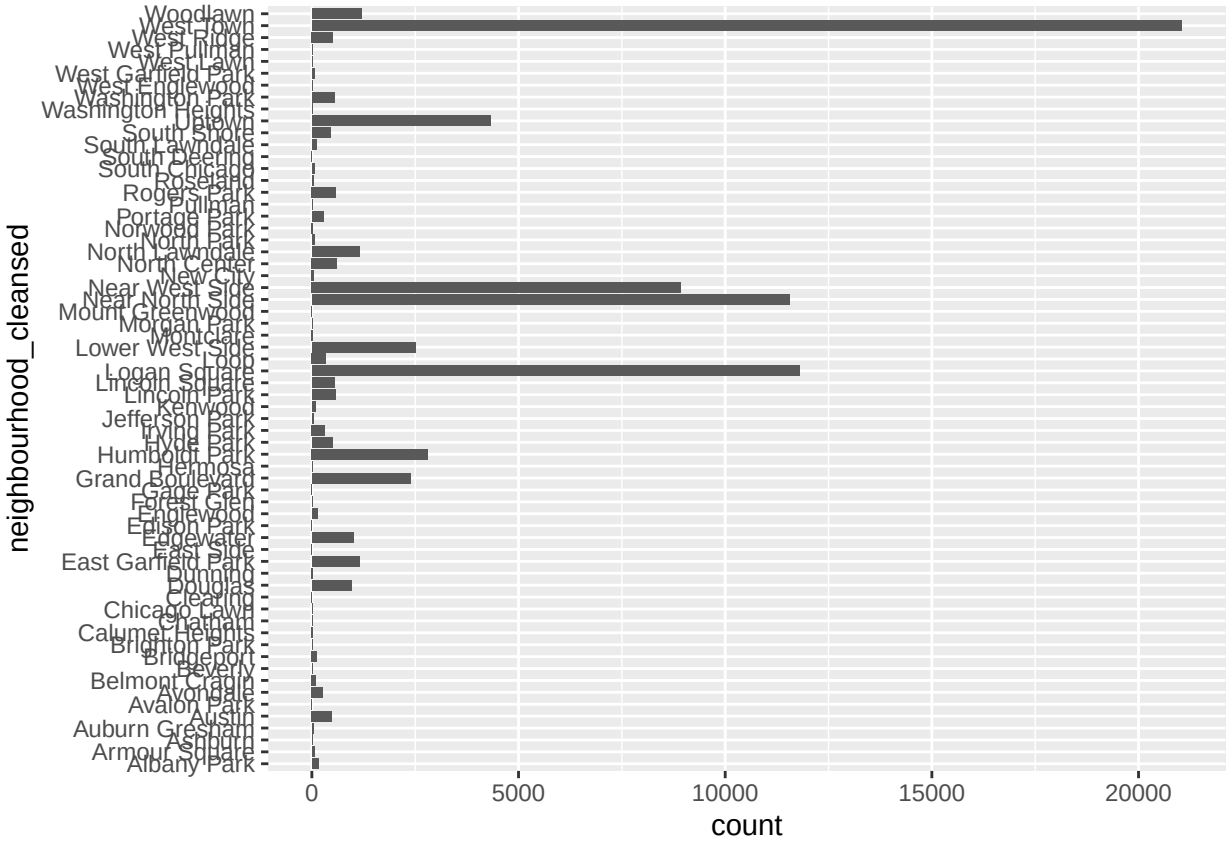
```
## The following object is masked from 'package:plyr':
##
##     mutate
```

```r
ggscatter(final_df, x = "airbnb_30_days_price", y = "Average_Rent",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Miles/(US) gallon", ylab = "Weight (1000 lbs)")
```

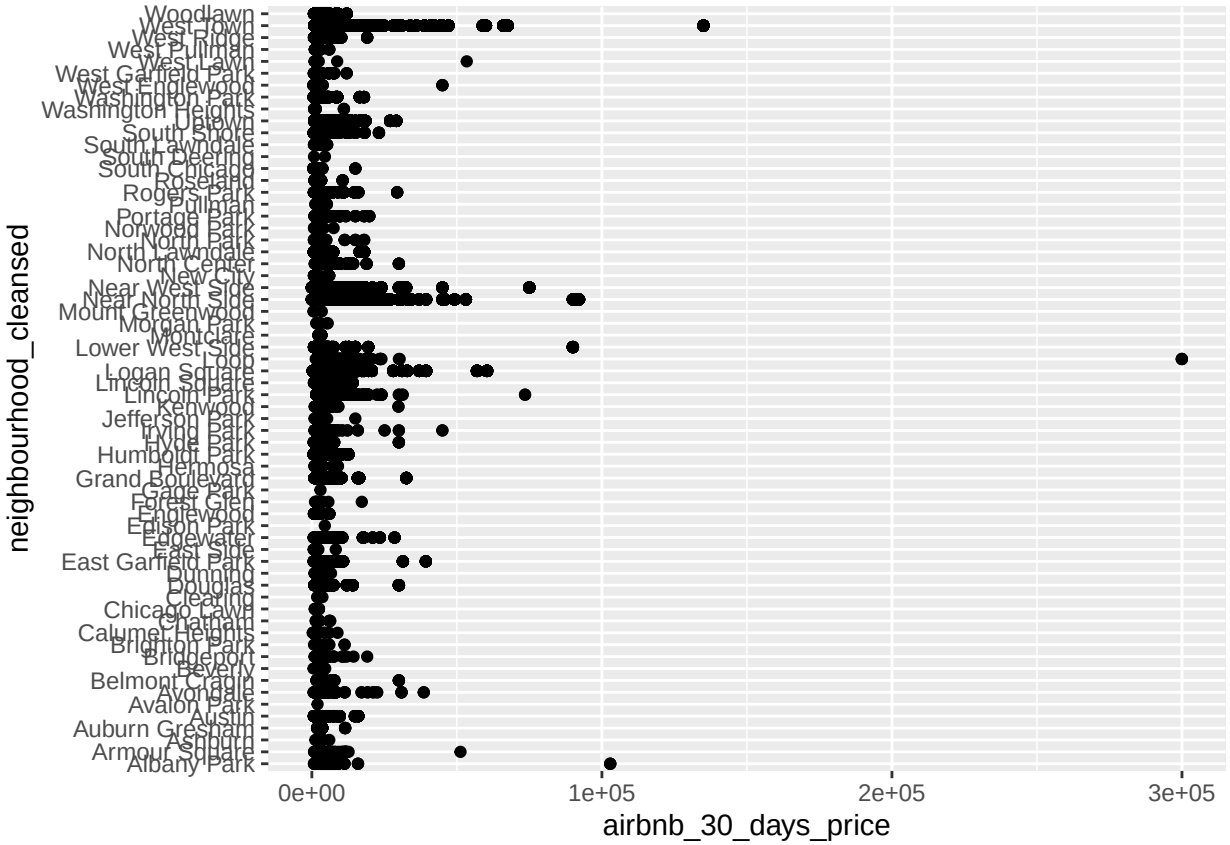



```
#a) What are the Airbnb rental prices for different areas in Chicago?
ggplot(data=final_df,aes(y=neighbourhood_cleansed)) + geom_histogram(stat = "count")
```

```
## Warning in geom_histogram(stat = "count"): Ignoring unknown parameters:
## `binwidth`, `bins`, and `pad`
```




```
ggplot(aes(y=neighbourhood_cleansed,x=airbnb_30_days_price),data=final_df)+
  geom_point()
```



From graph it looks like "West town" have major number of airbnb properties

Also the prices of "West town" properties are high for airbnb rental.

b) What is the correlation between the Airbnb rental prices and Chicago

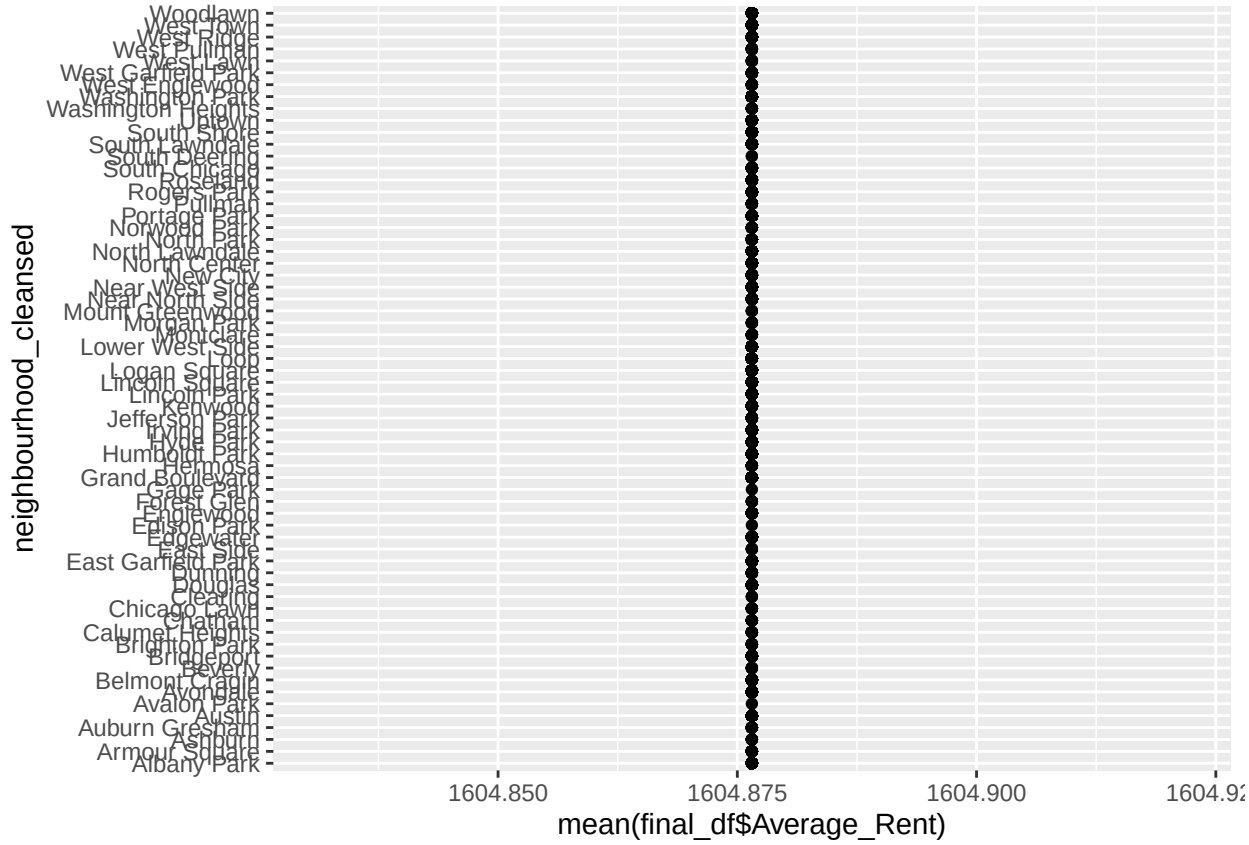
neighborhood rent prices?

```
cor(final_df$airbnb_30_days_price,final_df$Average_Rent)
```

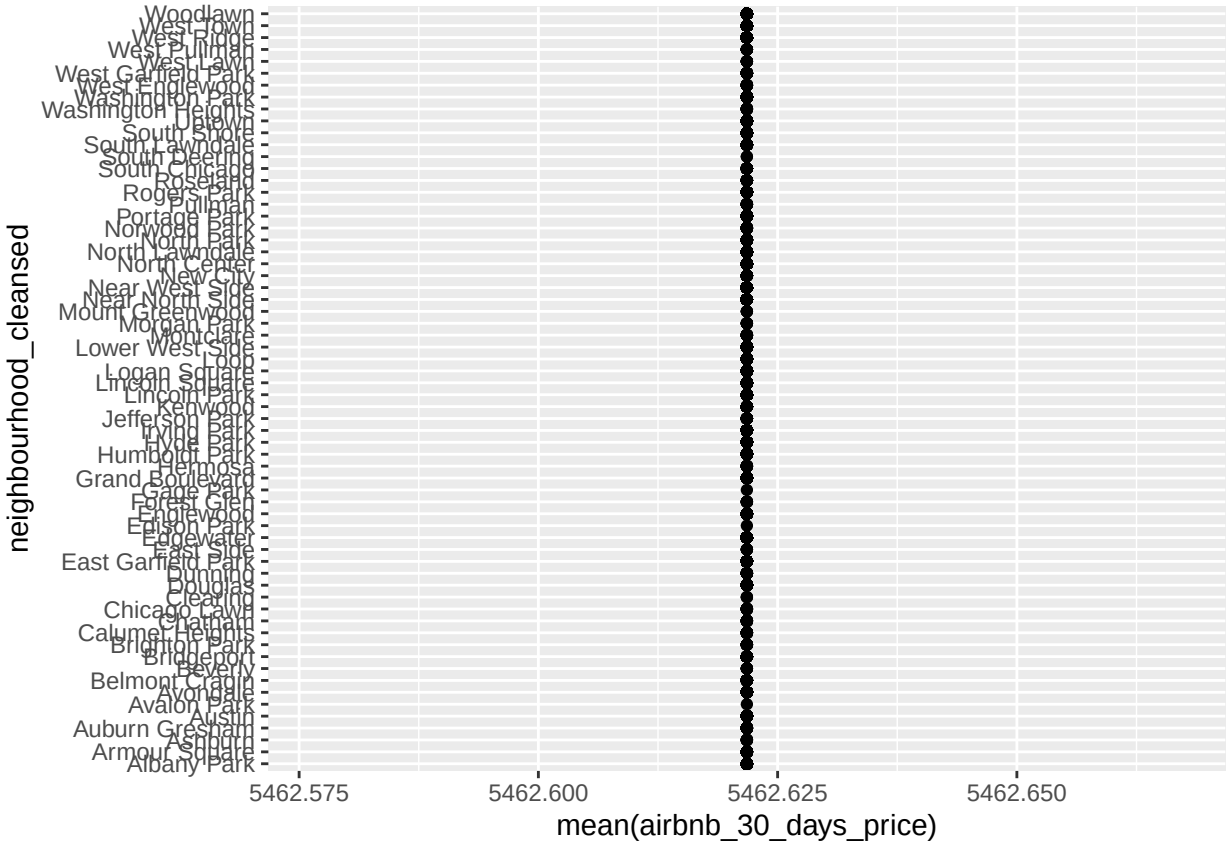
```
## [1] 0.1470344
```

```
# It is evident from the plots that there is positive correlation between
# airbnb prices and average rent

# c)What are the average rent prices by the neighborhood?
ggplot(aes(y=neighbourhood_cleansed,x=mean(final_df$Average_Rent)),data=final_df)+
 geom_point()
```

```
## Warning: Use of `final_df$Average_Rent` is discouraged.
## i Use `Average_Rent` instead.
```



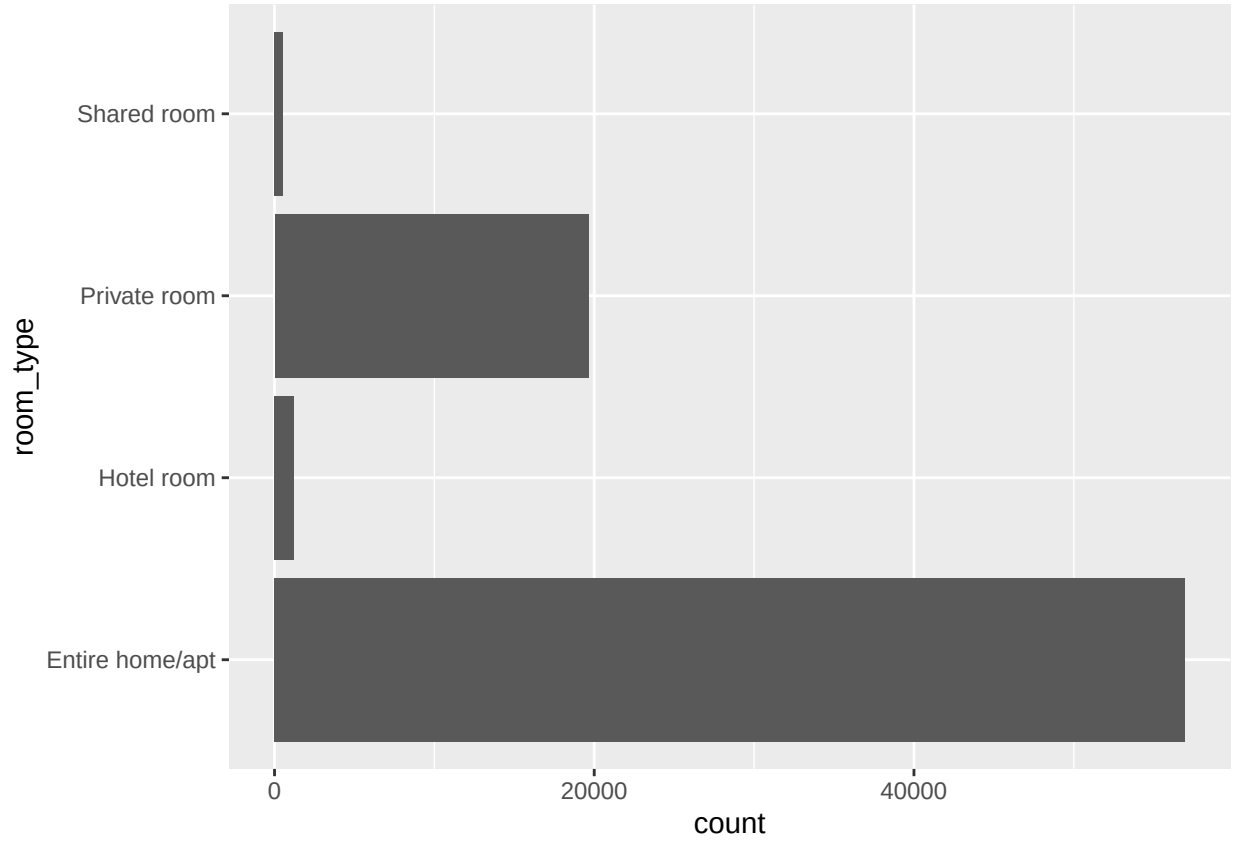

```
#The average rent price is ~1600 per month
# d)What are the average rent prices for Airbnb by the neighborhood?

ggplot(aes(y=neighbourhood_cleansed,x=mean(airbnb_30_days_price)),data=final_df)+
 geom_point()
```



#The average airbnb price is ~ 5400 per month

```
# e) What type of houses are most rented on Airbnb?
ggplot(data=final_df,aes(y=room_type)) + geom_histogram(stat ="count")
```

```
## Warning in geom_histogram(stat = "count"): Ignoring unknown parameters:
## `binwidth`, `bins`, and `pad`
```




```
#It looks like Entire home/apt are most rented on Airbnb

# f)What is the monthly rent from the Airbnb properties?
df1 <-final_df%>%select(neighbourhood_cleansed, airbnb_30_days_price, Average_Rent)
df1 %>% group_by(neighbourhood_cleansed) %>% summarize(mean_airbnb_30_days_price =
                                                         mean(airbnb_30_days_price))
```

```
##   mean_airbnb_30_days_price
## 1                  5462.622
```

#Airbnb monrthly average rent is 5462.622

9)Do you plan on incorporating any machine learning techniques to answer

your research questions? Explain.

performing multiple linear regression

splitting the data into training and test set

```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.2.3
```

```
mymodel_1 <-lm(airbnb_30_days_price ~ neighbourhood_cleansed,data = final_df)
summary(mymodel_1)
```

```
##
## Call:
## lm(formula = airbnb_30_days_price ~ neighbourhood_cleansed, data = final_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8592  -2980  -1513    342 290028
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             5483.21     573.47   9.561  < 2e-16
## neighbourhood_cleansedArmour Square     -408.37    1082.71  -0.377 0.706045
## neighbourhood_cleansedAshburn          -2558.21    2357.51  -1.085 0.277867
## neighbourhood_cleansedAuburn Gresham   -1158.21    1334.68  -0.868 0.385517
## neighbourhood_cleansedAustin           -2015.48     660.99  -3.049 0.002295
## neighbourhood_cleansedAvalon Park      -3533.21    7253.88  -0.487 0.626204
## neighbourhood_cleansedAvondale         -1028.10     722.87  -1.422 0.154957
## neighbourhood_cleansedBelmont Cragin     -776.21     922.91  -0.841 0.400327
## neighbourhood_cleansedBeverly          -3190.35    2792.64  -1.142 0.253286
## neighbourhood_cleansedBridgeport       -2362.47     870.33  -2.714 0.006640
## neighbourhood_cleansedBrighton Park    -2268.47    1755.27  -1.292 0.196230
## neighbourhood_cleansedCalumet Heights  -3804.36    1529.71  -2.487 0.012885
## neighbourhood_cleansedChatham          -2683.21    1798.29  -1.492 0.135681
## neighbourhood_cleansedChicago Lawn     -3705.71    1896.57  -1.954 0.050717
## neighbourhood_cleansedClearing         -2783.21    4214.12  -0.660 0.508969
## neighbourhood_cleansedDouglas          -1630.08     619.14  -2.633 0.008470
## neighbourhood_cleansedDunning          -2677.65    1505.17  -1.779 0.075248
## neighbourhood_cleansedEast Garfield Park -2425.49     611.67  -3.965 7.33e-05
## neighbourhood_cleansedEast Side        -3213.21    3007.30  -1.068 0.285312
## neighbourhood_cleansedEdgewater        -1515.71     617.04  -2.456 0.014036
## neighbourhood_cleansedEdison Park      -1013.21    7253.88  -0.140 0.888915
## neighbourhood_cleansedEnglewood        -3309.21     846.29  -3.910 9.23e-05
```

```
## neighbourhood_cleansedForest Glen        -1353.21    2477.67  -0.546 0.584957
## neighbourhood_cleansedGage Park          -2513.21    7253.88  -0.346 0.728995
## neighbourhood_cleansedGrand Boulevard    -1226.38     592.32  -2.070 0.038410
## neighbourhood_cleansedHermosa            -2099.21    1715.62  -1.224 0.221113
## neighbourhood_cleansedHumboldt Park      -2192.68     589.46  -3.720 0.000200
## neighbourhood_cleansedHyde Park          -2330.78     659.17  -3.536 0.000407
## neighbourhood_cleansedIrving Park        -1816.23     702.35  -2.586 0.009714
## neighbourhood_cleansedJefferson Park     -2425.93    1383.26  -1.754 0.079472
## neighbourhood_cleansedKenwood            -1838.21     934.64  -1.967 0.049215
## neighbourhood_cleansedLincoln Park        1511.69     647.80   2.334 0.019621
## neighbourhood_cleansedLincoln Square     -1749.84     650.84  -2.689 0.007177
## neighbourhood_cleansedLogan Square        -455.04     577.32  -0.788 0.430589
## neighbourhood_cleansedLoop                4458.63     694.09   6.424 1.34e-10
## neighbourhood_cleansedLower West Side    -1994.76     591.39  -3.373 0.000744
## neighbourhood_cleansedMontclare          -2765.21    1555.78  -1.777 0.075511
## neighbourhood_cleansedMorgan Park        -2426.96    2620.14  -0.926 0.354308
## neighbourhood_cleansedMount Greenwood    -4065.71    3660.79  -1.111 0.266739
## neighbourhood_cleansedNear North Side     1997.10     577.40   3.459 0.000543
## neighbourhood_cleansedNear West Side        62.76     578.55   0.108 0.913617
## neighbourhood_cleansedNew City           -3466.28    1155.18  -3.001 0.002695
## neighbourhood_cleansedNorth Center         109.69     644.44   0.170 0.864851
## neighbourhood_cleansedNorth Lawndale     -2449.68     611.64  -4.005 6.20e-05
## neighbourhood_cleansedNorth Park         -1515.38    1042.45  -1.454 0.146039
## neighbourhood_cleansedNorwood Park       -3511.78    1482.01  -2.370 0.017810
## neighbourhood_cleansedPortage Park       -1580.67     716.23  -2.207 0.027322
## neighbourhood_cleansedPullman            -2393.21    1715.62  -1.395 0.163035
## neighbourhood_cleansedRogers Park        -1769.62     646.72  -2.736 0.006215
## neighbourhood_cleansedRoseland           -2344.46    1190.90  -1.969 0.048997
## neighbourhood_cleansedSouth Chicago      -3429.09    1047.78  -3.273 0.001066
## neighbourhood_cleansedSouth Deering      -2858.21    5145.27  -0.556 0.578553
## neighbourhood_cleansedSouth Lawndale     -3129.42     894.40  -3.499 0.000467
## neighbourhood_cleansedSouth Shore        -1913.21     667.13  -2.868 0.004134
## neighbourhood_cleansedUptown              -542.04     583.93  -0.928 0.353272
## neighbourhood_cleansedWashington Heights -2447.21    1953.17  -1.253 0.210230
## neighbourhood_cleansedWashington Park    -1445.37     650.45  -2.222 0.026279
## neighbourhood_cleansedWest Englewood      2573.94    1678.95   1.533 0.125264
## neighbourhood_cleansedWest Garfield Park -2157.49    1076.51  -2.004 0.045056
## neighbourhood_cleansedWest Lawn           2440.13    2477.67   0.985 0.324703
## neighbourhood_cleansedWest Pullman       -2783.21    1953.17  -1.425 0.154170
## neighbourhood_cleansedWest Ridge         -2621.21     656.81  -3.991 6.59e-05
## neighbourhood_cleansedWest Town            829.30     575.63   1.441 0.149681
## neighbourhood_cleansedWoodlawn           -2688.14     610.07  -4.406 1.05e-05
##
## (Intercept)                              ***
## neighbourhood_cleansedArmour Square
## neighbourhood_cleansedAshburn
## neighbourhood_cleansedAuburn Gresham
## neighbourhood_cleansedAustin             **
## neighbourhood_cleansedAvalon Park
## neighbourhood_cleansedAvondale
## neighbourhood_cleansedBelmont Cragin
## neighbourhood_cleansedBeverly
## neighbourhood_cleansedBridgeport         **
## neighbourhood_cleansedBrighton Park
```

```
## neighbourhood_cleansedCalumet Heights    *
## neighbourhood_cleansedChatham
## neighbourhood_cleansedChicago Lawn       .
## neighbourhood_cleansedClearing
## neighbourhood_cleansedDouglas            **
## neighbourhood_cleansedDunning            .
## neighbourhood_cleansedEast Garfield Park ***
## neighbourhood_cleansedEast Side
## neighbourhood_cleansedEdgewater          *
## neighbourhood_cleansedEdison Park
## neighbourhood_cleansedEnglewood          ***
## neighbourhood_cleansedForest Glen
## neighbourhood_cleansedGage Park
## neighbourhood_cleansedGrand Boulevard    *
## neighbourhood_cleansedHermosa
## neighbourhood_cleansedHumboldt Park      ***
## neighbourhood_cleansedHyde Park          ***
## neighbourhood_cleansedIrving Park        **
## neighbourhood_cleansedJefferson Park     .
## neighbourhood_cleansedKenwood            *
## neighbourhood_cleansedLincoln Park       *
## neighbourhood_cleansedLincoln Square     **
## neighbourhood_cleansedLogan Square
## neighbourhood_cleansedLoop               ***
## neighbourhood_cleansedLower West Side    ***
## neighbourhood_cleansedMontclare          .
## neighbourhood_cleansedMorgan Park
## neighbourhood_cleansedMount Greenwood
## neighbourhood_cleansedNear North Side    ***
## neighbourhood_cleansedNear West Side
## neighbourhood_cleansedNew City           **
## neighbourhood_cleansedNorth Center
## neighbourhood_cleansedNorth Lawndale     ***
## neighbourhood_cleansedNorth Park
## neighbourhood_cleansedNorwood Park       *
## neighbourhood_cleansedPortage Park       *
## neighbourhood_cleansedPullman
## neighbourhood_cleansedRogers Park        **
## neighbourhood_cleansedRoseland           *
## neighbourhood_cleansedSouth Chicago      **
## neighbourhood_cleansedSouth Deering
## neighbourhood_cleansedSouth Lawndale     ***
## neighbourhood_cleansedSouth Shore        **
## neighbourhood_cleansedUptown
## neighbourhood_cleansedWashington Heights
## neighbourhood_cleansedWashington Park    *
## neighbourhood_cleansedWest Englewood
## neighbourhood_cleansedWest Garfield Park *
## neighbourhood_cleansedWest Lawn
## neighbourhood_cleansedWest Pullman
## neighbourhood_cleansedWest Ridge         ***
## neighbourhood_cleansedWest Town
## neighbourhood_cleansedWoodlawn           ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7231 on 78249 degrees of freedom
## Multiple R-squared:  0.03579,    Adjusted R-squared:  0.03501
## F-statistic:  46.1 on 63 and 78249 DF,  p-value: < 2.2e-16
```

```
mymodel_2 <-lm(airbnb_30_days_price ~ Zip.Code,data = final_df)
summary(mymodel_2)
```

```
##
## Call:
## lm(formula = airbnb_30_days_price ~ Zip.Code, data = final_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -5466  -3095  -1805    414 294504
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6839.40684 4311.83334   1.586    0.113
## Zip.Code      -0.02266    0.07109  -0.319    0.750
##
## Residual standard error: 7364 on 78191 degrees of freedom
##   (120 observations deleted due to missingness)
## Multiple R-squared:  1.3e-06,    Adjusted R-squared:  -1.149e-05
## F-statistic: 0.1016 on 1 and 78191 DF,  p-value: 0.7499
```

##How you addressed this problem statement # 1. Data research and collection – I have used the inside Airbnb website to #gather the data for AibBnb listing data, Affordable rental housing data, and #Average rent Chicago neighborhood data. #Each row of the data represents properties listed on Airbnb, their prices, #neighborhood, zip codes, and average rents in that neighborhood. The data is #focused on 'Chicago' city. #2. Data preparation and cleansing – Identified that there were some missing #data in the datasets. I have removed those records which have NA. I have also #dropped some of the fields that are not used for the analysis.I have merged the #3 datasets into one final dataset for the analysis.

##Analysis #EDA (Exploratory data analysis) #– I did the correlation analysis between variables and noted the strengths and #weaknesses of relationships. #a) I found that zip codes, neighborhood, price, average rent, and property type #are good predictors for the analysis. Once the predictors are decided then I #looked into the R2 , Adjusted R2 statistics, and p-value. #b) Visualize different aspects of the data to gain more knowledge. #c) I then calculated the betas for the predictors in the regression model. #It shows me how the 1 standard deviation change in predictor will impact the #dependent (response) variable. #d) I then calculated confidence intervals which indicate that the estimates of #how the model is likely to be representative of the true population values. #e) I then performed an analysis of variance on all models to compare the #performance of different models. #f) I then calculated standardized residuals, the leverage, cooks' distance, and #covariance rations #g) At last, I checked if the regression model was unbiased.

##Implications to consumer #The implication of the research is that the prices of the housing rental have #direct impact on AirBnB listing in that same neighborhood. The recommendation #from research is that there should be federal rule on how much the housing #prices should increase year by year. #Also AirBnB should consider the neighborhood housing prices when deciding #prices for their listed property.

##Limitations of the analysis #1) The research is limited to Chicago city only. #2) The research needs more sample size for accurate analysis. #3) The research datasets are gathered from only one source. #4) The research is limited based on neighborhood. There are other factors need #to be considered for more

analysis. #5) The research is done using linear regression. There is a scope for #improvement by fitting other ML algorithms.

##Concluding Remarks #The research helped in applying the concepts and knowledge of the statistics #gained in the course. # research project provides hands-on experience in a real-life case study. #The research methods defined in the courses were very helpful. #The visualization technique learned through this case study can be used in #another research too.