EXPLORATORY DATA ANALYSIS

Student Name: Clariza Look

Contents

1. Introduction

This project is about conducting an exploratory data analysis using R of an airbnb dataset in Washington in 2015. It also includes in-depth analysis of data using visualizations and plots to extract insights. The EDA will help the analyst discover information about dataset's structure, determine critical variables, identify anomalies and evaluate assumptions.

The methodology behind this process is to first understand the data by cleaning the data and converting the variables to the correct format. After that, we will extract the variables we wanted to do an analysis, then visualize and interpret results.

This data was made available by Data World Washington 2015 Airbnb Dataset (https://data.world/codefordc/airbnb-washington-d-c-2015-10-03/workspace/file?filename=Listings+-+Detailed.csv).

2. Description of Data

The file contains the scraped data captured by a web program with listing information from the airbnb website. This data may contain unformatted data points and many have duplicate entries. We'd want to clean and format the data prior to performing exploratory analysis, that will help better understand the available data and build some business context.

Business Understanding

Before digging into data understanding and preparation, it is crucial that we first know our end goal or be aware of why this specific dataset is to be explored. There are many possible insights that we want to take from a few business questions to be asked but we will be focusing the questions to be answered during the process of the analysis.

Problem Statements

Price

- How are the listings distributed by location, by price, by type of property?
- Which neighborhoods has the most expensive listings in DC?
- What are the different types of listings in DC? Does their price vary by neighborhood?
- Is there a relationships between price and other variables in the listing?

3. Load Packages

```
library(plyr)
library(dplyr)
library(formattable)
library(tidyverse)
library(ggplot2)
library(kableExtra)
library(leaflet)
library(treemapify)
```

4. Reading The Raw Data

```
airbnb_data <- read.csv('Listings.csv', header = TRUE, sep = ",")

kable(airbnb_data[1:7,1:28]) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"), full_width = F, font_size = 9) %>%
  scroll_box(width = "910px", height = "400px")
```

id	listing_url	scrape_id	last_scraped	name	summary	space	description	experiences_offered	neighborhood_overview	notes
7087327	https://www.airbnb.com/rooms/7087327 (https://www.airbnb.com/rooms/7087327)	2.0151e+13	2015-10-03	Historic DC Condo-Walk to Capitol!	Professional pictures coming soon! Welcome to Hill Flats. Enjoy being across the street from the Library of Congress, short walk to US Capitol, museums and Union Station. Restaurants and bars a few blocks from our peaceful neighborhood.		Professional pictures coming soon! Welcome to Hill Flats. Enjoy being across the street from the Library of Congress, short walk to US Capitol, museums and Union Station. Restaurants and bars a few blocks from our peaceful neighborhood.	none		

5. Data Understanding and Preparation

```
dim(airbnb_data)
```

```
## [1] 3723   92
```

The listings.csv data has 3723 rows/observations (refer as the listings) and 96 columns (refer as attributes or features) per listing.

```
glimpse(airbnb_data)
```

```
## Rows: 3,723
## Columns: 92
## $ id                                <int> 7087327, 975833, 8249488, 8409022,...
## $ listing_url                       <chr> "https://www.airbnb.com/rooms/7087...
## $ scrape_id                         <dbl> 2.0151e+13, 2.0151e+13, 2.0151e+13...
## $ last_scraped                      <chr> "2015-10-03", "2015-10-03", "2015-...
## $ name                              <chr> "Historic DC Condo-Walk to Capitol...
## $ summary                           <chr> "Professional pictures coming soon...
## $ space                             <chr> "", "Beautifully renovated Capitol...
## $ description                       <chr> "Professional pictures coming soon...
## $ experiences_offered               <chr> "none", "none", "none", "none", "n...
## $ neighborhood_overview             <chr> "", "", "", "", "Silver Spring is ...
## $ notes                             <chr> "", "", "", "", "", "Ask permissio...
## $ transit                           <chr> "", "", "", "", "You can walk to t...
## $ thumbnail_url                     <chr> "https://a2.muscache.com/ac/pictur...
## $ medium_url                        <chr> "https://a2.muscache.com/im/pictur...
## $ picture_url                       <chr> "https://a2.muscache.com/ac/pictur...
## $ xl_picture_url                    <chr> "https://a2.muscache.com/ac/pictur...
## $ host_id                           <int> 15830506, 5338703, 1487418, 169702...
## $ host_url                          <chr> "https://www.airbnb.com/users/show...
## $ host_name                         <chr> "Lize & Greg", "Sebastian", "Craig...
## $ host_since                        <chr> "2014-05-21", "2013-03-05", "2011-...
## $ host_location                     <chr> "Washington, District of Columbia,...
## $ host_about                        <chr> "We are two fun, friendly entrepre...
## $ host_response_time                <chr> "within a few hours", "within a da...
## $ host_response_rate                <chr> "92%", "90%", "90%", "100%", "92%"...
## $ host_acceptance_rate              <chr> "91%", "100%", "100%", "N/A", "67%...
## $ host_is_superhost                 <chr> "f", "f", "f", "f", "f", "f", "f",...
## $ host_thumbnail_url                <chr> "https://a1.muscache.com/ac/users/...
## $ host_picture_url                  <chr> "https://a1.muscache.com/ac/users/...
## $ host_neighbourhood                <chr> "Truxton Circle", "Capitol Hill", ...
## $ host_listings_count               <int> 26, 1, 2, 1, 1, 1, 1, 1, 8, 9, 1, ...
## $ host_total_listings_count         <int> 26, 1, 2, 1, 1, 1, 1, 1, 8, 9, 1, ...
## $ host_verifications                <chr> "['email', 'phone', 'facebook', 'g...
## $ host_has_profile_pic              <chr> "t", "t", "t", "t", "t", "t", "t",...
## $ host_identity_verified            <chr> "t", "f", "t", "f", "t", "t", "t",...
## $ street                            <chr> "3rd Street Southeast, Washington,...
## $ neighbourhood                     <chr> "Capitol Hill", "Capitol Hill", "C...
## $ neighbourhood_cleansed            <chr> "Capitol Hill, Lincoln Park", "Cap...
## $ neighbourhood_group_cleansed      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ city                              <chr> "Washington", "Washington", "Hyatt...
## $ state                             <chr> "DC", "DC", "MD", "DC", "MD", "DC"...
## $ zipcode                           <chr> "20003", "20003", "20782", "20024"...
## $ market                            <chr> "D.C.", "D.C.", "D.C.", "D.C.", "D...
## $ smart_location                    <chr> "Washington, DC", "Washington, DC"...
## $ country_code                      <chr> "US", "US", "US", "US", "US", "US"...
## $ country                           <chr> "United States", "United States", ...
## $ latitude                          <dbl> 38.89005, 38.88041, 38.95529, 38.8...
## $ longitude                         <dbl> -77.00281, -76.99048, -76.98601, -...
## $ is_location_exact                 <chr> "t", "t", "t", "f", "t", "f", "t",...
## $ property_type                     <chr> "House", "House", "House", "House"...
## $ room_type                         <chr> "Entire home/apt", "Entire home/ap...
## $ accommodates                      <int> 4, 6, 1, 2, 4, 4, 4, 2, 2, 2, 4, 1...
## $ bathrooms                         <dbl> 1.0, 3.0, 2.0, 1.0, 1.0, 1.0, 2.0,...
## $ bedrooms                          <int> 1, 3, 1, 1, 1, 2, 2, 1, 1, 1, 2, 1...
## $ beds                              <int> 2, 3, 1, 1, 1, 4, 2, 1, 1, 1, 2, 1...
## $ bed_type                          <chr> "Real Bed", "Real Bed", "Real Bed"...
## $ amenities                         <chr> "{TV,\"Wireless Internet\",\"Air C...
## $ square_feet                       <int> NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ price                             <chr> "$160.00", "$350.00", "$50.00", "$...
## $ weekly_price                      <chr> "$1,125.00", "", "$300.00", "", "$...
## $ monthly_price                     <chr> "$5,225.00", "", "$700.00", "$2,65...
## $ security_deposit                  <chr> "$100.00", "", "", "", "$450.00", ...
## $ cleaning_fee                      <chr> "$115.00", "$100.00", "", "", "$15...
## $ guests_included                   <int> 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 4, 1...
## $ extra_people                      <chr> "$0.00", "$0.00", "$0.00", "$0.00"...
## $ minimum_nights                    <int> 1, 2, 2, 1, 7, 1, 3, 1, 2, 2, 4, 3...
## $ maximum_nights                    <int> 1125, 30, 1125, 1125, 1125, 1125, ...
## $ calendar_updated                  <chr> "today", "today", "4 days ago", "n...
## $ has_availability                  <chr> "t", "t", "t", "t", "t", "t", "t",...
## $ availability_30                   <int> 0, 12, 0, 30, 16, 26, 13, 30, 25, ...
## $ availability_60                   <int> 0, 38, 30, 60, 46, 56, 13, 60, 55,...
```

```
## $ availability_90                 <int> 8, 68, 60, 90, 76, 86, 17, 90, 85,...
## $ availability_365                <int> 283, 343, 60, 365, 351, 361, 21, 3...
## $ calendar_last_scraped           <chr> "2015-10-02", "2015-10-02", "2015-...
## $ number_of_reviews               <int> 0, 65, 1, 0, 0, 0, 0, 0, 1, 4, 5, ...
## $ first_review                    <chr> "", "2013-03-22", "2015-09-10", ""...
## $ last_review                     <chr> "", "2015-09-28", "2015-09-10", ""...
## $ review_scores_rating            <int> NA, 94, NA, NA, NA, NA, NA, NA, 10...
## $ review_scores_accuracy          <int> NA, 10, NA, NA, NA, NA, NA, NA, 10...
## $ review_scores_cleanliness       <int> NA, 9, NA, NA, NA, NA, NA, NA, 10,...
## $ review_scores_checkin           <int> NA, 10, NA, NA, NA, NA, NA, NA, 10...
## $ review_scores_communication     <int> NA, 10, NA, NA, NA, NA, NA, NA, 10...
## $ review_scores_location          <int> NA, 9, NA, NA, NA, NA, NA, NA, 10,...
## $ review_scores_value             <int> NA, 9, NA, NA, NA, NA, NA, NA, 10,...
## $ requires_license                <chr> "f", "f", "f", "f", "f", "f", "f",...
## $ license                         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ jurisdiction_names              <chr> "DISTRICT OF COLUMBIA, WASHINGTON"...
## $ instant_bookable                <chr> "f", "f", "f", "f", "f", "f", "f",...
## $ cancellation_policy             <chr> "flexible", "strict", "flexible", ...
## $ require_guest_profile_picture   <chr> "f", "f", "f", "f", "f", "f", "f",...
## $ require_guest_phone_verification <chr> "f", "f", "f", "f", "f", "f", "f",...
## $ calculated_host_listings_count  <int> 18, 1, 1, 1, 1, 1, 1, 1, 2, 1, ...
## $ reviews_per_month               <dbl> NA, 2.11, 1.00, NA, NA, NA, NA, NA...
```

- These columns are of a character datatype (string format) with the dollar symbol '$' . If one of the these attributes will be used in the calculation, they need to be transformed into a numerical data.
    - price
    - weekly_price
    - monthly_price
    - security_deposit
    - cleaning_fee
    - extra_people
- Columns that should be categorical data are also not in the right format such as below:
    - neighbourhood
    - neighbourhood_cleansed
    - city
    - state
    - property_type

If these attributes will be used in the analysis, they need to be transformed into a categorical data.

Checking Missing Values

```
colSums(sapply(airbnb_data, is.na))
```

```
##                             id                    listing_url
##                              0                              0
##                      scrape_id                   last_scraped
##                              0                              0
##                           name                        summary
##                              0                              0
##                          space                    description
##                              0                              0
##             experiences_offered            neighborhood_overview
##                              0                              0
##                          notes                        transit
##                              1                              0
##                  thumbnail_url                     medium_url
##                              0                              0
##                    picture_url                  xl_picture_url
##                              0                              0
##                        host_id                      host_url
##                              0                              0
##                      host_name                    host_since
##                              0                              0
##                  host_location                    host_about
##                              0                              0
##              host_response_time              host_response_rate
##                              0                              0
##             host_acceptance_rate               host_is_superhost
##                              0                              0
##              host_thumbnail_url                host_picture_url
##                              0                              0
##              host_neighbourhood             host_listings_count
##                              0                              0
##        host_total_listings_count              host_verifications
##                              0                              0
##             host_has_profile_pic           host_identity_verified
##                              0                              0
##                         street                   neighbourhood
##                              0                              0
##          neighbourhood_cleansed     neighbourhood_group_cleansed
##                              0                           3723
##                           city                          state
##                              0                              0
##                        zipcode                         market
##                              0                              0
##                 smart_location                   country_code
##                              0                              0
##                        country                       latitude
##                              0                              0
##                      longitude              is_location_exact
##                              0                              0
##                  property_type                      room_type
##                              0                              0
##                   accommodates                      bathrooms
##                              0                             27
##                       bedrooms                           beds
##                             21                             11
##                       bed_type                     amenities
##                              0                              0
##                    square_feet                          price
##                           3641                              0
##                   weekly_price                  monthly_price
##                              0                              0
##                security_deposit                   cleaning_fee
##                              0                              0
##                guests_included                   extra_people
##                              0                              0
##                 minimum_nights                 maximum_nights
##                              0                              0
##                calendar_updated               has_availability
##                              0                              0
##                availability_30                availability_60
##                              0                              0
##                availability_90                availability_365
##                              0                              0
```

```
##            calendar_last_scraped               number_of_reviews
##                                0                               0
##                     first_review                     last_review
##                                0                               0
##             review_scores_rating          review_scores_accuracy
##                              868                             875
##        review_scores_cleanliness           review_scores_checkin
##                              876                             876
##      review_scores_communication          review_scores_location
##                              872                             872
##              review_scores_value                requires_license
##                              872                               0
##                          license              jurisdiction_names
##                             3722                               0
##                 instant_bookable             cancellation_policy
##                                0                               0
##    require_guest_profile_picture require_guest_phone_verification
##                                0                               0
##   calculated_host_listings_count               reviews_per_month
##                                0                             830
```

- Some data have null values. If one of the these attributes will be used in the data exploration, this needs to be addressed.

Remove Noise from Data

- In this section we will filter the unnecessary columns from the dataset and therefore, we only choose relevant columns that we think we will be using in the analysis. Thus we will NOT include columns such as + id + listing_url + scrape_id + last_scraped + thumbnail_url + medium_url + picture_url + xl_picture_url + host_url + host_thumbnail_url + host_picture_url
    - This will also include columns that has a large number of NULL values. + license + square_feet + neighbourhood_group_cleansed

```
#Remove unnecessary columns
new_airb_data <- select(airbnb_data, -id, -listing_url, -scrape_id, -last_scraped, -thumbnail_url,
                        -medium_url,-picture_url,-xl_picture_url,-host_url, -host_thumbnail_url,
                        -host_picture_url, -license, -square_feet, -neighbourhood_group_cleansed)

dim(new_airb_data)
```

```
## [1] 3723   78
```

6. Data Exploration

Choosing the Relevant Columns For Analysis

- Since the dataset is composed of properties in other states, we would like focus our analysis only for properties in DC.

```
unique(new_airb_data[,"state"])
```

```
## [1] "DC"            "MD"            "VA"            "NY"
## [5] "Washington DC"
```

Load data that has properties only in Washington, DC

```
#Filter data
dc_dataframe <- filter(new_airb_data,state == "DC")

## Count rows & columns
dim(dc_dataframe)
```

```
## [1] 3696   78
```

Now we have a clean dataframe that will be ready for the next step which is doing the Price Analysis.

- **Chosen Columns and Their Descriptions**
    - Based from the problem statements in above, we will be using these columns:
        - “price”: The price for a one-night stay.
        - “property_type”: refers to type of property

- “room_type”: refers to room type
- “bedrooms”: refers to number of bedrooms
- “bed_type”: refers to type of bed
- “bathrooms”: refers to number of bathrooms
- “guests_included”: refers to number of guests are included in the price
- “extra_people”: refers to price for every extra person on top of “guests_included”
- “accommodates”: refers to number of people the property can accommodate
- “minimum_nights”: refers to the minimum number of nights guest could stay
- “host_name”: refers to name of host, names can be the same but these are actually different people
- “host_id” refers to to ID of host, hosts can have 1 or more listings
- “number_of_reviews” refers to number of people that gave reviews
- “review_scores_rating” refers to rating from 30% (less positive) to 100% (positive)
- “latitude” of listing
- “longitude” of listing
- “neighbourhood” The neighborhood location of the property
    - For the “neighbourhood” data, some rows are missing values, so we will need more info from another column which is below
        - “neighbourhood_cleansed:” The neighborhood location of the property with more detailed information

```
#Store columns in a new dataframe
airb_df <- dc_dataframe %>% select(price, property_type, room_type, bedrooms, bed_type, bathrooms, guests_included, extra_pe
ople, accommodates, neighbourhood, neighbourhood_cleansed, minimum_nights, host_name, host_id, number_of_reviews, review_sco
res_rating, latitude, longitude)

#check data in a formatted table
head(formattable(airb_df))
```

price	property_type	room_type	bedrooms	bed_type	bathrooms	guests_included	extra_people	accommodates	neighbourhood	nei
$160.00	House	Entire home/apt	1	Real Bed	1	1	$0.00	4	Capitol Hill	(
$350.00	House	Entire home/apt	3	Real Bed	3	1	$0.00	6	Capitol Hill	(
$95.00	House	Private room	1	Real Bed	1	1	$0.00	2		So
$99.00	Boat	Entire home/apt	2	Real Bed	1	2	$25.00	4		So
$100.00	Condominium	Entire home/apt	2	Real Bed	2	2	$40.00	4	Takoma	
$149.00	Apartment	Entire home/apt	1	Real Bed	1	1	$99.00	1	Woodley Park	C

```
#Check data's datatype
glimpse(airb_df)
```

```
## Rows: 3,696
## Columns: 18
## $ price                  <chr> "$160.00", "$350.00", "$95.00", "$99.00", "$...
## $ property_type          <chr> "House", "House", "House", "Boat", "Condomin...
## $ room_type              <chr> "Entire home/apt", "Entire home/apt", "Priva...
## $ bedrooms               <int> 1, 3, 1, 2, 2, 1, 1, 1, 2, 1, 1, 1, 1, 1,...
## $ bed_type               <chr> "Real Bed", "Real Bed", "Real Bed", "Real Be...
## $ bathrooms              <dbl> 1.0, 3.0, 1.0, 1.0, 2.0, 1.0, 1.0, 1.5, 1.0,...
## $ guests_included        <int> 1, 1, 1, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 1, 1,...
## $ extra_people           <chr> "$0.00", "$0.00", "$0.00", "$25.00", "$40.00...
## $ accommodates           <int> 4, 6, 2, 4, 4, 1, 4, 4, 3, 4, 2, 2, 4, 2, 2,...
## $ neighbourhood          <chr> "Capitol Hill", "Capitol Hill", "", "", "Tak...
## $ neighbourhood_cleansed <chr> "Capitol Hill, Lincoln Park", "Capitol Hill,...
## $ minimum_nights         <int> 1, 2, 1, 1, 3, 3, 2, 1, 5, 2, 2, 1, 1, 1, 2,...
## $ host_name              <chr> "Lize & Greg", "Sebastian", "Christopher", "...
## $ host_id                <int> 15830506, 5338703, 16970249, 951119, 4628, 9...
## $ number_of_reviews      <int> 0, 65, 0, 0, 0, 46, 84, 25, 4, 2, 83, 19, 94...
## $ review_scores_rating   <int> NA, 94, NA, NA, NA, 100, 99, 91, 100, 100, 9...
## $ latitude               <dbl> 38.89005, 38.88041, 38.87213, 38.86249, 38.9...
## $ longitude              <dbl> -77.00281, -76.99048, -77.01964, -77.01506, ...
```

```
#Check NA or missing values
colSums(sapply(airb_df, is.na))
```

```
##                  price          property_type              room_type
##                      0                      0                      0
##               bedrooms              bed_type              bathrooms
##                     21                      0                     25
##        guests_included           extra_people           accommodates
##                      0                      0                      0
##          neighbourhood neighbourhood_cleansed         minimum_nights
##                      0                      0                      0
##              host_name                host_id      number_of_reviews
##                      0                      0                      0
##   review_scores_rating               latitude              longitude
##                    854                      0                      0
```

```
#Check empty cells
colSums(airb_df == "")
```

```
##                  price          property_type              room_type
##                      0                      1                      0
##               bedrooms              bed_type              bathrooms
##                     NA                      0                     NA
##        guests_included           extra_people           accommodates
##                      0                      0                      0
##          neighbourhood neighbourhood_cleansed         minimum_nights
##                    343                      0                      0
##              host_name                host_id      number_of_reviews
##                      0                      0                      0
##   review_scores_rating               latitude              longitude
##                     NA                      0                      0
```

Insights From The Raw Data To Be Analyzed

- Looking at the data, we can see that
  - It has 3,696 rows and 15 columns
  - “price” & “extra_people” are in a string form (e.g “$99.00”). They should be converted into float so we can use as a price format.
  - categorical variables that needs to convert
    - “property_type”, “room_type”, “bedrooms”, “bed_type”, bathrooms“,”neighbourhood“,”neighbourhood_cleansed“,”host_name“,”host_id“,”review_scores_rating"
- **NA values** :
  - “bedrooms” has 21 NA values, “bathrooms” has 25
  - “bathrooms” has a 0 value and it’s impossible to have 0 bathroom in a property
    - We convert “Bedrooms” Na Values to 0 (assuming it’s a studio-type property)
    - We convert “bathrooms” Na Values to 1 (assuming every property has at least 1 bathroom)
      - We convert “bathrooms” with 0 value to 1 (assuming every property has at least 1 bathroom)
  - “review_scores_rating” has 854, these can be replaced with 0 as it refers to 0 review

- **Empty Cells** (different from NA Values but can impact the analysis):
  - “property_type” has 1 empty cell, we can place it with “Other” property_type
  - “neighbourhood” has 342 empty cells therefore we need extract details from “neighbourhood_cleansed” to fill in the empty cells

Data Cleaning

```
# Remove '$' sign in String from 'price' & 'extra_people'
# Replace new data in the "temp_price$price" column
temp_price <- gsub("\\$","", airb_df$price)
temp_extra_people <- gsub("\\$","", airb_df$extra_people)

# Because there are rows in the cell that has "," comma (e.g. 2,500.00)
# It automatically becomes NA when we convert them to float
# So to solve that problem, we have to remove "," in the columns
temp_price <- gsub("\\,","", temp_price)
temp_extra_people <- gsub("\\,","", temp_extra_people)

# Then convert number to float
float_price <- as.double(temp_price)
float_extra_people <- as.double(temp_extra_people)

# Check datatype again to make sure there are no NA values
summary(float_price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    10.0    85.0   115.0   149.5   165.0  2822.0
```

```
summary(float_extra_people)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   11.33   20.00  300.00
```

```
#Now put back the cleaned data to "p_neighborhood_df$price"
airb_df$price <- float_price
airb_df$extra_people <- float_extra_people
```

A. Dealing with Empty cells in “neighbourhood” and “property_type” column

```
## "neighbourhood" has 342 empty cells therefore we need extract details
## from  "neighbourhood_cleansed" to fill in the empty cells
## Extract string after first comma and store it in new column "new_neighbourhood"
airb_df$new_neighbourhood <- sub(",.*", "", airb_df$neighbourhood_cleansed)

# Replace that empty "property_type" as "other"
airb_df$property_type[airb_df$property_type==''] <- "Other"

# Remove "neighbourhood","neighbourhood_cleansed" columns since we will not use them in
# the analysis as we created the "new_neighborhood" for analysis
cleaned_airb_df <- subset(airb_df, select = -c(neighbourhood,neighbourhood_cleansed))

head(formattable(cleaned_airb_df))
```

price	property_type	room_type	bedrooms	bed_type	bathrooms	guests_included	extra_people	accommodates	minimum_nights	host
160	House	Entire home/apt	1	Real Bed	1	1	0	4	1	Lize
350	House	Entire home/apt	3	Real Bed	3	1	0	6	2	Se
95	House	Private room	1	Real Bed	1	1	0	2	1	Chri
99	Boat	Entire home/apt	2	Real Bed	1	2	25	4	1	
100	Condominium	Entire home/apt	2	Real Bed	2	2	40	4	3	M

price	property_type	room_type	bedrooms	bed_type	bathrooms	guests_included	extra_people	accommodates	minimum_nights	host
149	Apartment	Entire home/apt	1	Real Bed	1	1	99	1	3	

```
str(cleaned_airb_df)
```

```
## 'data.frame':    3696 obs. of  17 variables:
##  $ price              : num  160 350 95 99 100 149 150 175 239 65 ...
##  $ property_type      : chr  "House" "House" "House" "Boat" ...
##  $ room_type          : chr  "Entire home/apt" "Entire home/apt" "Private room" "Entire home/apt" ...
##  $ bedrooms           : int  1 3 1 2 2 1 1 1 2 1 ...
##  $ bed_type           : chr  "Real Bed" "Real Bed" "Real Bed" "Real Bed" ...
##  $ bathrooms          : num  1 3 1 1 2 1 1 1.5 1 1 ...
##  $ guests_included    : int  1 1 1 2 2 1 2 2 1 2 ...
##  $ extra_people       : num  0 0 0 25 40 99 20 25 0 0 ...
##  $ accommodates       : int  4 6 2 4 4 1 4 4 3 4 ...
##  $ minimum_nights     : int  1 2 1 1 3 3 2 1 5 2 ...
##  $ host_name          : chr  "Lize & Greg" "Sebastian" "Christopher" "Todd" ...
##  $ host_id            : int  15830506 5338703 16970249 951119 4628 966914 2070536 12725500 9540128 22207701 ...
##  $ number_of_reviews  : int  0 65 0 0 0 46 84 25 4 2 ...
##  $ review_scores_rating: int  NA 94 NA NA NA 100 99 91 100 100 ...
##  $ latitude           : num  38.9 38.9 38.9 38.9 39 ...
##  $ longitude          : num  -77 -77 -77 -77 -77 ...
##  $ new_neighbourhood  : chr  "Capitol Hill" "Capitol Hill" "Southwest Employment Area" "Southwest Employment Area" ...
```

B. Dealing with the NA

```
colSums(sapply(cleaned_airb_df, is.na))
```

```
##               price       property_type           room_type
##                   0                   0                   0
##            bedrooms            bed_type           bathrooms
##                  21                   0                  25
##     guests_included        extra_people        accommodates
##                   0                   0                   0
##      minimum_nights           host_name             host_id
##                   0                   0                   0
##   number_of_reviews review_scores_rating            latitude
##                   0                 854                   0
##           longitude    new_neighbourhood
##                   0                   0
```

```
################################################################################
## We convert "bedrooms" Na Values to 0 (assuming it's a studio-type property)
## We convert "bathrooms" Na Values to 1 (assuming every property has at least 1 bathroom
## We convert "review_scores_rating" Na Values with 0 as it refers to 0 review
## We convert "bathrooms" with 0 value to 1 (assuming every property has at least 1 bathroom)
################################################################################

cleaned_airb_df$bedrooms[is.na(cleaned_airb_df$bedrooms)] <- 0
cleaned_airb_df$bathrooms[is.na(cleaned_airb_df$bathrooms)] <- 1
cleaned_airb_df$review_scores_rating[is.na(cleaned_airb_df$review_scores_rating)] <- 0
cleaned_airb_df$bathrooms[cleaned_airb_df$bathrooms == 0] <- 1

## Check if NA values are gone
colSums(sapply(cleaned_airb_df, is.na))
```

```
##               price        property_type            room_type
##                   0                    0                    0
##            bedrooms             bed_type            bathrooms
##                   0                    0                    0
##     guests_included         extra_people         accommodates
##                   0                    0                    0
##      minimum_nights            host_name              host_id
##                   0                    0                    0
##    number_of_reviews review_scores_rating             latitude
##                   0                    0                    0
##           longitude    new_neighbourhood
##                   0                    0
```

B. Converting to categorical data

```
# Convert  to categorical data
temp <- cleaned_airb_df %>% mutate_at(vars(room_type, bedrooms, bed_type, bathrooms, host_name, host_id, review_scores_ratin
g, new_neighbourhood, property_type, bed_type),list(factor))
str(temp)
```

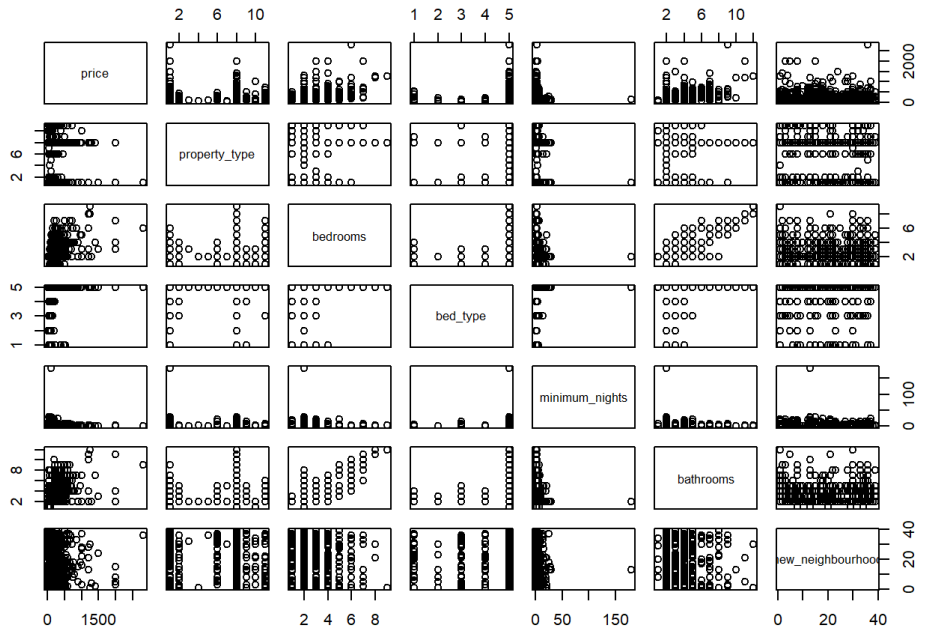
```
## 'data.frame':    3696 obs. of  17 variables:
##  $ price               : num  160 350 95 99 100 149 150 175 239 65 ...
##  $ property_type       : Factor w/ 11 levels "Apartment","Bed & Breakfast",..: 8 8 8 3 6 1 1 1 8 1 ...
##  $ room_type           : Factor w/ 3 levels "Entire home/apt",..: 1 1 2 1 1 1 1 1 1 2 ...
##  $ bedrooms            : Factor w/ 9 levels "0","1","2","3",..: 2 4 2 3 3 2 2 2 3 2 ...
##  $ bed_type            : Factor w/ 5 levels "Airbed","Couch",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ bathrooms           : Factor w/ 12 levels "0.5","1","1.5",..: 2 6 2 2 4 2 2 3 2 2 ...
##  $ guests_included     : int  1 1 1 2 2 1 2 2 1 2 ...
##  $ extra_people        : num  0 0 0 25 40 99 20 25 0 0 ...
##  $ accommodates        : int  4 6 2 4 4 1 4 4 3 4 ...
##  $ minimum_nights      : int  1 2 1 1 3 3 2 1 5 2 ...
##  $ host_name           : Factor w/ 1537 levels "'G' (Gurinder)",..: 886 1280 297 1436 1013 110 715 986 972 1455 ...
##  $ host_id             : Factor w/ 2708 levels "1585","2798",..: 1570 818 1632 232 8 237 391 1399 1165 1877 ...
##  $ number_of_reviews   : int  0 65 0 0 0 46 84 25 4 2 ...
##  $ review_scores_rating: Factor w/ 42 levels "0","30","40",..: 1 36 1 1 1 42 41 33 42 42 ...
##  $ latitude            : num  38.9 38.9 38.9 38.9 39 ...
##  $ longitude           : num  -77 -77 -77 -77 -77 ...
##  $ new_neighbourhood   : Factor w/ 39 levels "Brightwood Park",..: 3 3 32 32 34 6 30 30 30 30 ...
```

```
cleaned_airb_df <- temp
```

Price Analysis

A. Checking Price Relationships with other variables

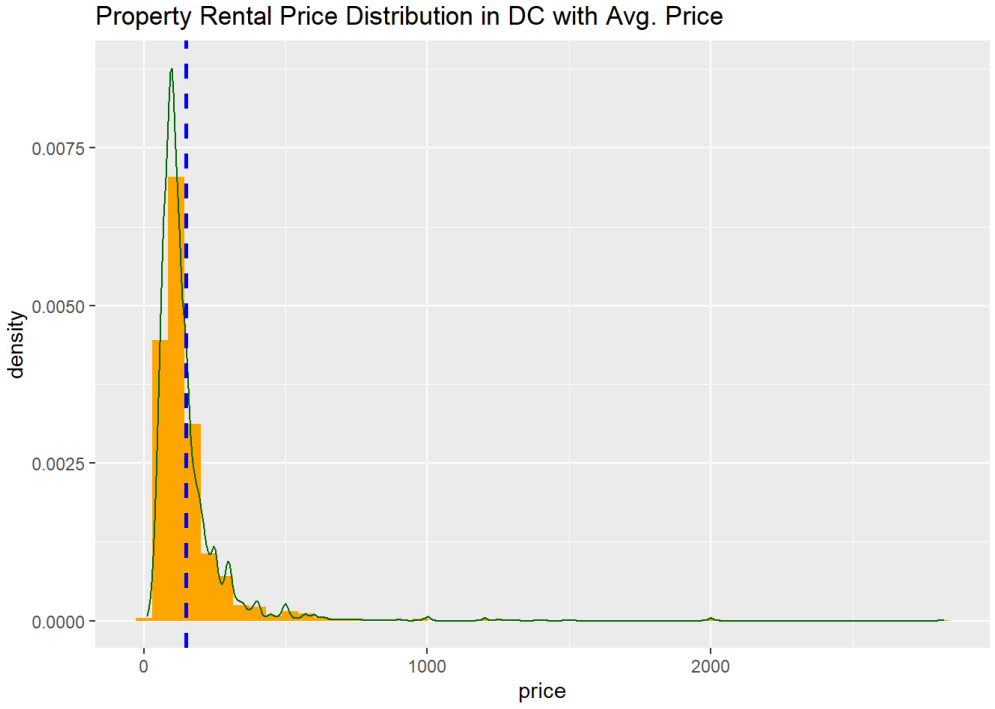
```
pairs(~price+property_type+bedrooms+bed_type+minimum_nights+bathrooms+new_neighbourhood, data=cleaned_airb_df)
```



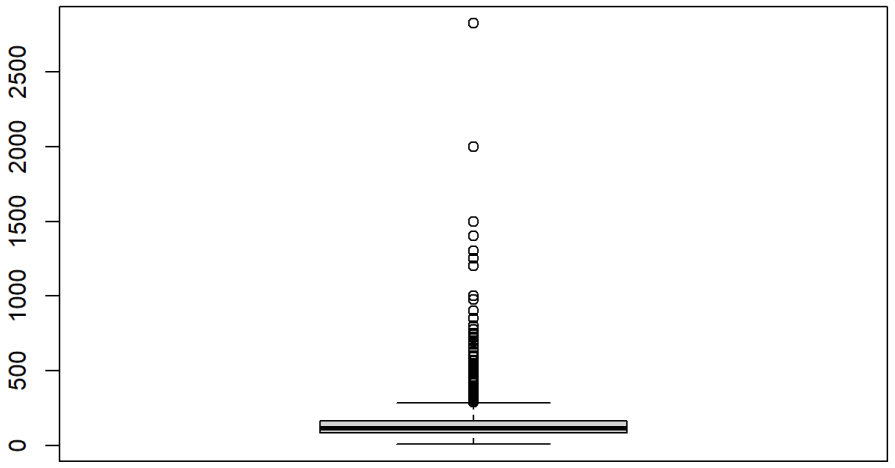
- **Interpretation**
  - There seems to have no clear patterns in the scatter plots that is interesting to do a linear correlation except for bedrooms and bathrooms variables.

B. PRICE FREQUENCY

```
## Histogram of Price Frequency
densplot <- ggplot(data = cleaned_airb_df, mapping = aes(x = price))+
  geom_histogram(aes(y=..density..), bins=50, fill = "orange")+
  geom_density(color='dark green') +
  ggtitle("Property Rental Price Distribution in DC with Avg. Price")

   densplot + geom_vline(aes(xintercept=mean(price)),
           color="blue", linetype="dashed", size=1)
```



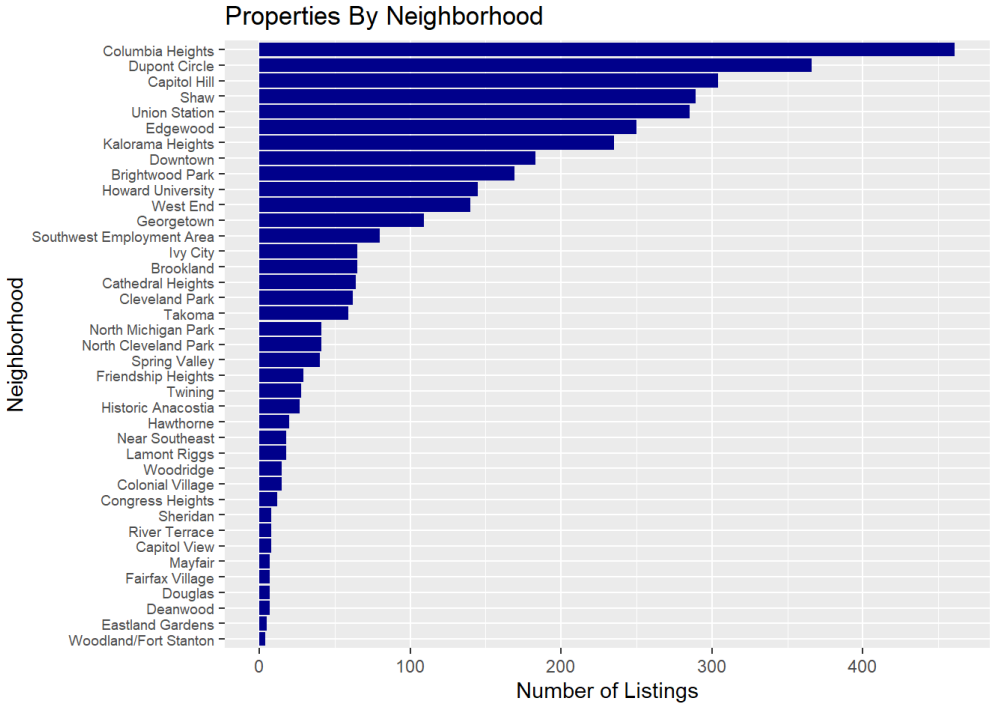

```
##Checking outliers in property price
boxplot(cleaned_airb_df$price)
```



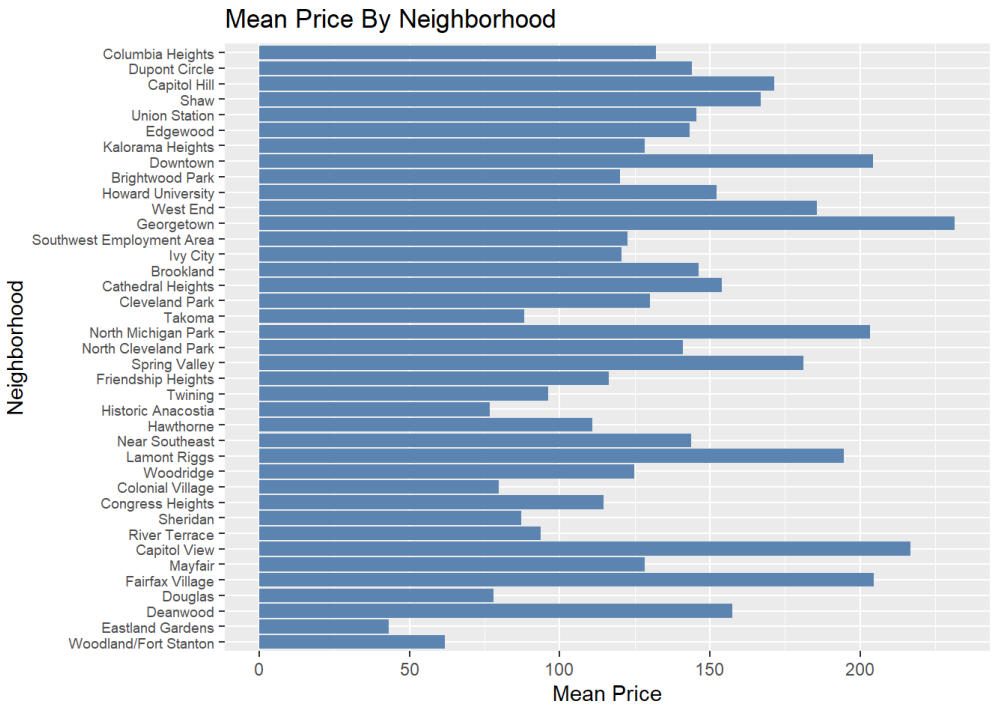
- **Interpretation**
  - The histogram depicts that the price is right skewed which means that the average property rental prices are greater than the median price in DC.
  - Boxplot indicates that there are many outliers in prices that range from $400 to$2500. This could indicate that that these properties have house features which are making it more expensive. It can be more bedrooms,more amenities, prime location, or can accommodate more people

B. PRICE BY NEIGHBORHOOD

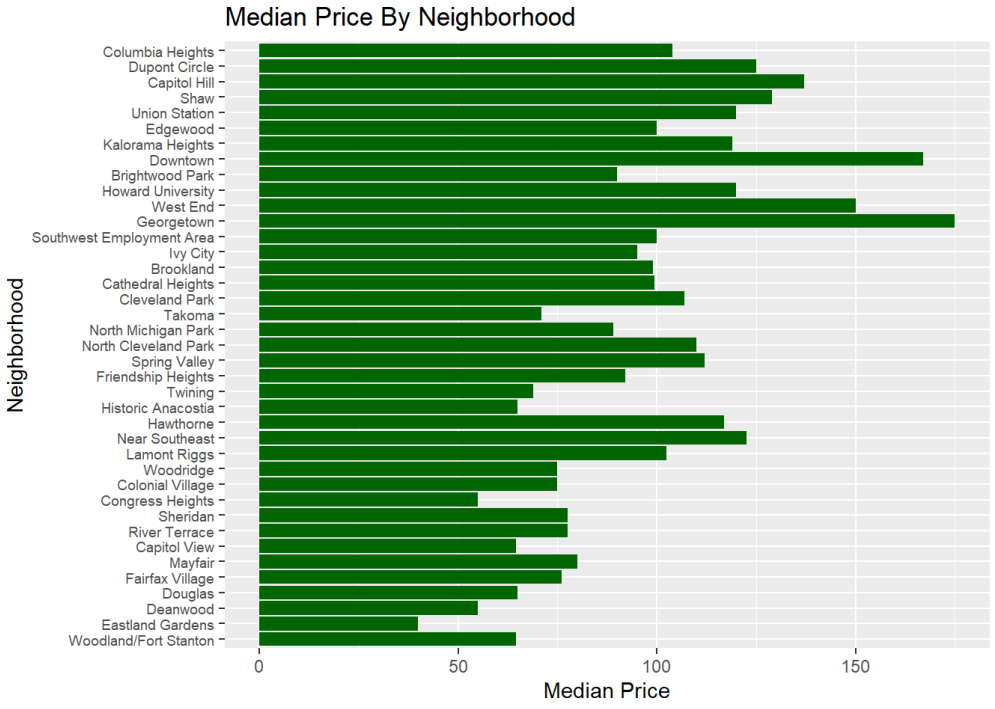
```
## Group price by Location and take average
p2 <- cleaned_airb_df %>%
        group_by(new_neighbourhood) %>%
        summarise(p_mean = mean(price),
                  p_median = median(price),
                  p_count= n()
                  )


ct <- ggplot(p2, mapping=aes(x=reorder(new_neighbourhood,p_count), y=p_count))+
  geom_col(fill="dark blue") +
  ggtitle("Properties By Neighborhood") +
  xlab("Neighborhood") +
  ylab("Number of Listings") +
  coord_flip() +
  theme(axis.text.y=element_text(size=rel(0.8)))

mn <- ggplot(p2, mapping=aes(x=reorder(new_neighbourhood,p_count), y=p_mean))+
  geom_col(fill="#5B84B1FF") +
  ggtitle("Mean Price By Neighborhood") +
  xlab("Neighborhood") +
  ylab("Mean Price") +
  coord_flip() +
  theme(axis.text.y=element_text(size=rel(0.8)))


md <-ggplot(p2, mapping=aes(x=reorder(new_neighbourhood,p_count), y=p_median))+
  geom_col(fill="dark green") +
  ggtitle("Median Price By Neighborhood") +
  xlab("Neighborhood") +
  ylab("Median Price") +
  coord_flip() +
  theme(axis.text.y=element_text(size=rel(0.8)))




ct
```




```
mn
```



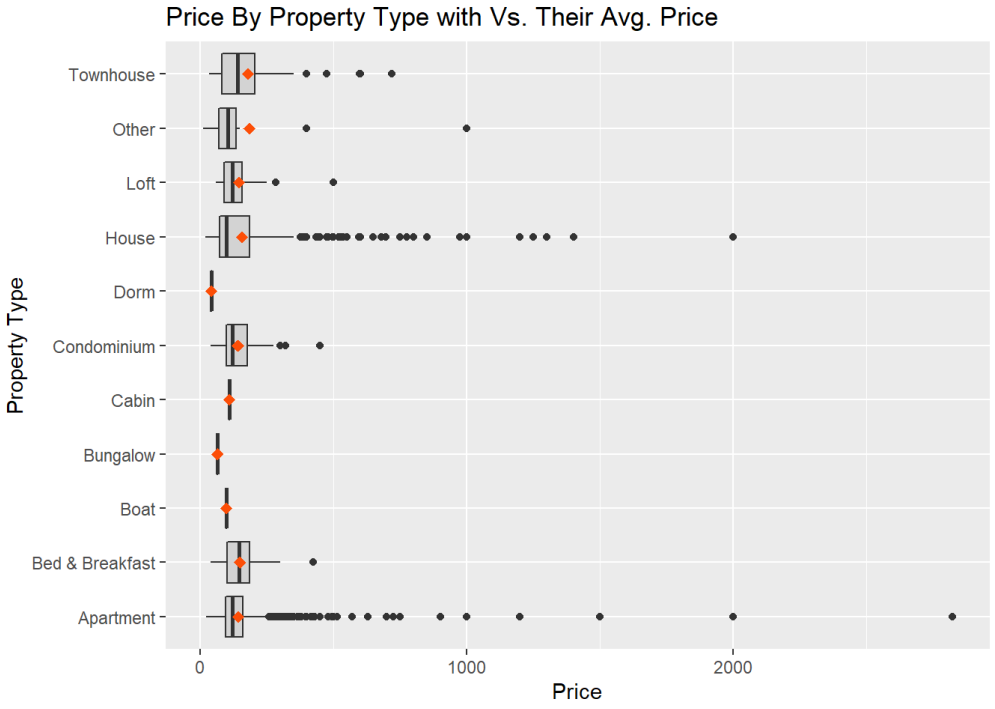

```
md
```



- **Interpretation**
  - “Columbia Heights” has the most number of listings however the prices of the listings varies which affects the mean and median (price in Downtown has higher mean and median despite number of listings in the area are low)

C. PRICE BY PROPERTY TYPE

```
##Checking outliers in property price with mean points
prop_type_outliers <- ggplot(cleaned_airb_df, aes(x = price, y = property_type))

prop_type_outliers + geom_boxplot(notch = FALSE, fill = "lightgray") +
  ggtitle("Price By Property Type with Vs. Their Avg. Price") +
  xlab("Price") +
  ylab("Property Type") +
  stat_summary(fun = mean, geom = "point",
               shape = 18, size = 2.5, color = "#FC4E07")
```

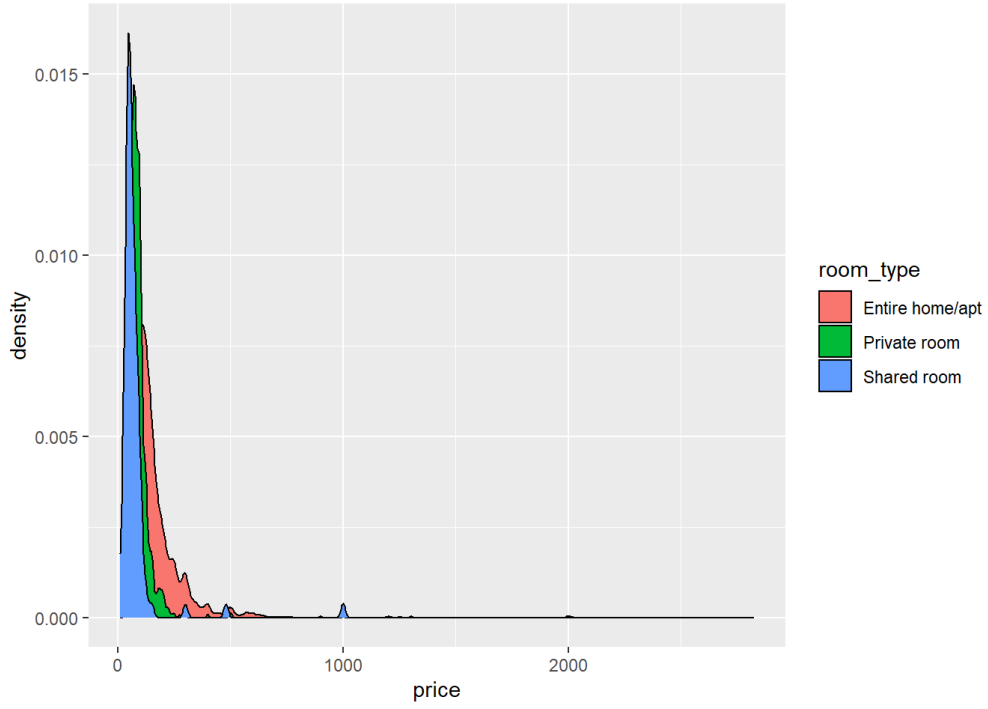


Interpretation

- We can see that “house” and “apartment” have many outliers than other types of property
    - The average price varies by property type

D. PRICE BY ROOM TYPE

```
ggplot(cleaned_airb_df, aes(x=price, fill=room_type)) +
  geom_density()
```

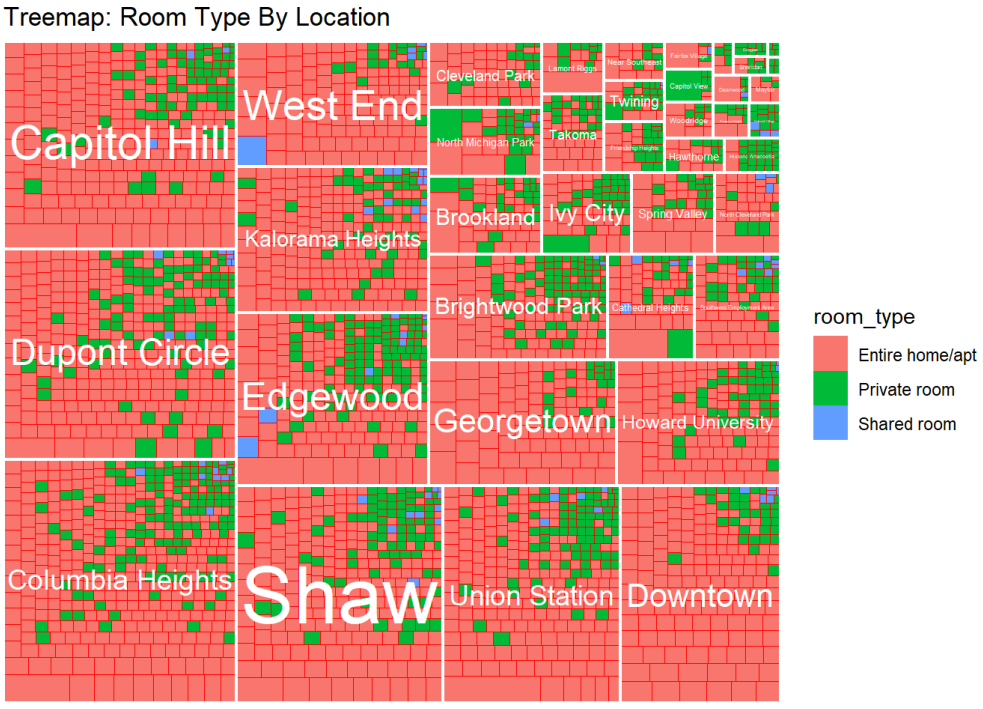


Interpretation

- According to the data, shared room has the most prices than any other room type

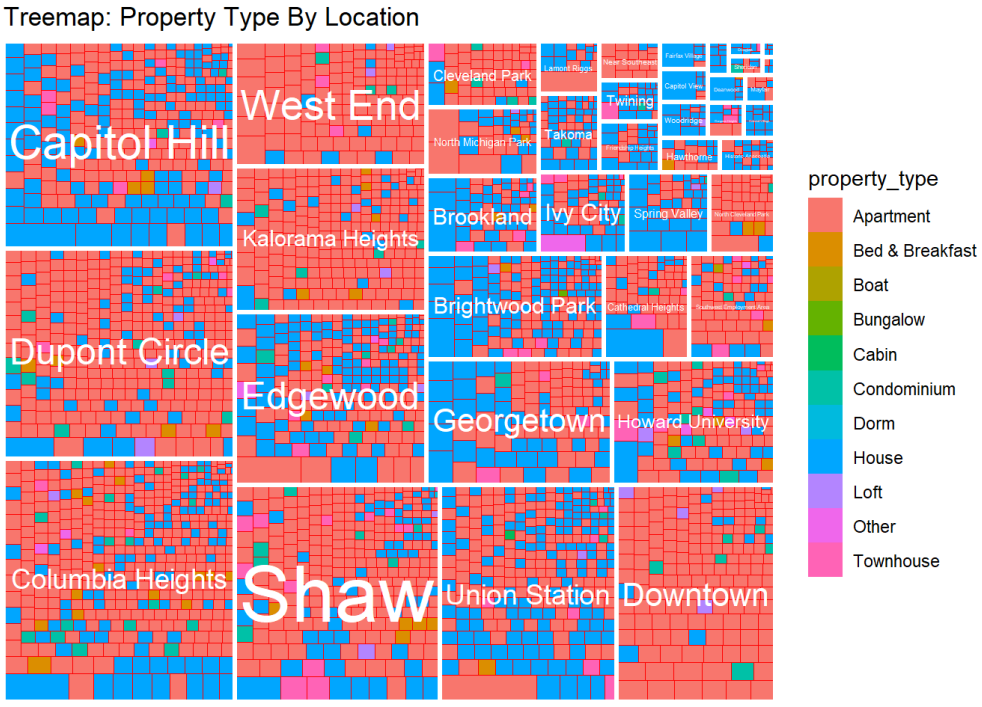
E. Room TypeBy Neighbourhood Treemap

```
tree_viz <- ggplot(cleaned_airb_df, aes(area = price, fill = room_type, subgroup = new_neighbourhood)) +
  geom_treemap(color = "red") +
  geom_treemap_subgroup_border(color = "white", size = 2) +
  geom_treemap_subgroup_text(place = "centre", grow = T, alpha = 1, colour =
                               "white", min.size = 1) +
  labs(title="Treemap: Room Type By Location")
tree_viz
```



E.1. Property Type By Neighbourhood Treemap

```
tree_viz1 <- ggplot(cleaned_airb_df, aes(area = price, fill = property_type, subgroup = new_neighbourhood)) +
  geom_treemap(color = "red") +
  geom_treemap_subgroup_border(color = "white", size = 3) +
  geom_treemap_subgroup_text(place = "centre", grow = T, alpha = 1, colour =
                               "white", min.size = 1) +
  labs(title="Treemap: Property Type By Location")
tree_viz1
```

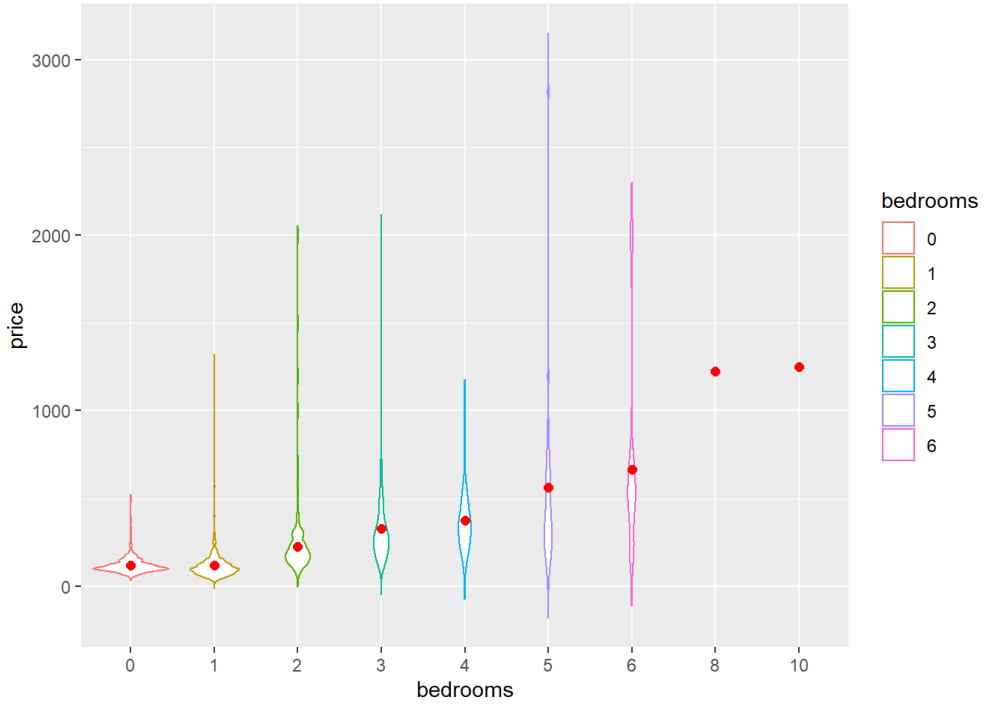


Interpretation

- The treemap depicts the rental prices among all room types and property types by location
  - This is to give an overview of prices by room_type and property types per location
  - Capitol Hill and Union Station seems to have high rental price for “Entire home/apt” in “E. Room TypeBy Neighbourhood Treemap” while it also have high price in “house” property type in “E.1. Property Type By Neighbourhood Treemap”
  - In “E.1. Property Type By Neighbourhood Treemap” West End“,”Downtown" and “Kalorama Heights” has the most apartments compare to other property types in their respective areas.

F. Price By Number of Bedrooms

```
violin_bdrm <- ggplot(cleaned_airb_df, aes(x=price, y=bedrooms, color=bedrooms)) +
  geom_violin(trim=FALSE) +
   coord_flip() +
   stat_summary(fun=mean, geom="point", size=2, color="red")

violin_bdrm
```

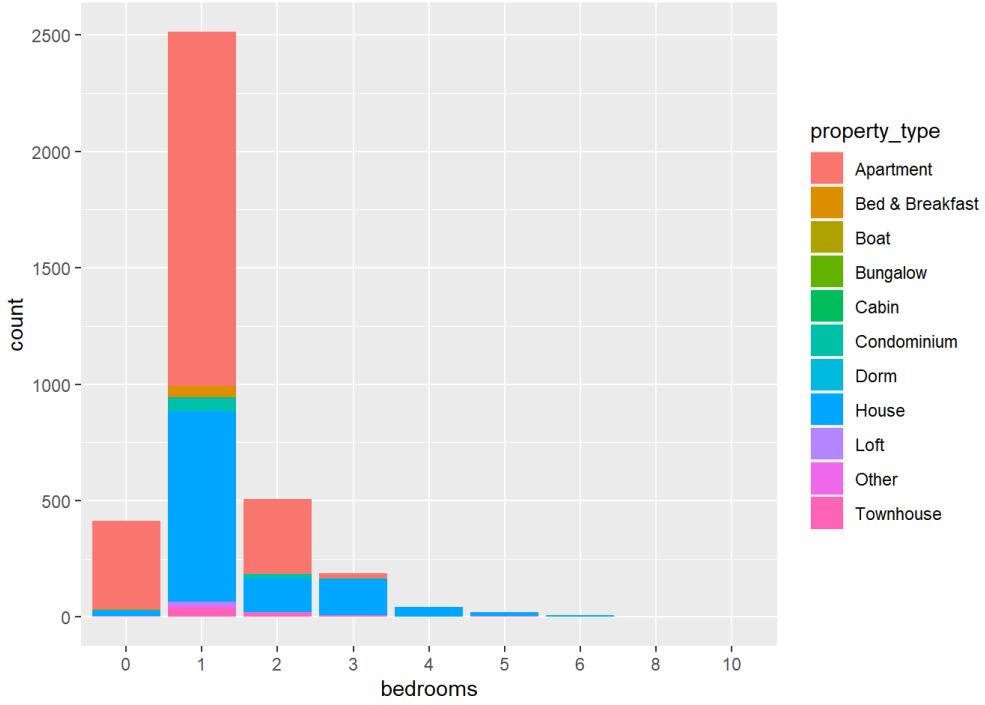


Interpretation

- The violin plot represents the distribution of price per number of bedrooms related to their outlier prices
  - It shows there are many properties with 0 bedrooms (assuming it is a studio type apartment) and rental prices are just below $100.
  - It shows that the 5 bedrooms has a wide range of min to max prices

F. Number of Bedrooms by Property Type

```
ggplot(cleaned_airb_df, aes(x=bedrooms, fill=property_type)) +
  geom_bar()
```

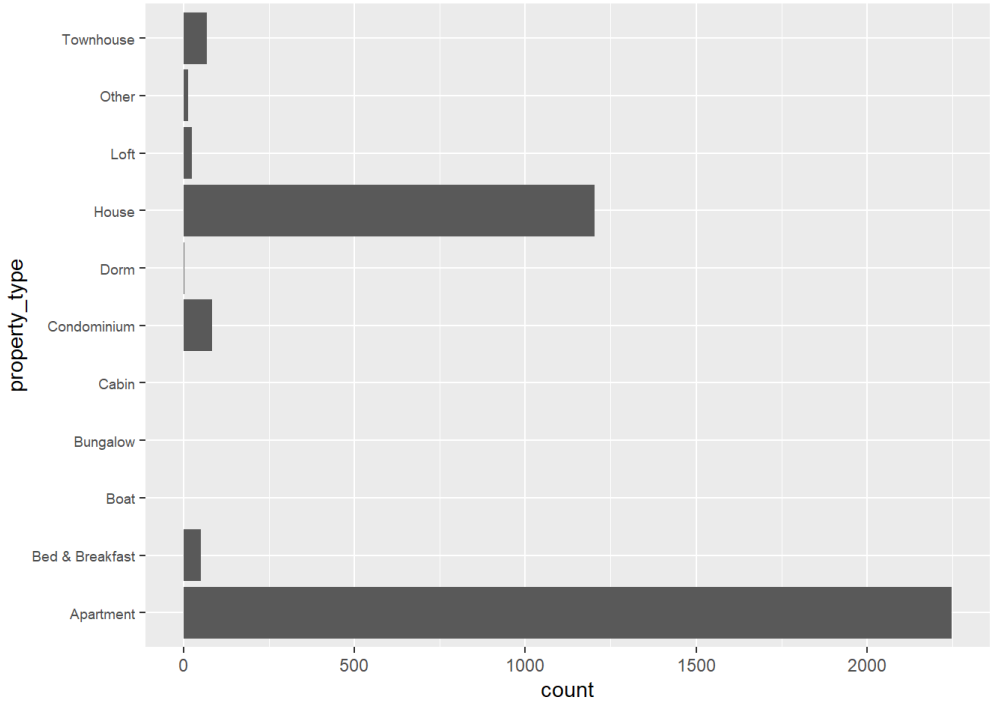


Interpretation

- The stack bar plot represents the number of bedrooms by property
  - The data shows that most of the “Apartments” listed are having 1 bedroom
  - It also shows that all “boats” listings have 1 bedroom

H. Property Type by BedType

```
ggplot(cleaned_airb_df, aes(x=property_type)) +
  geom_bar() +
   coord_flip() +
  theme(axis.text.y=element_text(size=rel(0.8)))
```



Interpretation

- Many of the property types are using “read beds” than any other bedtype
    - The data shows that the least number of bed type being used is couch

I. Number Accommodates by Property Type

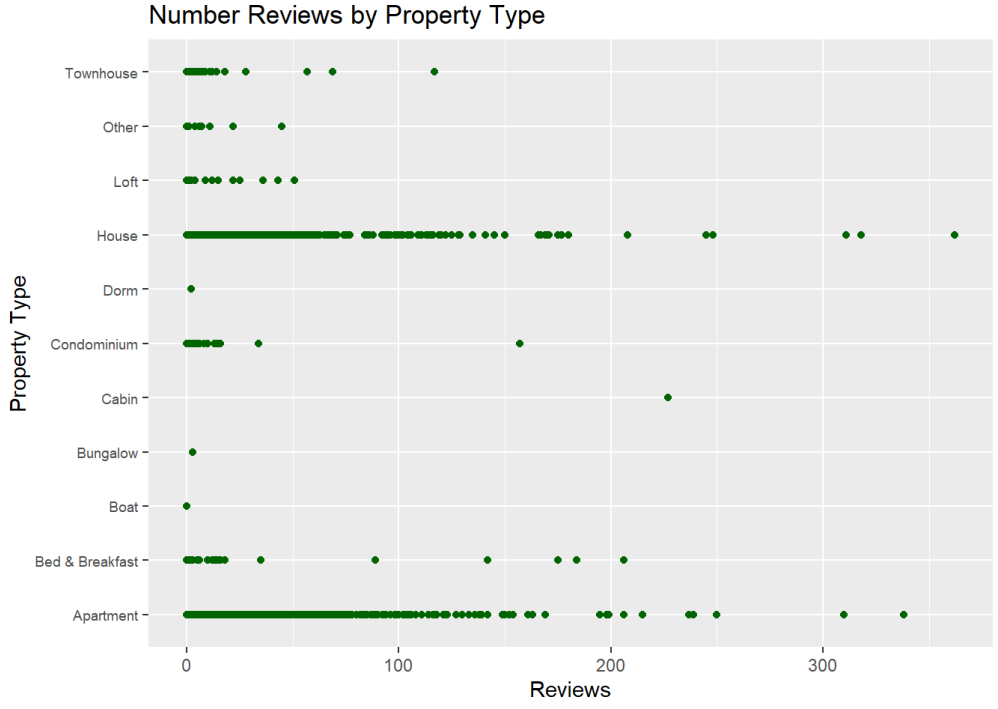
```
pss <- ggplot(cleaned_airb_df, aes(property_type, accommodates))
pss + geom_point(colour="blue") +
   coord_flip() +
   ylab("Accommodates") +
   xlab("Property Type") +
   ggtitle("Number Accommodates by Property Type") +
  theme(axis.text.y=element_text(size=rel(0.8)))
```

Interpretation

- House has the most flexible number of accommodations among all the property types
    - Dorm seems to accept up to 6 people in one stay while boat can take up to 4 people

I. Number of Reviews by Property Type

```
pss <- ggplot(cleaned_airb_df, aes(property_type, number_of_reviews))
pss + geom_point(colour="dark green") +
   coord_flip() +
   ylab("Reviews") +
   xlab("Property Type") +
   ggtitle("Number Reviews by Property Type") +
  theme(axis.text.y=element_text(size=rel(0.8)))
```

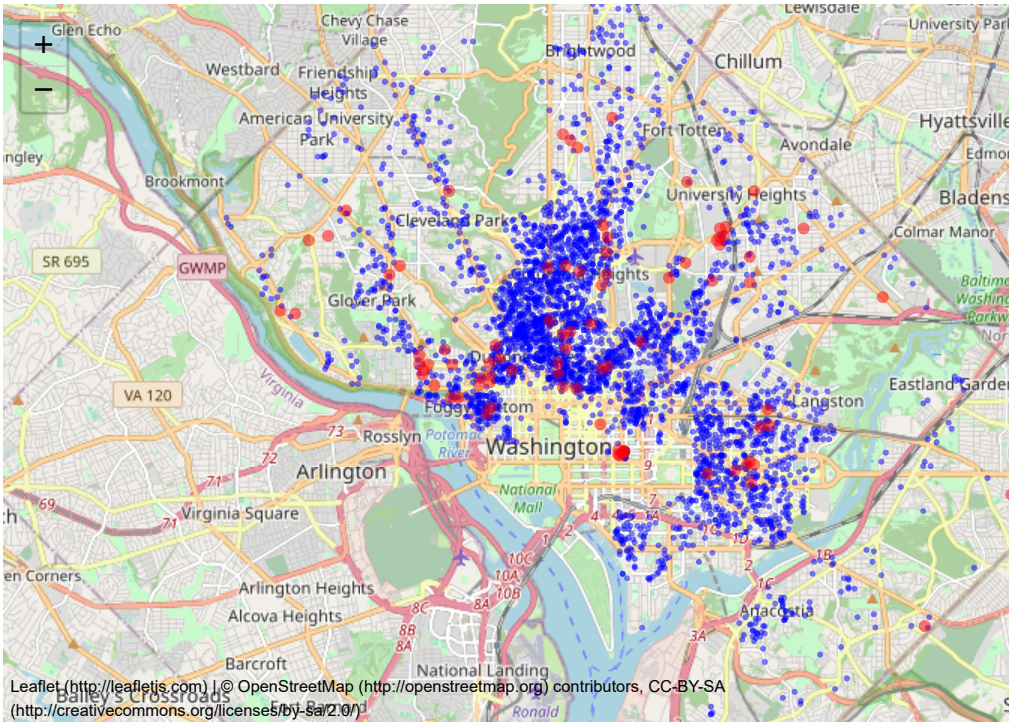


Interpretation

- Reviews indicate the number of people who have stayed and left a review
    - House and Apartment has the most number of reviews which could possibly mean that during this time of the data was captured, House and Apartment property types are the most stayed property type.

J. Listing Map

```
expensive_listings <- cleaned_airb_df %>%
  filter(cleaned_airb_df$price >= 500)

inexpensive_listings <- cleaned_airb_df %>%
  filter(cleaned_airb_df$price < 500)

listing_map <- leaflet() %>%
   addTiles() %>%
   setView(lng = -77.046331, lat = 38.909702 , zoom = 12) %>%
   addCircleMarkers( lng = inexpensive_listings$longitude,
                     lat = inexpensive_listings$latitude,
                     radius = 2,
                     stroke = FALSE,
                     color = "blue",
                     fillOpacity = 0.5,
                     group = "Inexpensive Properties"
                     )  %>%
   addCircleMarkers( lng = expensive_listings$longitude,
                     lat = expensive_listings$latitude,
                     radius = 4,
                     stroke = FALSE,
                     color = "red",
                     fillOpacity = 0.5,
                     group = "Expensive Properties"
                     )


listing_map
```



Interpretation

- The listings map above represents the high price rental properties (price above $500) vs. low price (price below $500) properties
    - Looking at the map alone does not clearly specify patterns of high price based on just the location having features such as “prime spot”, or “near the park”, “near the airport” , “near the river” , etc.

7. Summary & Conclusions

The data analysis done above a great stat to give insights about pricing by different variables.

Problem Statements

- How are the listings distributed by location, by price, by type of property?
    - They are distributed around Washington DC and the price varies by property type, number of bedrooms and perhaps some other variables that is not covered by our dataset
- Which neighborhoods has the most expensive listings in DC?
    - In the “Listing Map” map section, it shows the distribution of high priced listing compare to low priced listings by map level. However in the “B. PRICE BY NEIGHBORHOOD” section, it states that based on median price, Georgetown neighbourhood has the highest price.
- What are the different types of listings in DC? Does their price vary by neighborhood?
    - These includes Townhouse, Loft, House, Dorm, Condominium, Cabin, Bungalow, Boat, Bed & Breakfast, Apartment
    - In the “E. Room Type Treemap” section, it shows the prices by room type per location which gives us a rough idea of prices by neighborhood
- Is there a relationships between price and other variables in the listing?
    - This part needs further analysis and perhaps would need more variables in the dataset or perhaps analyse the “description” column and extract details about property features.

However we can also do further analysis on:

- Price with date and time to determine demand/supply of house price rentals in the location
- Revenue with all variables (similar to pricing analysis) to determine which location is best to invest an airbnb property

8. References

- http://www.sthda.com/english/articles/32-r-graphics-essentials/132-plot-grouped-data-box-plot-bar-plot-and-more/ (http://www.sthda.com/english/articles/32-r-graphics-essentials/132-plot-grouped-data-box-plot-bar-plot-and-more/)
- https://www.kaggle.com/chrisbow/e-commerce-eda-and-segmentation-with-r (https://www.kaggle.com/chrisbow/e-commerce-eda-and-segmentation-with-r)
- https://rkabacoff.github.io/datavis/Bivariate.html (https://rkabacoff.github.io/datavis/Bivariate.html)
- https://www.kaggle.com/notaapple/detailed-exploratory-data-analysis-using-r (https://www.kaggle.com/notaapple/detailed-exploratory-data-analysis-using-r)
- https://analyticsindiamag.com/tutorial-get-started-with-exploratory-data-analysis-and-data-preprocessing/ (https://analyticsindiamag.com/tutorial-get-started-with-exploratory-data-analysis-and-data-preprocessing/)
- https://learn.r-journalism.com/en/mapping/leaflet_maps/leaflet/ (https://learn.r-journalism.com/en/mapping/leaflet_maps/leaflet/)