
New York City
KNOW BEFORE YOU GO

Airbnb in the Big Apple:

Understanding Room Types, Prices & Location Trends in New York City

INFO-B518 : Applied Statistics In Biomedical Informatics
Project presentation

GROUP 3
Megha Moncy, Mohith Surya Kiran Kasula, Nisha Thakur, Pallavi Singh, Pallavi Vaswani
May 04, 2023

List of Contents

New York City Airbnb Open Data

- Dimensions : 48895, 16
- Selected columns and their datatypes
- Summary Statistics

```
{r}
No of rows and columns
dim(data)
```

```
[1] 48895    16
```

RStudio: Notebook Output

```
'data.frame':   48895 obs. of  8 variables:
 $ neighbourhood_group: chr  "Brooklyn" "Manhattan" "Manhattan" "Brooklyn" ...
 $ neighbourhood      : chr  "Kensington" "Midtown" "Harlem" "Clinton Hill" ...
 $ latitude           : num  40.6 40.8 40.8 40.7 40.8 ...
 $ longitude          : num  -74 -74 -73.9 -74 -73.9 ...
 $ room_type          : chr  "Private room" "Entire home/apt" "Private room" "Entire home/apt" ...
 $ price              : int  149 225 150 89 80 200 60 79 79 150 ...
 $ minimum_nights     : int  1 1 3 1 10 3 45 2 2 1 ...
 $ availability_365   : int  365 355 365 194 0 129 0 220 0 188 ...
```

```
{r}
# Generating a summary of all data attributes with the summary() function
summary(data)
```

RStudio: Notebook Output

```
 neighbourhood_group neighbourhood        latitude       longitude       room_type             price          minimum_nights    availability_365
 Length:48895        Length:48895       Min.   :40.50   Min.   :-74.24   Length:48895       Min.   :    0.0   Min.   :   1.00   Min.   :  0.0
 Class :character    Class :character   1st Qu.:40.69   1st Qu.:-73.98   Class :character   1st Qu.:   69.0   1st Qu.:   1.00   1st Qu.:  0.0
 Mode  :character    Mode  :character   Median :40.72   Median :-73.96   Mode  :character   Median :  106.0   Median :   3.00   Median : 45.0
                                        Mean   :40.73   Mean   :-73.95                      Mean   :  152.7   Mean   :   7.03   Mean   :112.8
                                        3rd Qu.:40.76   3rd Qu.:-73.94                      3rd Qu.:  175.0   3rd Qu.:   5.00   3rd Qu.:227.0
                                        Max.   :40.91   Max.   :-73.71                      Max.   :10000.0   Max.   :1250.00   Max.   :365.0
```

```
{r}
data <- select(data, neighbourhood_group, neighbourhood, latitude, longitude, room_type, price,
minimum_nights,availability_365)
# Viewing the first DataFrame records
head(data, 6)
```

The average price per night is $152.70, with a range from $0 to $10,000.
The average minimum stay is 7.03 nights, with some listings requiring only 1 night and others up to 1,250 nights.
Listings have an avg availability of 112.8 days/year, ranging from completely unavailable (0 days) to available year-round (365 days).

Hypothesis - 1

Null Hypothesis:

The average price of reservations made available by Airbnb is the **same** across all room types.

Alternate Hypothesis:

The average price of reservations made available by Airbnb **differs significantly** among all room types.

Descriptive Analysis

- Unique Values of room_type

```r
```{r}
c(unique(data["room_type"]))
```
```

```
$room_type
[1] "Private room"    "Entire home/apt" "Shared room"
```

- Minimum – Maximum values of price

```r
```{r}
library(glue)

glue("Price Minimum: {min(data$price)} | Price Maximum: {max(data$price)}")
```
```

```
Price Minimum: 0 | Price Maximum: 10000
```

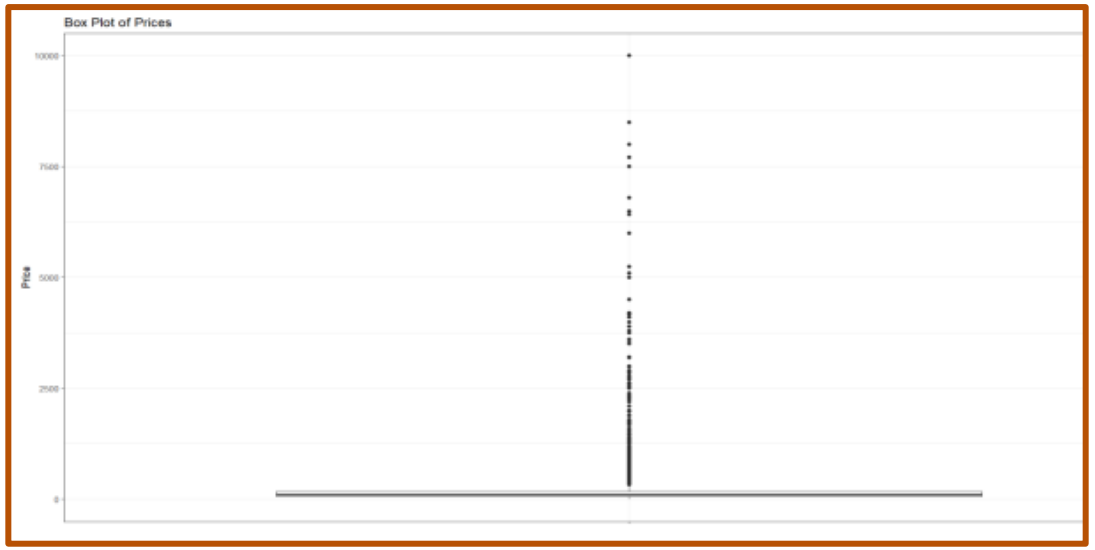
Data Cleaning

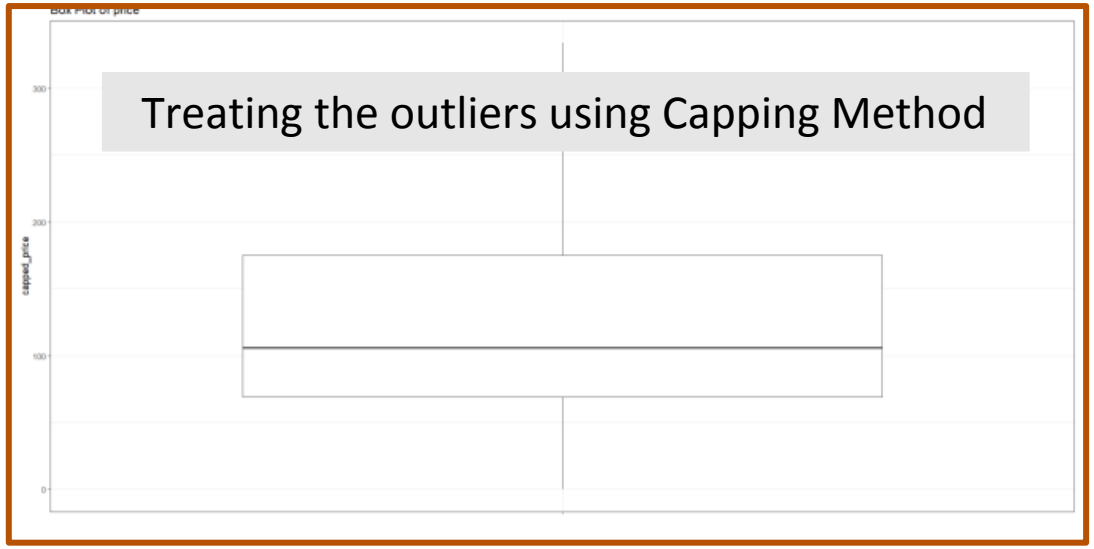
```r
#calculating upper and lower bound

price_IQR <- IQR(data$price, na.rm = TRUE)
price_Q1 <- quantile(data$price, 0.25, na.rm = TRUE)
price_Q3 <- quantile(data$price, 0.75, na.rm = TRUE)

lower_bound_price <- price_Q1 - 1.5 * price_IQR
upper_bound_price <- price_Q3 + 1.5*price_IQR
```

```r
#capping the price variable
data_capped <- data
data_capped$capped_price <- ifelse(data$price < lower_bound_price, lower_bound_price,
                                   ifelse(data$price > upper_bound_price, upper_bound_price,data$price))
```

Checking for outliers in 'price' column
From the box plot, we can observe the presence of outliers in the 'price' column of the New York City Airbnb dataset.

```r
library(ggplot2)
ggplot(data, aes(x = "", y = price)) +
  geom_boxplot() +
  labs(title = "Box Plot of Prices", x = "", y = "Price") +
theme_bw()
```





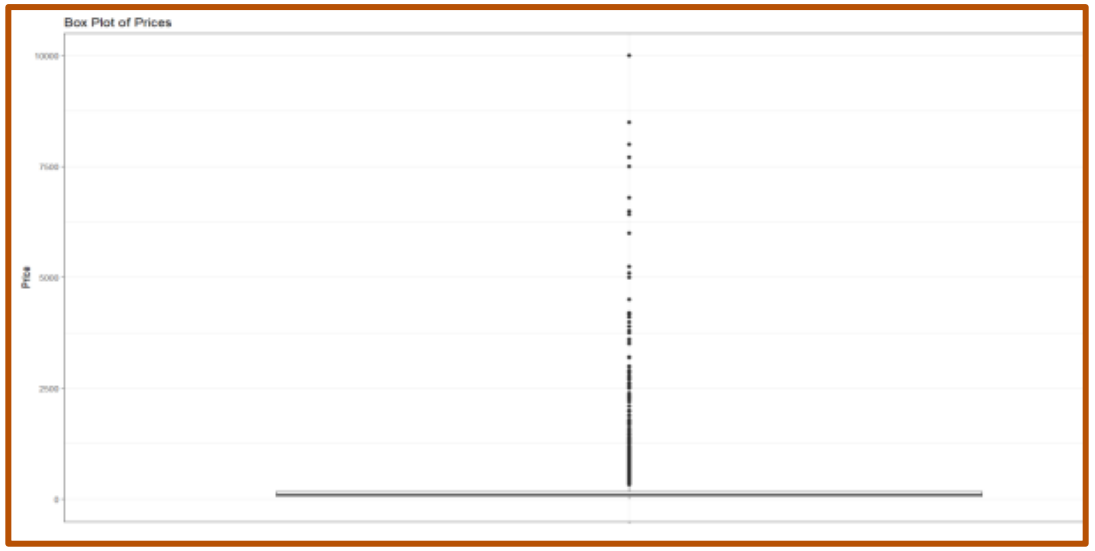
Visualizing 'price' column after outlier removal

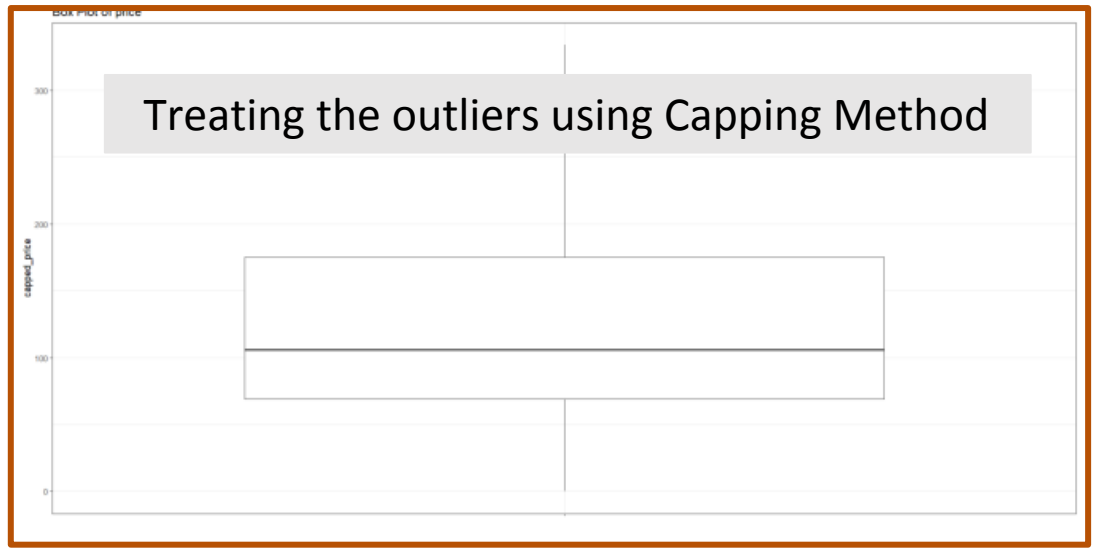
Data Cleaning

```
334
335
336 ```{r}
337 library(glue)
338
339 glue("Price Minimum: {min(data_capped$capped_price)} | Price Maximum: {max(data_capped$capped_price)}")
340
341 ```
```

```
Price Minimum: 0 | Price Maximum: 334
```

So, after treating the outliers, The minimum price is 0, which means there are listings in the dataset with a price of $0 per night. The maximum price is $334 per night, which is the highest price among all listings in the dataset.

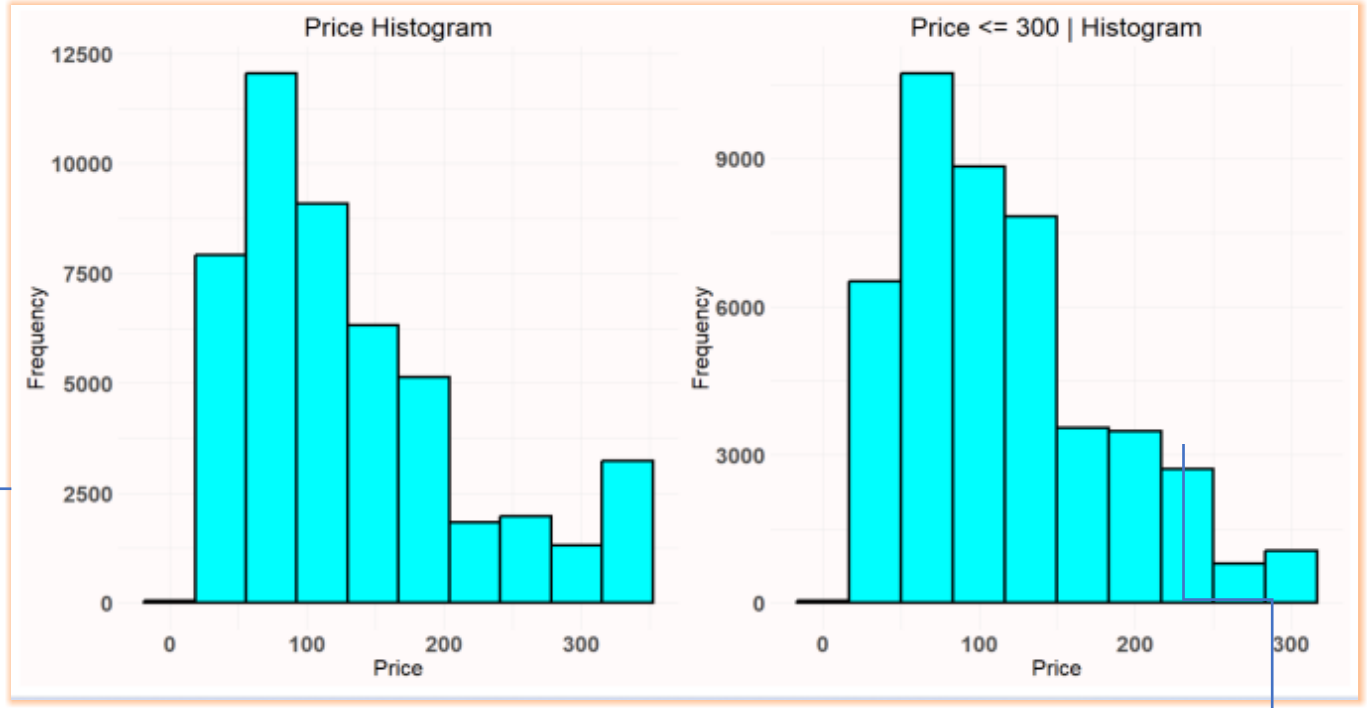




Visualizing 'price' column after outlier removal

Data Visualization : Price

```r
library(ggplot2)
library(cowplot)
tema <- theme(plot.background = element_rect(fill = "#FFFAFA", color = "#FFFAFA"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 19, face = "bold"),
              axis.text.y = element_text(size = 19, face = "bold"),
              axis.title.x = element_text(size = 19),
              axis.title.y = element_text(size = 19),
              legend.position = "none")

options(repr.plot.width=14, repr.plot.height=6)
a <- ggplot(data = data_capped, mapping = aes(x = capped_price)) +
         geom_histogram(fill = "cyan", bins = 10, size = 1.3, color = "black") +
         theme_minimal() +
         ylab("Frequency") +
         xlab("Price") +
         ggtitle("Price Histogram") +
         tema
df <- data.frame(price = data_capped["capped_price"][data_capped["capped_price"] <= 300])
b <- ggplot(data = df, mapping = aes(x = price)) +
         geom_histogram(fill = "cyan", bins = 10, size = 1.3, color = "black") +
         theme_minimal() +
         ylab("Frequency") +
         xlab("Price") +
         ggtitle("Price <= 300 | Histogram") +
         tema


plot_grid(a, b, ncol=2, nrow=1)
```



X-axis : price values : Y-axis : frequency of price

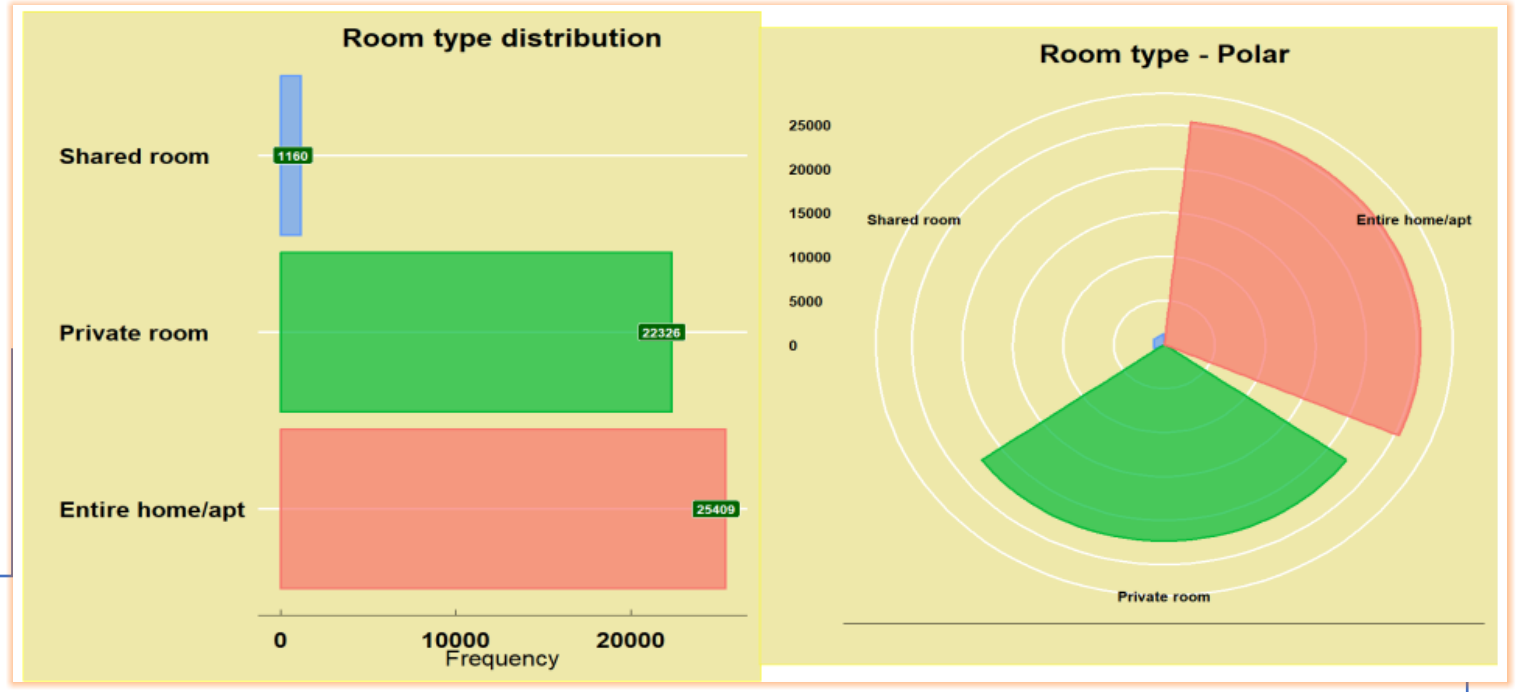
- The entire distribution of "price" values, divided into 10 equally spaced bins
- The majority of the values are concentrated in the left part of the histogram, indicating that most listings have a lower price, and the dataset is right skewed.

- The distribution of the "price" variable for a subset of the data, where the price is less than or equal to 300, respectively.
- The dataset is right skewed indicating that most of the price listings ranges between 0 and 300.

Data Visualization : Room Type

```r
library(ggthemes)
library(ggplot2)
library(cowplot)
tema <- theme(plot.background = element_rect(fill = "#EEEBAA", color = "yellow"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 19, face = "bold"),
              axis.text.y = element_text(size = 19, face = "bold"),
              axis.title.x = element_text(size = 19),
              axis.title.y = element_text(size = 19),
              legend.position = "none")

tema1 <- theme(plot.background = element_rect(fill = "#EEEBAA",color = "yellow"),
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 12, face = "bold"),
              axis.text.y = element_text(size = 12, face = "bold"),
              axis.title.x = element_text(size = 12),
              axis.title.y = element_text(size = 12),
              legend.position = "none")
options(repr.plot.width=15, repr.plot.height=6)
a <- ggplot(data = freq_type, mapping = aes(x = Frequency, y = row.names(freq_type))) +
        geom_bar(stat = "identity", mapping = aes(fill = row.names(freq_type), color = row.names(freq_type)), alpha = .7,
linewidth = 1.1) +
        geom_label(mapping = aes(label=Frequency), fill = "#D6E8D8",linewidth = 6, color = "white", fontface = "bold",
hjust=.7) +
        ylab("") +
        ggtitle("Room type distribution") +
        theme_economist() +
        tema

b <- ggplot(data = freq_type, aes(x = row.names(freq_type), y = Frequency)) +
        geom_bar(stat = "identity", mapping = aes(fill = row.names(freq_type), color = row.names(freq_type)), alpha = .7,
size = 1.1) +
        theme_economist() +
        xlab("") +
        ylab("") +
        ggtitle("Room type - Polar") +
        tema1
```

```r
plot_grid(a, b + coord_polar(), ncol=2, nrow=1)
```



X-axis : Room type: Y-axis : frequency of price

- The bar chart shows that the majority of listings in the dataset are for entire homes/apartments (25409), followed by private rooms (22326) and shared rooms (60).

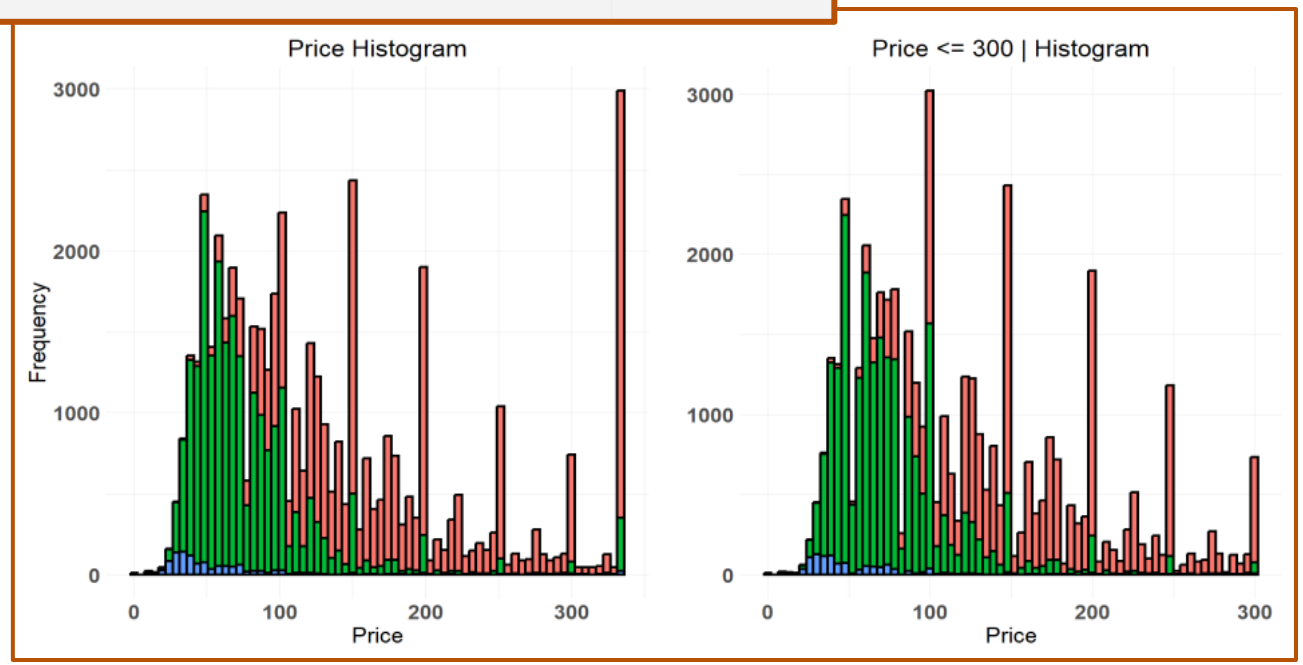
- **The polar bar chart** displays the same information, but in a radial format. The bars are arranged in a circle, with the longest bar representing entire homes/apartments, followed by private rooms and shared rooms

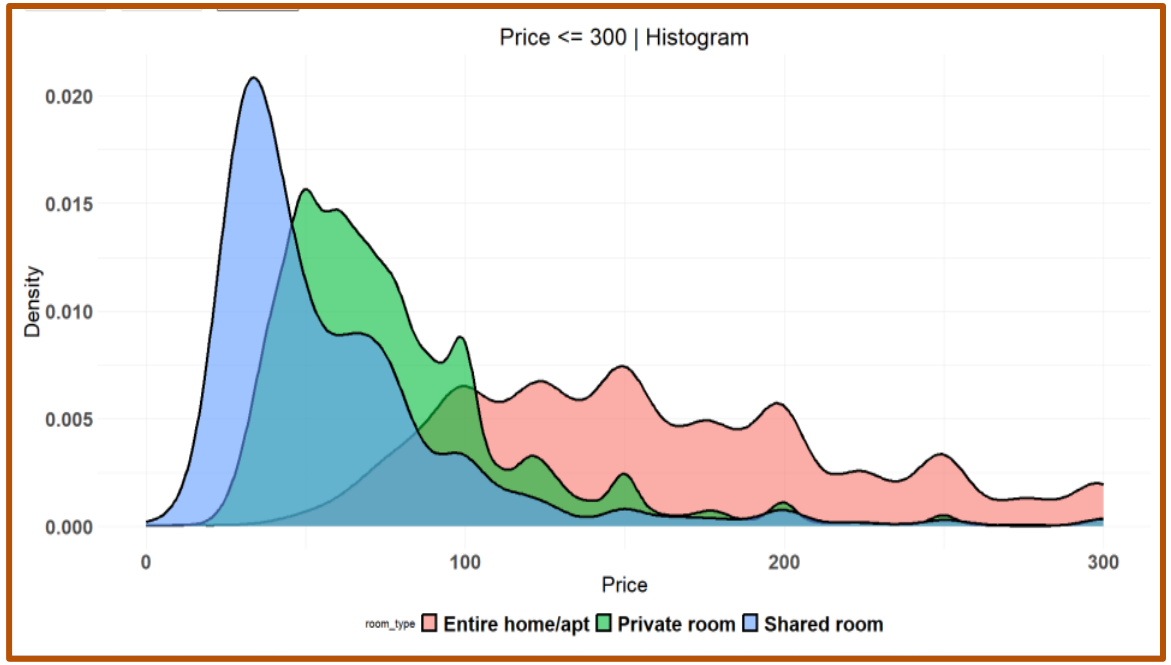
Normality Testing (Histogram): Price Vs Room_Type

```r
tema <- theme(
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 19, face = "bold"),
              axis.text.y = element_text(size = 19, face = "bold"),
              axis.title.x = element_text(size = 21),
              axis.title.y = element_text(size = 21),
              legend.position = "none")
tema1 <- theme(
              plot.title = element_text(size = 23, hjust = .5),
              axis.text.x = element_text(size = 19, face = "bold"),
              axis.text.y = element_text(size = 19, face = "bold"),
              axis.title.x = element_text(size = 21),
              axis.title.y = element_text(size = 21))
options(repr.plot.width=14, repr.plot.height=6)
a <- ggplot(data = data_capped, mapping = aes(x = capped_price)) +
        geom_histogram(mapping = aes(fill = room_type), bins = 70, size = 1.3, color = 'black') +
        theme_minimal() +
        ylab("Frequency") +
        xlab("Price") +
        ggtitle("Price Histogram") +
        tema
df <- data.frame(capped_price = data_capped["capped_price"][data_capped["capped_price"] <= 300], room_type =
data_capped["room_type"][data_capped["capped_price"] <= 300])
b <- ggplot(data = df, mapping = aes(x = capped_price)) +
        geom_histogram(mapping = aes(fill = room_type), bins = 70, size = 1.3, color = 'black') +
        theme_minimal() +
        ylab("") +
        xlab("Price") +
        ggtitle("Price <= 300 | Histogram") +
        tema
df <- data.frame(capped_price = data_capped["capped_price"][data_capped["capped_price"] <= 300], room_type =
data_capped["room_type"][data_capped["capped_price"] <= 300])
c <- ggplot(data = df, mapping = aes(x = capped_price, fill = room_type)) +
        geom_density(mapping = aes(fill = room_type), bins = 70, size = 1.3, color = 'black', alpha = .6, size = 1.5) +
        theme_minimal() +
        ylab("Density") +
        xlab("Price") +
        ggtitle("Price <= 300 | Histogram") +
        tema1 +
        theme(legend.position="bottom", legend.text = element_text(colour='black', size=20,
                                   face="bold"))
plot_grid(a, b, ncol=2, nrow=1)
c
```

- The plots indicate that the distribution of price values in the airbnb dataset is right-skewed, with a few listings having very high prices.
- "Entire home/apt" listings tend to have higher price values than "Private room" or "Shared room" listings.

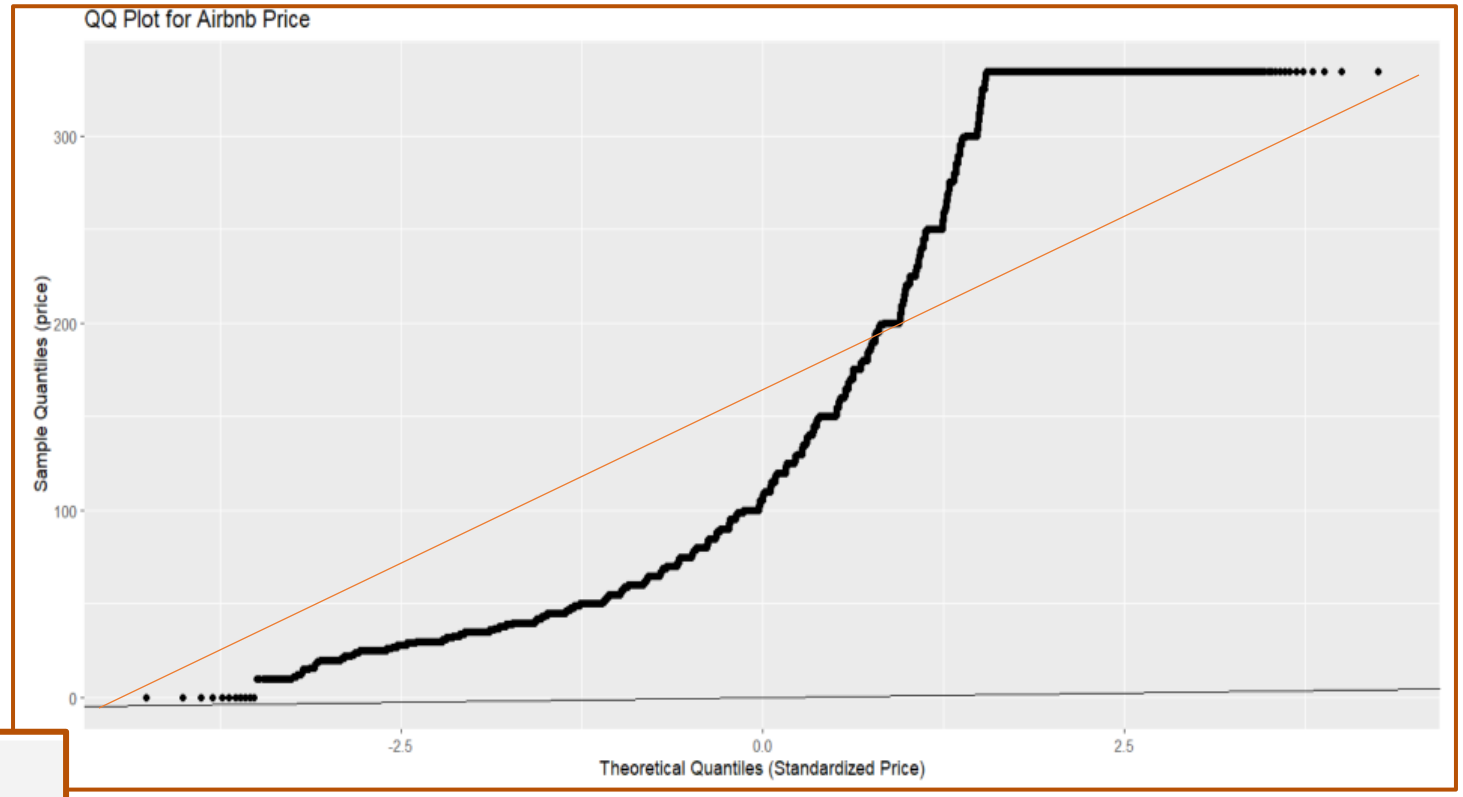
X-axis : price values | Y-axis : frequency





Normality Testing (QQ Plot)

- From this plot, the points appear to deviate from the diagonal line, particularly at the tails of the distribution.
- This suggests that the price variable may not be normally distributed.




```r
# create a QQ plot comparing the price and room_type_new variables
ggplot(data_capped, aes(sample = capped_price)) +
  stat_qq() +
  geom_abline(slope = 1, intercept = 0) +
  xlab("Theoretical Quantiles (Standardized Price)") +
  ylab("Sample Quantiles (price)") +
  ggtitle("QQ Plot for Airbnb Price")
```

X-axis : price values | Y-axis : frequency

Due to Violations of Parametric Test Assumptions:

- **Normality test:** We used Q-Q plots and histogram test to check the normality assumption for each group and found that it was not normally distributed.
- **Homogenity of variance assumption:** Bartlett's test to check the homogeneity of variance assumption.

```
904
905 ```{r}
906 # Bartlett's test for homogeneity of variance
907 bartlett.test(capped_price ~ room_type, data = data_capped)
908 ```
```

```
Bartlett test of homogeneity of variances

data:  capped_price by room_type
Bartlett's K-squared = 4464.1, df = 2, p-value < 2.2e-16
```

Interpretation of Bartlett's test of homogenity of variances:

- The test was performed on the price variable with room_type as the grouping variable.
- Bartlett's K-squared value, which is 4464.1.
- The degrees of freedom (df) are 2, which is the number of groups minus 1.
- The p-value is less than 2.2e-16, which is very small and indicates strong evidence against the null hypothesis of equal variances.

Therefore, we **reject the null hypothesis of equal variances** and conclude that the variances of the price variable are **not equal** across the different room_type groups. With this we proceed with Kruskal Wallis as bartletts test has shown heterogenecity.

Kruskal Wallis test

```
#Using Kruskal Wallis Test to test the null hypothesis

```{r}
# Convert room_type to a factor variable
data_capped$room_type <- as.factor(data_capped$room_type)

# Perform Kruskal-Wallis test
kruskal_test= kruskal.test(capped_price ~ room_type, data = data_capped)
kruskal_test
```
```

Kruskal-Wallis rank sum test

data: capped_price by room_type
Kruskal-Wallis chi-squared = 22434, df = 2, p-value < 2.2e-16

Interpretation :

The Kruskal-Wallis rank sum test was performed to test the hypothesis that the average price of reservations made available by Airbnb is the same across all room types.

The test yielded a chi-squared value of 22434 with 2 degrees of freedom and a **p-value less than 2.2e-16**, indicating that there is **strong evidence to reject the null hypothesis.**

We conclude that the average price of reservations made available by Airbnb is not the same across all room types.

Dunn’s test (Post Hoc test)

```r
# Install the dunn.test package if you haven't already
if (!requireNamespace("dunn.test", quietly = TRUE)) {
  install.packages("dunn.test")
}
# Load the package
library(dunn.test)


dunn_result <- dunn.test(data_capped[["capped_price"]], data_capped[["room_type"]], method = "bonferroni")
  print(dunn_result)
```

```
  Kruskal-Wallis rank sum test

data: x and group
Kruskal-Wallis chi-squared = 22434.2294, df = 2, p-value = 0


                           Comparison of x by group
                                 (Bonferroni)
Col Mean-|
Row Mean |   Entire h   Private
---------+----------------------
Private  |   145.1443
         |    0.0000*
         |
Shared r |   57.33895   12.95393
         |    0.0000*    0.0000*

alpha = 0.05
Reject Ho if p <= alpha/2
$chi2
[1] 22434.23

$Z
[1] 145.14439  57.33895  12.95394

$P
[1] 0.000000e+00 0.000000e+00 1.116001e-38

$P.adjusted
[1] 0.000000e+00 0.000000e+00 3.348003e-38

$comparisons
[1] "Entire home/apt - Private room" "Entire home/apt - Shared room"  "Private room - Shared room"
```

Pairwise comparisons:

- Entire home/apt vs. Private room: Z = 145.1443, adjusted p-value = 0.0000*
- Entire home/apt vs. Shared room: Z = 57.33895, adjusted p-value = 0.0000*
- Private room vs. Shared room: Z = 12.95394, adjusted p-value = 3.348003e-38*

All three pairwise comparisons have adjusted p-values less than the significance level (alpha = 0.05), which means that there are significant differences in the variable "price" between all pairs of room types.

The positive Z values indicate that the Entire home/apt group has a higher mean rank than the Private room and Shared room groups, and the Private room group has a higher mean rank than the Shared room group.

In conclusion, the Kruskal-Wallis test and Dunn's post-hoc test reveal significant differences in the variable “price” among all three room types: Entire home/apt, Private room, and Shared room.

Conclusion of Hypothesis : 1

The mean and median prices for each of the three room types

```r
# Calculate mean and median prices for each room type
library(dplyr)
summary_stats <- data_capped %>%
  group_by(room_type) %>%
  summarize(mean_capped_price = mean(capped_price), median_capped_price = median(capped_price))

# Print summary statistics
print(summary_stats)
```

A tibble: 3 × 3

room_type <chr>	mean_capped_price <dbl>	median_capped_price <dbl>
Entire home/apt	180.20819	160
Private room	82.78738	70
Shared room	64.50345	45

3 rows

For Entire home/apt, the mean capped price is $180.21 and the median capped price is $160. This suggests that the typical price of an entire home or apartment on Airbnb in New York City is around $160-$180 per night after removing outliers.

For Private room, the mean capped price is $82.79 and the median capped price is $70. This suggests that the typical price of a private room on Airbnb in New York City is around $70-$83 per night after removing outliers.

For Shared room, the mean capped price is $64.50 and the median capped price is $45. This suggests that the typical price of a shared room on Airbnb in New York City is around $45-$65 per night after removing outliers.

Overall, the output suggests that Entire home/apt is the most expensive room type, followed by Private room and then Shared room. However, it's important to note that the prices are based on a capped dataset and may not be representative of the entire population of Airbnb listings in New York City.

Hypothesis - 2

Null Hypothesis:

There is **no significant association** between the distance of an Airbnb from the city center and the price of Airbnb.

Alternate Hypothesis:

There is a **significant association** between the distance of an Airbnb from the city center and the price of the Airbnb.

Descriptive Analysis

```r
library(glue)

glue("Price Minimum: {min(data$price)} | Price Maximum: {max(data$price)}")
```

```
Price Minimum: 0 | Price Maximum: 10000
```

Minimum – Maximum values of price

Minimum – Maximum longitude and latitude

```r
glue("Minimum Longitude : {min(data$longitude)} | Maximum Latitude: {max(data$longitude)}")
```

```
Minimum Longitude : -74.24442 | Maximum Latitude: -73.71299
```

```r
glue("Minimum Latitude : {min(data$latitude)} | Maximum Latitude: {max(data$latitude)}")
```

```
Minimum Latitude : 40.49979 | Maximum Latitude: 40.91306
```

Data Cleaning

Checking null values from 'price' , 'latitude' & 'longitude' columns.

```r
null_values_in_price <- sum(is.na(data$price))
print(null_values_in_price)
```

```
[1] 0
```

```r
null_values_in_latitude <- sum(is.na(data$latitude))
print(null_values_in_latitude)
```

```
[1] 0
```

```r
null_values_in_longitude <- sum(is.na(data$longitude))
print(null_values_in_longitude)
```

```
[1] 0
```

We found no null values in the above-mentioned columns.

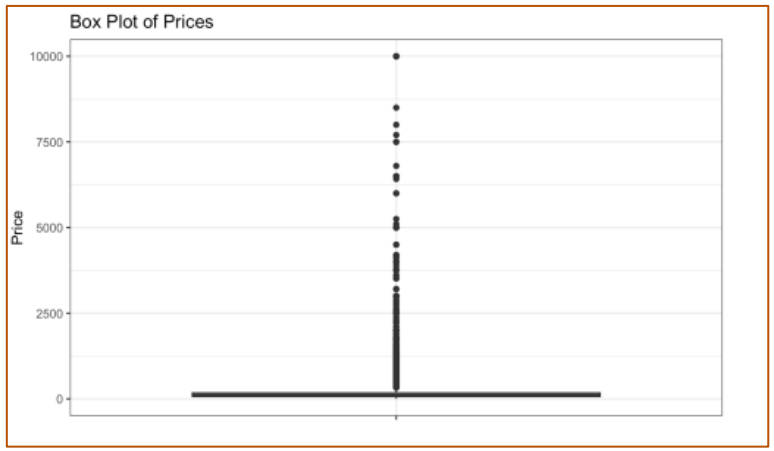
Data Cleaning

ii) Checking for outliers in 'price' , 'latitude' & 'longitude' columns of the New York City Airbnb dataset.

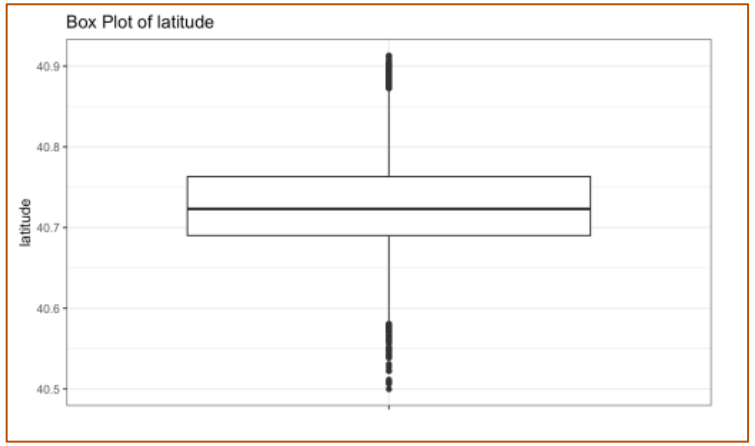
```r
ggplot(data, aes(x = "", y = price)) +
  geom_boxplot() +
  labs(title = "Box Plot of Prices", x = "", y = "Price") +
  theme_bw()
```

From the box plot below, we can observe the presence of outliers

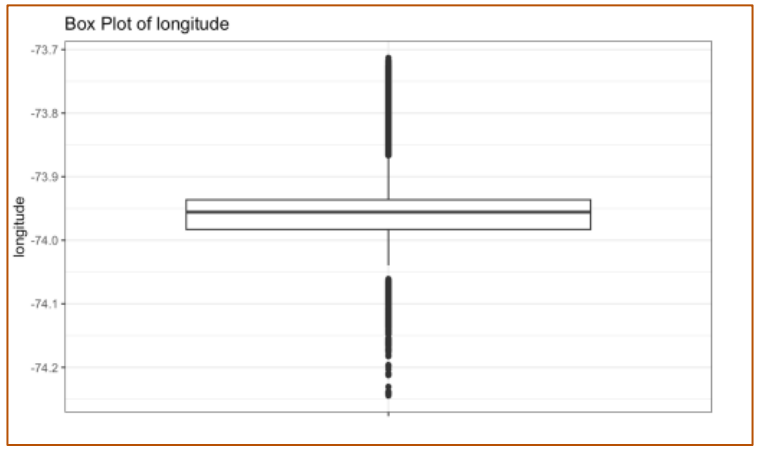
Prices



Latitude



Longitude



Data Cleaning

iii) Treating the outliers using "Capping Method"

```r
```{r}
install.packages("tidyverse")
library(tidyverse)
# Calculate the lower and upper bounds for detecting outliers using the 1.5 * IQR rule for 'price'
price_IQR <- IQR(data$price, na.rm = TRUE)
price_Q1 <- quantile(data$price, 0.25, na.rm = TRUE)
price_Q3 <- quantile(data$price, 0.75, na.rm = TRUE)

lower_bound_price <- price_Q1 - 1.5 * price_IQR
upper_bound_price <- price_Q3 + 1.5 * price_IQR

# Calculate the lower and upper bounds for detecting outliers using the 1.5 * IQR rule for 'latitude'
latitude_IQR <- IQR(data$latitude, na.rm = TRUE)
latitude_Q1 <- quantile(data$latitude, 0.25, na.rm = TRUE)
latitude_Q3 <- quantile(data$latitude, 0.75, na.rm = TRUE)

lower_bound_latitude <- latitude_Q1 - 1.5 * latitude_IQR
upper_bound_latitude <- latitude_Q3 + 1.5 * latitude_IQR

# Calculate the lower and upper bounds for detecting outliers using the 1.5 * IQR rule for 'longitude'
longitude_IQR <- IQR(data$longitude, na.rm = TRUE)
longitude_Q1 <- quantile(data$longitude, 0.25, na.rm = TRUE)
longitude_Q3 <- quantile(data$longitude, 0.75, na.rm = TRUE)

lower_bound_longitude <- longitude_Q1 - 1.5 * longitude_IQR
upper_bound_longitude <- longitude_Q3 + 1.5 * longitude_IQR

# Cap the outliers in the 'price', 'latitude', and 'longitude' columns based on the lower and upper bounds
data_capped <- data %>%
  mutate(
    capped_price = ifelse(price < lower_bound_price, lower_bound_price,
                          ifelse(price > upper_bound_price, upper_bound_price, price)),
    capped_latitude = ifelse(latitude < lower_bound_latitude, lower_bound_latitude,
                             ifelse(latitude > upper_bound_latitude, upper_bound_latitude, latitude)),
    capped_longitude = ifelse(longitude < lower_bound_longitude, lower_bound_longitude,
                              ifelse(longitude > upper_bound_longitude, upper_bound_longitude, longitude))
  )
```

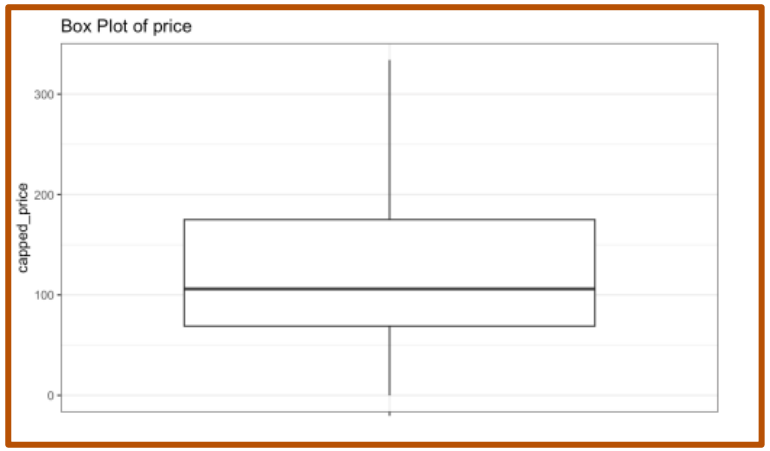
The lower and upper bounds for detecting outliers were calculated using the 1.5 * IQR rule for 'price' , 'latitude' & 'longitude' columns. By capping the outliers at the respective lower and upper bounds, the box plot provides a clearer representation of the central tendency, spread, and non-outlier values
```

Data Cleaning

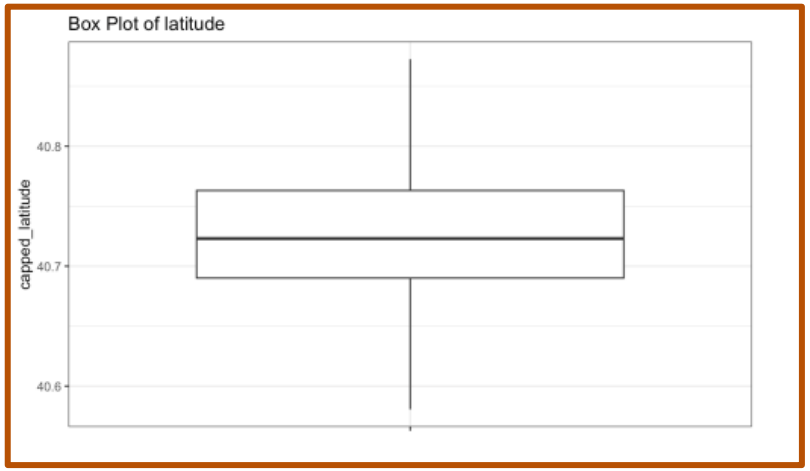
iv) Visualizing 'price' , 'latitude' & 'longitude' columns after outlier removal

```r
```{r}
install.packages("ggplot2")
library(ggplot2)
ggplot(data_capped, aes(x = "", y = capped_price )) +
  geom_boxplot() +
  labs(title = "Box Plot of price", x = "", y = "capped_price") +
  theme_bw()
```
```

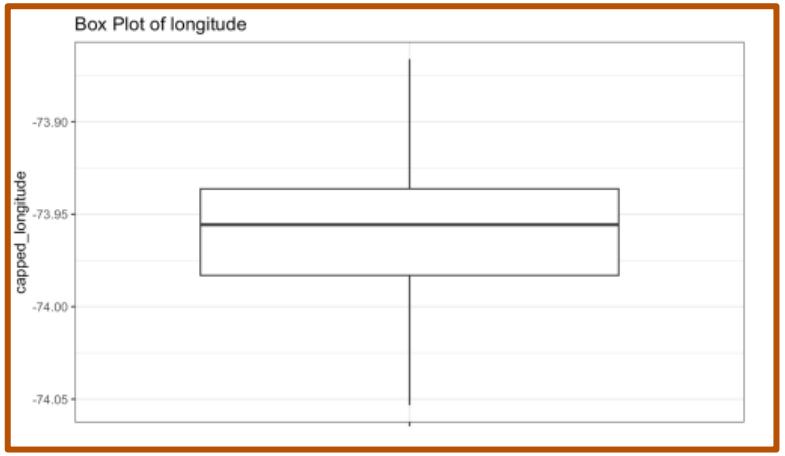
Prices



Latitude



Longitude



- **Defining the coordinates of the city center i.e., Times Square:**

```r
city_center <- c(40.758896, -73.985130) # Times Square
```

- **Calculating the distance of each listing from the city center using the 'distVincentySphere' function from the 'geosphere' package:**

```r
install.packages("geosphere")
library(geosphere)
data_capped$distance_to_center <- distVincentySphere(p1 = data_capped[, c("capped_latitude", "capped_longitude")], p2 =
city_center)
```

- **Calculating the median distance from the city center and creating two sub-categories:**

```r
# Calculate the median distance
median_distance <- median(data_capped$distance_to_center)
data_capped$distance_category <- ifelse(data_capped$distance_to_center <= median_distance,
                                        "close_to_center", "far_from_center")
```

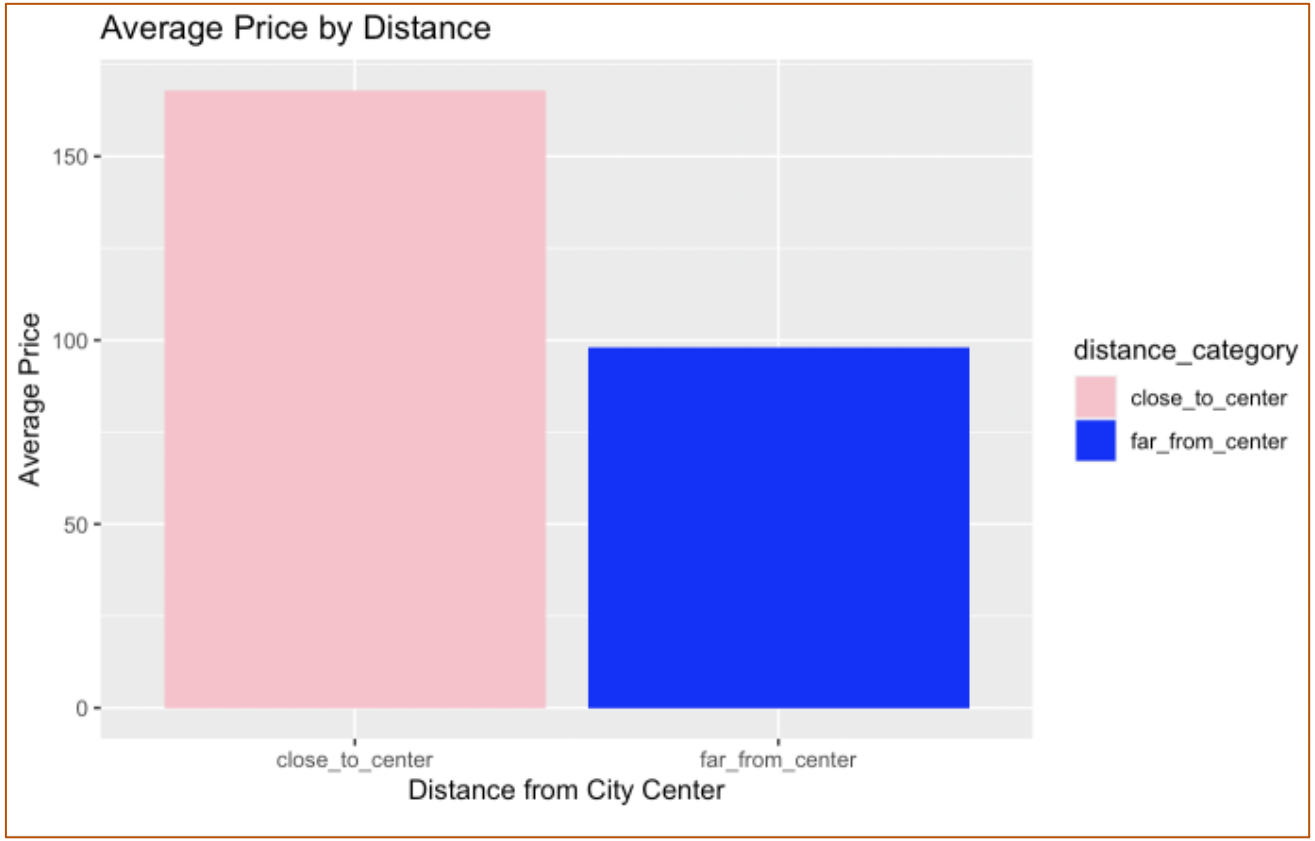
- **After performing outlier treatment and creating sub-groups, we have a new data frame with updated columns.**

capped_price <dbl>	capped_latitude <dbl>	capped_longitude <dbl>	distance_to_center <dbl>	distance_category <chr>
149	40.64749	-73.97237	3705.8157	close_to_center
225	40.75362	-73.98377	221.7602	close_to_center
150	40.80902	-73.94190	5053.1746	far_from_center
89	40.68514	-73.95976	3621.4426	close_to_center
80	40.79851	-73.94399	4738.9179	far_from_center
200	40.74767	-73.97500	1179.2243	close_to_center
60	40.68688	-73.95596	3929.9674	close_to_center
79	40.76489	-73.98493	185.4278	close_to_center
79	40.80178	-73.96723	2388.9329	close_to_center
150	40.71344	-73.99037	1512.7861	close_to_center

Data Visualization

Bar plot visualizing the average price of Airbnb listings for each distance category: "close_to_center" and "far_from_center"

```r
# Calculate the average price for each distance category
library(tidyverse)
avg_price <- data_capped %>%
  group_by(distance_category) %>%
  summarise(avg_price = mean(capped_price))

#Define a colour palette
colors <- c("pink", "blue")

ggplot(avg_price, aes(x = distance_category, y = avg_price, fill = distance_category)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = colors) +
  xlab("Distance from City Center") +
  ylab("Average Price") +
  ggtitle("Average Price by Distance")
```



By comparing height of the bars, we can determine that the average prices of Airbnb that are closer to the city center are high as compared to those far from the center.

Data Visualization

Violin plot visualizing the average price of Airbnb listings for each distance category: "close_to_center" and "far_from_center"

```r
```{r}
# Create a violin plot of price by distance category
ggplot(data_capped, aes(x = distance_category, y = capped_ price, fill = distance_category)) +
  geom_violin() +
  xlab("Distance from City Center") +
  ylab("Price") +
  ggtitle("Violin Plot of Price by Distance")
```
```

Airbnb that are closer to the city center have a wider range of prices, with higher densities at higher price levels.
Airbnb farther from the city center have a narrower distribution of prices, with the density of listings skewed towards lower price levels.

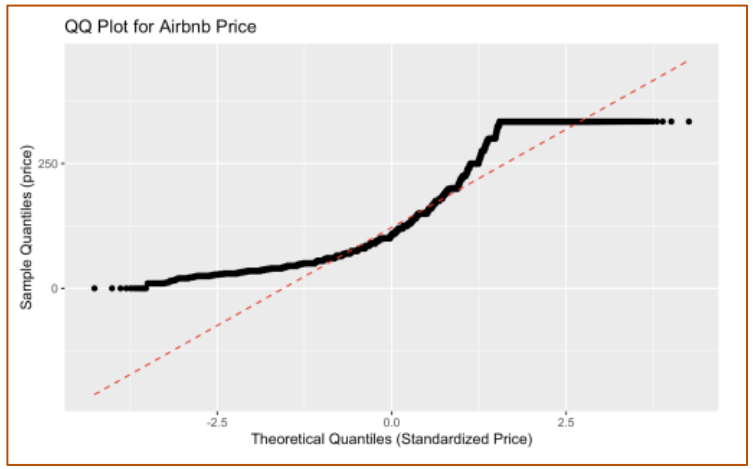
Normality Testing (QQ Plot)

```r
```{r}
library(ggplot2)

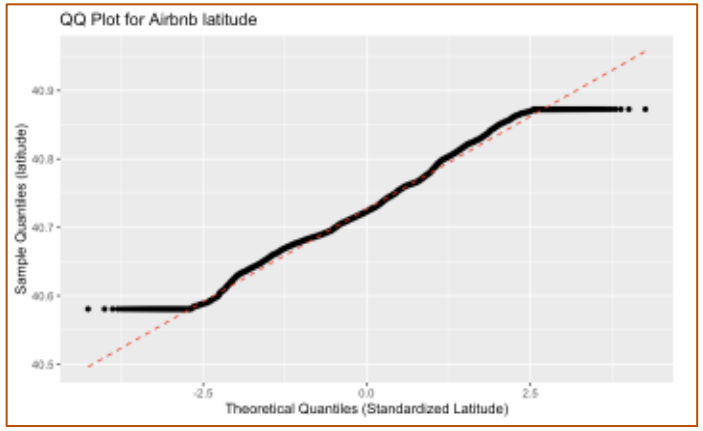
ggplot(data_capped, aes(sample = capped_price)) +
  stat_qq() +
  stat_qq_line(color = "red", linetype = "dashed") +
  xlab("Theoretical Quantiles (Standardized Price)") +
  ylab("Sample Quantiles (price)") +
  ggtitle("QQ Plot for Airbnb Price")

```
```

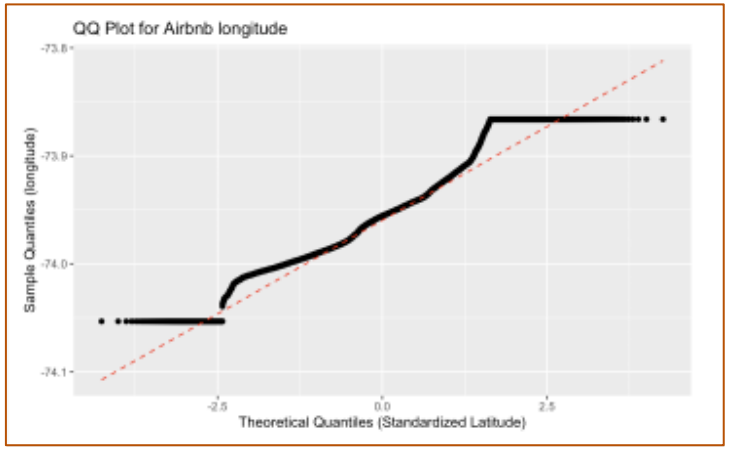
Prices



Latitude



Longitude



The points appear to deviate from the diagonal line, suggesting that the variables may not be normally distributed.

Interpretation(QQ Plot)

Interpretation of QQ plots-

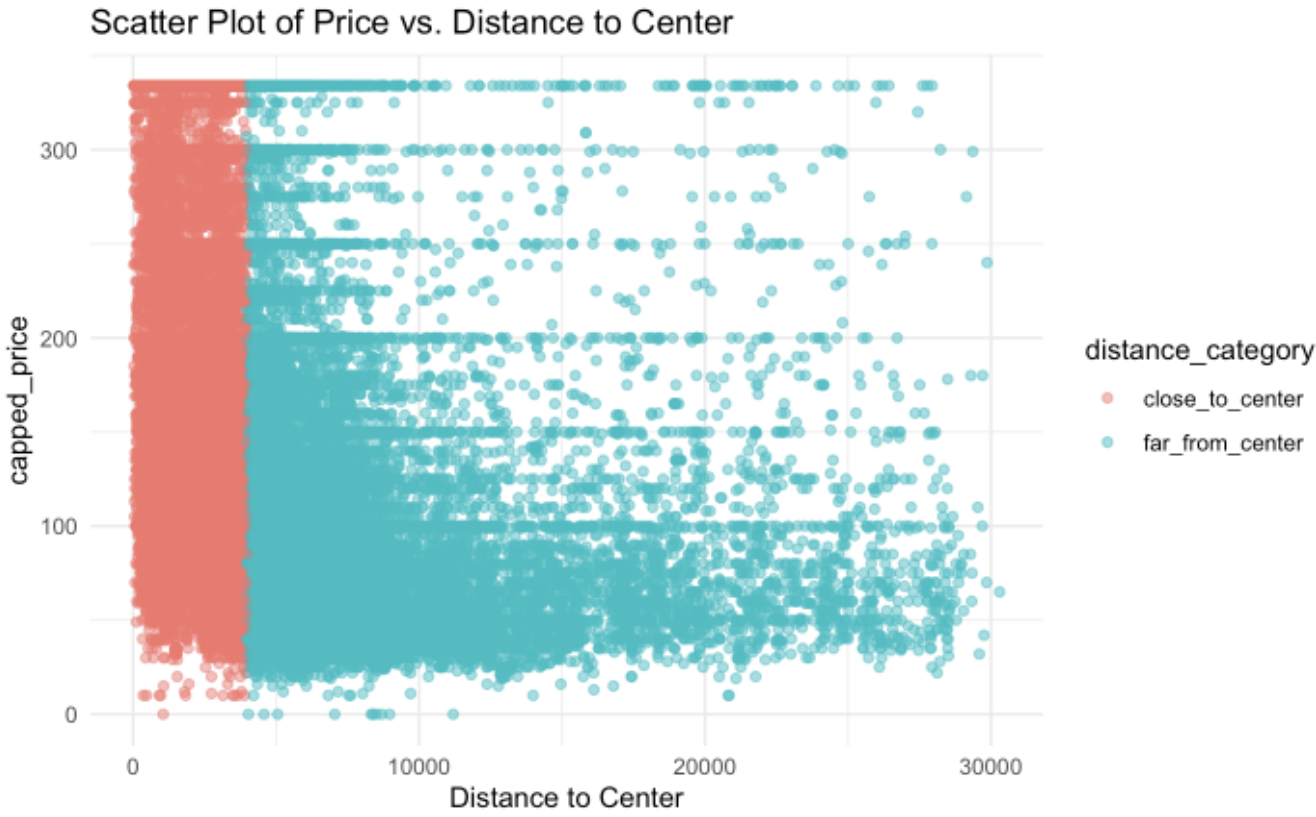
The points in the QQ plots above deviate from the red reference line, it indicates that the 'price' , 'latitude' & 'longitude' columns may not follow a normal distribution. The specific pattern of deviation can give insight into the nature of the data's distribution (e.g., skewness, kurtosis, or other non-normal characteristics).

Therefore, non-parametric test i.e., the Mann-Whitney U test (also known as Wilcoxon rank-sum test) will be conducted.

Assumptions of Mann-Whitney U test

1. Independence of observations:

```r
library(ggplot2)

ggplot(data_capped, aes(x = distance_to_center, y = capped_price, color = distance_category)) +
  geom_point(alpha = 0.5) +
  labs(title = "Scatter Plot of Price vs. Distance to Center",
       x = "Distance to Center",
       y = "capped_price") +
  theme_minimal()
```



There is no clear pattern or clustering of points within each distance category, therefore, it supports the assumption of independence between the observations.

Assumptions of Mann-Whitney U test

2. Identical distributions under the null hypothesis:

```r
# Box plots
library(ggplot2)
ggplot(data_capped, aes(x = distance_category, y = capped_price)) +
  geom_boxplot() +
  labs(title = "Box Plots of Price by Distance Category",
       x = "Distance Category",
       y = "Price") +
  theme_minimal()

# Density plots
ggplot(data_capped, aes(x = capped_price, fill = distance_category)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Plots of Price by Distance Category",
       x = "capped_price",
       y = "Density") +
  theme_minimal()
```

The null hypothesis for the Mann-Whitney U test states that the two populations should have identical distributions. Since the two density curves largely overlap, it indicates that the price distributions are identical between the two categories.

Mann-Whitney U test

Since the assumptions of Mann-Whitney test are fulfilled, we proceed by performing the test

```r
median_distance <- median(data_capped$distance_to_center)
data_capped$distance_category <- ifelse(data_capped$distance_to_center <= median_distance,
                                        "close_to_center", "far_from_center")
```

```r
mann_whitney_test <- wilcox.test(capped_price ~ distance_category, data = data_capped)
print(mann_whitney_test)
```

```
        Wilcoxon rank sum test with continuity correction

data:  capped_price by distance_category
W = 456671250, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

Based on the extremely small p-value (< 2.2 x 10^-16), we reject the null hypothesis in favor of the alternative hypothesis. This suggests that there is a significant association between the distance of an Airbnb listing from the city center and the price of the listing in New York City Airbnb listings.

Validation of Mann-Whitney U test

Bootstrapping: To validate the Mann-Whitney U Test, we will perform Bootstrapping.

It is a resampling technique that can be used to validate the Mann-Whitney U test results. It involves repeatedly sampling from the original data with replacement to create multiple bootstrap samples and perform Mann-Whitney U Test on each sub-sample.

```r
```{r}
# Assuming you have a dataframe named "data_capped" with variables "capped_price" and "distance_category"

# Set the number of bootstrap iterations
n_bootstrap <- 1000

# Create an empty vector to store the bootstrap test statistics
bootstrap_stats <- numeric(n_bootstrap)

# Perform bootstrapping
for (i in 1:n_bootstrap) {
  # Create a bootstrap sample by resampling from the original data with replacement
  bootstrap_sample <- data_capped[sample(nrow(data_capped), replace = TRUE), ]

  # Calculate the Mann-Whitney U test statistic for the bootstrap sample
  bootstrap_stat <- wilcox.test(bootstrap_sample$capped_price ~ bootstrap_sample$distance_category)$statistic

  # Store the bootstrap test statistic
  bootstrap_stats[i] <- bootstrap_stat
}

# Calculate the original Mann-Whitney U test statistic
original_stat <- mann_whitney_test$statistic

# Calculate the p-value by comparing the original statistic to the bootstrap distribution
p_value <- sum(bootstrap_stats >= original_stat) / n_bootstrap

# Display the p-value
cat("p-value:", p_value, "\n")
```
```

Interpretation:
This means that Mann Whitney U test was incorrectly testing the hypothesis.

That is, the observed difference in the original Mann-Whitney U test statistic is likely due to random variation rather than a true difference between the groups.

Permutation Testing

To further validate the hypothesis, we performed Permutation test.

It is a non-parametric test that can be used to compare distributions of two groups. It involves randomly permuting the observations between the groups and calculating the test statistic many times to estimate the null distribution.

```
# Define the observed test statistic (e.g., mean, median, etc.)
observed_stat <- median(close_to_center) - median(far_from_center)

# Set the number of permutations
n_permutations <- 1000

# Create an empty vector to store the permutation test statistics
perm_stats <- numeric(n_permutations)

# Perform the permutation test
for (i in 1:n_permutations) {
  # Combine the data and shuffle the distance category labels
  combined_data <- c(close_to_center, far_from_center)
  shuffled_labels <- sample(c(rep("close_to_center", length(close_to_center)),
                              rep("far_from_center", length(far_from_center))))

  # Calculate the test statistic for the permuted data
  perm_stat <- median(combined_data[shuffled_labels == "close_to_center"]) -
    median(combined_data[shuffled_labels == "far_from_center"])

  # Store the permutation test statistic
  perm_stats[i] <- perm_stat
}

# Calculate the p-value by comparing the observed statistic to the null distribution
p_value <- sum(abs(perm_stats) >= abs(observed_stat)) / n_permutations

# Display the p-value
cat("p-value:", p_value, "\n")
```

The p-value of 0 suggests strong evidence against the null hypothesis.
Therefore, the observed difference is not due to random variation, as shown by bootstrapping, but rather indicates a true difference in pricing between the two categories.

Conclusion of Hypothesis 2

```r
# install.packages("dplyr")

# Load the dplyr library
library(dplyr)

# Calculate summary statistics for price variation by distance category
price_summary <- data_capped %>%
  group_by(distance_category) %>%
  summarize(
    min_price = min(capped_price),
    q1_price = quantile(capped_price, 0.25),
    median_price = median(capped_price),
    mean_price = mean(capped_price),
    q3_price = quantile(capped_price, 0.75),
    max_price = max(capped_price)
  )

# Print the summary statistics
print(price_summary)
```

- Based on our comprehensive analysis, supported by rigorous statistical testing and validation, we reject the null hypothesis.
- Based on the summary statistics, we observe that Airbnbs closer to the city center tend to have higher prices compared to those farther from the center.
- The mean price is 168.03$ for properties categorized as "close_to_center", while it is 97.92$ for properties that are "far_from_center".
- The median price is 150$ for properties categorized as "close_to_center", while it is 79$ for properties that are "far_from_center".

distance_category <chr>	min_price <dbl>	q1_price <dbl>	median_price <dbl>	mean_price <dbl>	q3_price <dbl>	max_price <dbl>
close_to_center	0	99	150	168.03505	225	334
far_from_center	0	55	79	97.92302	120	334

Limitations

Kruskal-Wallis Test :

- Sensitive to outliers: Outliers can influence results.
- Loss of information: Test is based on ranks, potentially losing information from ties.
- Applicable only to independent groups: Not suitable for paired or repeated measures data.
- No directional information: Indicates significant differences but not direction.
- Requires post hoc tests: Additional tests needed for pairwise comparisons (e.g., Dunn's test).

Mann-Whitney U Test :

- Limited to two groups: Cannot compare more than two groups.
- Sensitive to outliers: Outliers can impact results.
- Loss of information: Test is based on ranks, potentially losing information from ties.
- Applicable only to independent groups: Not suitable for paired or repeated measures data.
- No directional information: Indicates significant differences but not direction.
- No effect size: Does not provide magnitude of the difference between groups.


THANK YOU