Project

Trinath Sai Subhash Reddy Pittala, Uma Maheswara R Meleti, Hemanth Vasireddy

2023-03-24

Introduction

Airbnb Price Determinants in Europe

We want to work on Airbnb's dataset from kaggle.com. It provides information about hotel rooms in Europe.

Each major city has its dataset for weekends and weekdays Variables included in the dataset: Host ID (Id) The total price of listing (realSum) Room type: private, shared, entire home, apt (room_type) Whether or not a room is shared (room_shared) Max number of people allowed in property (person_capacity) Whether or not the host is superhost (host_is_superhost) Whether or not it is multiple rooms (multi) Whether for business or family use (biz) Distance from the city center (dist) Distance from nearest metro (metro_dist) Latitude and longitude (lat long) Guest satisfaction (guest_satisfaction_overall) Cleanliness (cleanliness_rating) The total quantity of bedrooms available among all properties for a single host (bedrooms)

Questions we can answer with the dataset: Price Forecasting: use pricing, room type, and amenities to predict potential rental prices in the future. Hotspots: use listing location in relation to business and tourism centers and correlate this with pricing to determine where Airbnb rentals would be most profitable Customer sentiment analysis: analyze customer comments and satisfaction ratings to evaluate listing on overall customer experience and use it to optimize hosts' services to improve user satisfaction ratings.

How can this information be used: Data can help travelers find accommodation that meets their needs without exceeding budget. Can help hosts set competitive pricing and optimize listings to get more bookings. Help investors evaluate the value of investing in real estate in different European cities based on pricing trends.

Pre Processing and Cleaning the Data

Data loading

```
# Set the relative directory path
my_dir <- "./archive"

# List all the files in the directory
files <- list.files(path = my_dir, full.names = TRUE)
```

Combining the Data from all Files

```r
# Get a list of all the csv files in the directory
file_list <- list.files(path = my_dir, pattern = "*.csv", full.names = TRUE)

# Initialize an empty list to store the data frames
df_list <- list()

# Loop through each file and read it into a data frame
for (i in seq_along(file_list)) {
    df <- read.csv(file_list[i])

    # Add a new column with the city_day
    df$city_day <- basename(file_list[i])

    # Append the data frame to the list
    df_list[[i]] <- df
}

# Combine all the data frames into a single dataset
my_data <- bind_rows(df_list)

# Removing the .csv ext
my_data$city_day <- gsub("\\.csv", "", my_data$city_day)

# Print the first few rows of the data
head(my_data)
```

```
##   X  realSum    room_type room_shared room_private person_capacity
## 1 0 194.0337 Private room       False         True               2
## 2 1 344.2458 Private room       False         True               4
## 3 2 264.1014 Private room       False         True               2
## 4 3 433.5294 Private room       False         True               4
## 5 4 485.5529 Private room       False         True               2
## 6 5 552.8086 Private room       False         True               3
##   host_is_superhost multi biz cleanliness_rating guest_satisfaction_overall
## 1             False     1   0                 10                         93
## 2             False     0   0                  8                         85
## 3             False     0   1                  9                         87
## 4             False     0   1                  9                         90
## 5              True     0   0                 10                         98
## 6             False     0   0                  8                        100
##   bedrooms      dist metro_dist attr_index attr_index_norm rest_index
## 1        1 5.0229638  2.5393800   78.69038        4.166708   98.25390
## 2        1 0.4883893  0.2394039  631.17638       33.421209  837.28076
## 3        1 5.7483119  3.6516213   75.27588        3.985908   95.38695
## 4        2 0.3848620  0.4398761  493.27253       26.119108  875.03310
## 5        1 0.5447382  0.3186926  552.83032       29.272733  815.30574
## 6        2 2.1314201  1.9046682  174.78896        9.255191  225.20166
##   rest_index_norm     lng      lat           city_day
## 1        6.846473 4.90569 52.41772 amsterdam_weekdays
## 2       58.342928 4.90005 52.37432 amsterdam_weekdays
## 3        6.646700 4.97512 52.36103 amsterdam_weekdays
## 4       60.973565 4.89417 52.37663 amsterdam_weekdays
## 5       56.811677 4.90051 52.37508 amsterdam_weekdays
```

```
## 6       15.692376 4.87699 52.38966 amsterdam_weekdays
```

Spilt Training and Testing Data

```
set.seed(123456789)
my_data_train <- my_data[sample(nrow(my_data), 0.7 * nrow(my_data)),
    ]
my_data_test <- my_data[setdiff(1:nrow(my_data), rownames(my_data_train)),
    ]
head(my_data_train)
```

```
##          X  realSum       room_type room_shared room_private person_capacity
## 44098  462 368.6905 Entire home/apt       False        False               4
## 47120 3484 240.3385 Entire home/apt       False        False               4
## 29708 2631 257.7671    Private room       False         True               2
## 9772   856 485.9543 Entire home/apt       False        False               2
## 16896  196 269.4653 Entire home/apt       False        False               4
## 2325   244 220.2798 Entire home/apt       False        False               6
##       host_is_superhost multi biz cleanliness_rating guest_satisfaction_overall
## 44098             False     0   1                 8                          80
## 47120             False     0   1                 8                         100
## 29708             False     0   0                10                         100
## 9772              False     0   0                 9                          84
## 16896             False     0   1                10                          93
## 2325              False     1   0                10                          98
##       bedrooms      dist metro_dist attr_index attr_index_norm rest_index
## 44098        1 1.4966090  0.5553737 1758.54461       38.961335  2076.9803
## 47120        1 2.7716179  0.5800582  574.88024       12.736727  1479.5874
## 29708        1 6.5817328  2.0478099  172.41005       11.984895   385.5078
## 9772         1 1.0793735  0.3433846  472.88903       18.258792  1141.3205
## 16896        1 0.8901721  0.5452364  339.08856       11.194767   684.5563
## 2325         3 2.0502381  0.5606083   74.35373        2.803444   107.7512
##       rest_index_norm      lng      lat           city_day
## 44098       45.252362 12.48423 41.90030      rome_weekends
## 47120       32.236620 12.46991 41.90711      rome_weekends
## 29708        6.899917 -0.18190 51.45989    london_weekends
## 9772        25.070976  2.16725 41.37764 barcelona_weekends
## 16896       30.615911 -9.12964 38.71413    lisbon_weekdays
## 2325         8.090617 23.75717 37.98218    athens_weekdays
```

```
head(my_data_test)
```

```
##     X  realSum       room_type room_shared room_private person_capacity
## 3   2 264.1014    Private room       False         True               2
## 4   3 433.5294    Private room       False         True               4
## 10  9 276.5215    Private room       False         True               2
## 12 11 319.6401    Private room       False         True               2
## 17 16 368.8515    Private room       False         True               2
## 22 21 933.8458 Entire home/apt       False        False               4
##    host_is_superhost multi biz cleanliness_rating guest_satisfaction_overall
## 3              False     0   1                 9                          87
```

```
## 4              False     0   1                 9                          90
## 10             False     1   0                10                          88
## 12              True     1   0                10                          97
## 17             False     0   0                10                          90
## 22             False     0   0                10                          96
##    bedrooms     dist metro_dist attr_index attr_index_norm rest_index
## 3         1 5.748312  3.6516213   75.27588        3.985908   95.38695
## 4         2 0.384862  0.4398761  493.27253       26.119108  875.03310
## 10        1 3.142361  0.9244044  206.25286       10.921226  238.29126
## 12        1 2.182707  1.5903814  191.50134       10.140123  229.29740
## 17        1 1.327797  0.1195281  539.01288       28.541090  573.89657
## 22        2 1.014066  0.3771037  477.79407       25.299513  664.05325
##    rest_index_norm     lng      lat           city_day
## 3          6.64670 4.97512 52.36103 amsterdam_weekdays
## 4         60.97357 4.89417 52.37663 amsterdam_weekdays
## 10        16.60448 4.87600 52.34700 amsterdam_weekdays
## 12        15.97777 4.92496 52.37107 amsterdam_weekdays
## 17        39.98994 4.88971 52.36148 amsterdam_weekdays
## 22        46.27219 4.89088 52.36422 amsterdam_weekdays
```

Filtering out the Outliers from Data Out of IQR Ranges

```r
# Initialize an empty list to store the outliers
outliers_list <- list()

# Initialize an empty list to store the filtered data
# frames
df_list_filtered <- list()

# Loop through each file and read it into a data frame
# after removing outliers
for (i in seq_along(file_list)) {
    df_filtered <- read.csv(file_list[i])

    # Add a new column with the city_day
    df_filtered$city_day <- gsub("\\.csv", "", basename(file_list[i]))

    iqr_var1 <- IQR(df_filtered$realSum)

    # Calculate the upper and lower bounds for each
    # variable
    upper_var1 <- quantile(df_filtered$realSum, 0.75) + 1.5 *
        iqr_var1
    lower_var1 <- quantile(df_filtered$realSum, 0.25) - 1.5 *
        iqr_var1

    # Filter the data based on the upper and lower bounds
    # for each variable
    filtered_data <- filter(df_filtered, realSum > lower_var1 &
        realSum < upper_var1)

    # Append the filtered data frame to the list
```

```
    df_list_filtered[[i]] <- filtered_data

    # Get the rows that were removed while filtering
    outliers <- anti_join(df_filtered, filtered_data)

    # Append the outliers to the list
    outliers_list[[i]] <- outliers
}


# Combine all the filtered data frames into a single
# dataset
my_data_filtered <- bind_rows(df_list_filtered)

# Removing the .csv ext
my_data_filtered$city_day <- gsub("\\.csv", "", my_data_filtered$city_day)

# summary(my_data_filtered)

# Combine all the outliers into a single dataset
my_outliers <- bind_rows(outliers_list)

# Removing the .csv ext
my_outliers$city_day <- gsub("\\.csv", "", my_outliers$city_day)

summary(my_outliers)
```

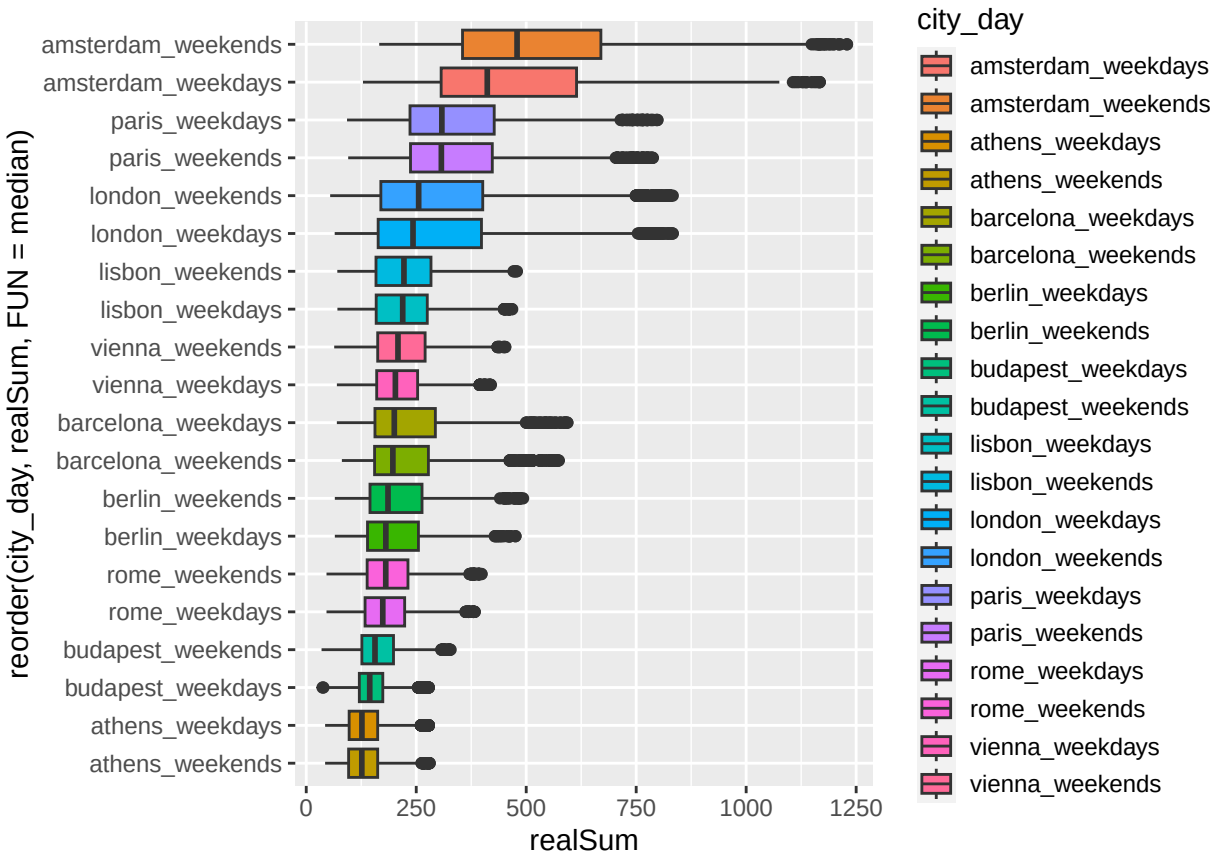
```
##        X           realSum         room_type         room_shared
##  Min.   :   0   Min.   :  279.4   Length:2737        Length:2737
##  1st Qu.: 666   1st Qu.:  469.2   Class :character   Class :character
##  Median :1237   Median :  691.9   Mode  :character   Mode  :character
##  Mean   :1614   Mean   :  915.5
##  3rd Qu.:2310   3rd Qu.:  996.3
##  Max.   :5374   Max.   :18545.5
##  room_private       person_capacity host_is_superhost      multi
##  Length:2737        Min.   :2.000   Length:2737        Min.   :0.000
##  Class :character   1st Qu.:4.000   Class :character   1st Qu.:0.000
##  Mode  :character   Median :5.000   Mode  :character   Median :0.000
##                     Mean   :4.628                      Mean   :0.221
##                     3rd Qu.:6.000                      3rd Qu.:0.000
##                     Max.   :6.000                      Max.   :1.000
##       biz         cleanliness_rating guest_satisfaction_overall    bedrooms
##  Min.   :0.0000   Min.   : 2.000     Min.   : 20.00              Min.   :0.000
##  1st Qu.:0.0000   1st Qu.: 9.000     1st Qu.: 91.00              1st Qu.:1.000
##  Median :0.0000   Median :10.000     Median : 97.00              Median :2.000
##  Mean   :0.4965   Mean   : 9.509     Mean   : 93.65              Mean   :1.886
##  3rd Qu.:1.0000   3rd Qu.:10.000     3rd Qu.:100.00              3rd Qu.:2.000
##  Max.   :1.0000   Max.   :10.000     Max.   :100.00              Max.   :6.000
##       dist            metro_dist         attr_index     attr_index_norm
##  Min.   : 0.01504   Min.   :0.006171   Min.   :  20.5   Min.   :  1.468
##  1st Qu.: 1.04119   1st Qu.:0.218081   1st Qu.: 225.1   1st Qu.: 11.719
##  Median : 1.89579   Median :0.352339   Median : 385.0   Median : 17.958
##  Mean   : 2.30674   Mean   :0.498426   Mean   : 456.2   Mean   : 20.892
```

```
##  3rd Qu.: 3.00820   3rd Qu.:0.576430   3rd Qu.: 610.6   3rd Qu.: 25.953
##  Max.   :21.29515   Max.   :8.918036   Max.   :2040.4   Max.   :100.000
##    rest_index     rest_index_norm        lng               lat
##  Min.   :  27.9   Min.   :  0.667   Min.   :-9.22476   Min.   :37.96
##  1st Qu.: 408.5   1st Qu.: 14.187   1st Qu.:-0.06677   1st Qu.:41.41
##  Median : 739.9   Median : 30.001   Median : 4.88384   Median :47.51
##  Mean   : 904.9   Mean   : 31.734   Mean   : 7.88764   Mean   :45.93
##  3rd Qu.:1269.7   3rd Qu.: 45.426   3rd Qu.:13.44666   3rd Qu.:51.50
##  Max.   :4183.1   Max.   :100.000   Max.   :23.75400   Max.   :52.58
##    city_day
##  Length:2737
##  Class :character
##  Mode  :character
##
##
##
```

Exploratory Data Analysis

Boxplot of Price Vs City

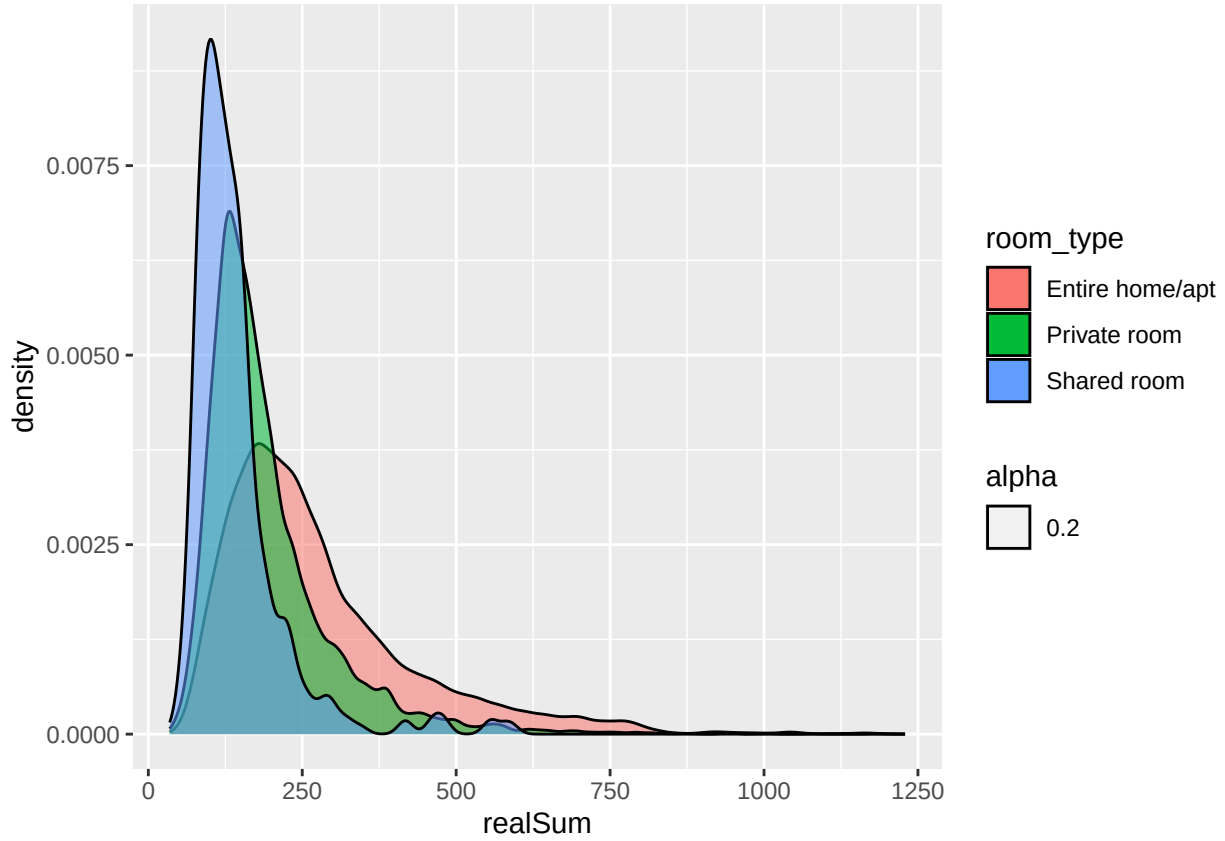
```
ggplot(my_data_filtered, aes(x = reorder(city_day, realSum, FUN = median),
    y = realSum, fill = city_day)) + geom_boxplot() + coord_flip() +
    theme(legend.key.height = unit(0.5, "cm"), legend.key.size = unit(1,
        "lines"))
```



The highest prices in europe are found in amsterdam.

Density plot of Price vs Room type

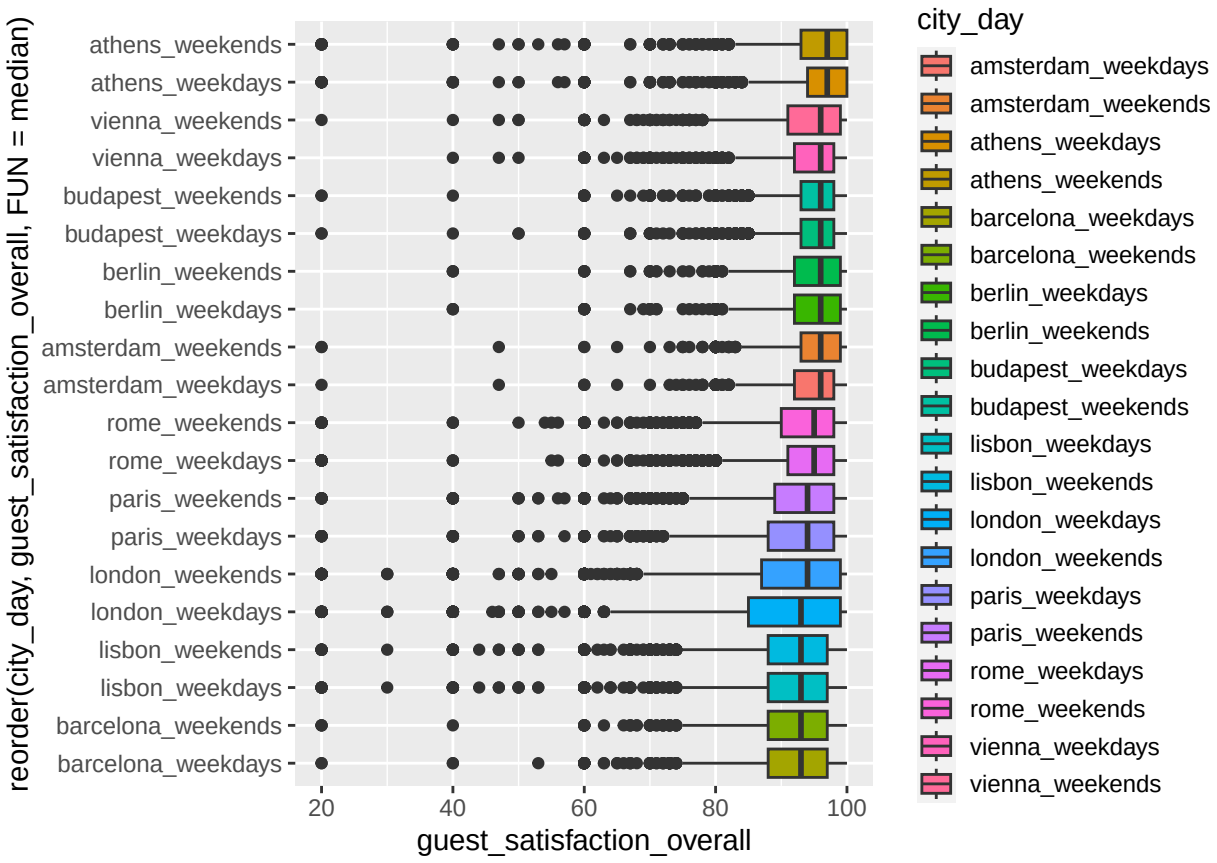
```
ggplot(my_data_filtered, aes(x = realSum, group = room_type,
    fill = room_type, alpha = 0.2)) + geom_density()
```



The prices of entire home are high comparitively

Boxplot of City vs Guest Satisfaction

```
ggplot(my_data_filtered, aes(x = reorder(city_day, guest_satisfaction_overall,
    FUN = median), y = guest_satisfaction_overall, fill = city_day)) +
    geom_boxplot() + coord_flip() + theme(legend.key.height = unit(0.5,
    "cm"), legend.key.size = unit(1, "lines"))
```



This plot shows there is no major difference in Guest Satisfaction vs City.

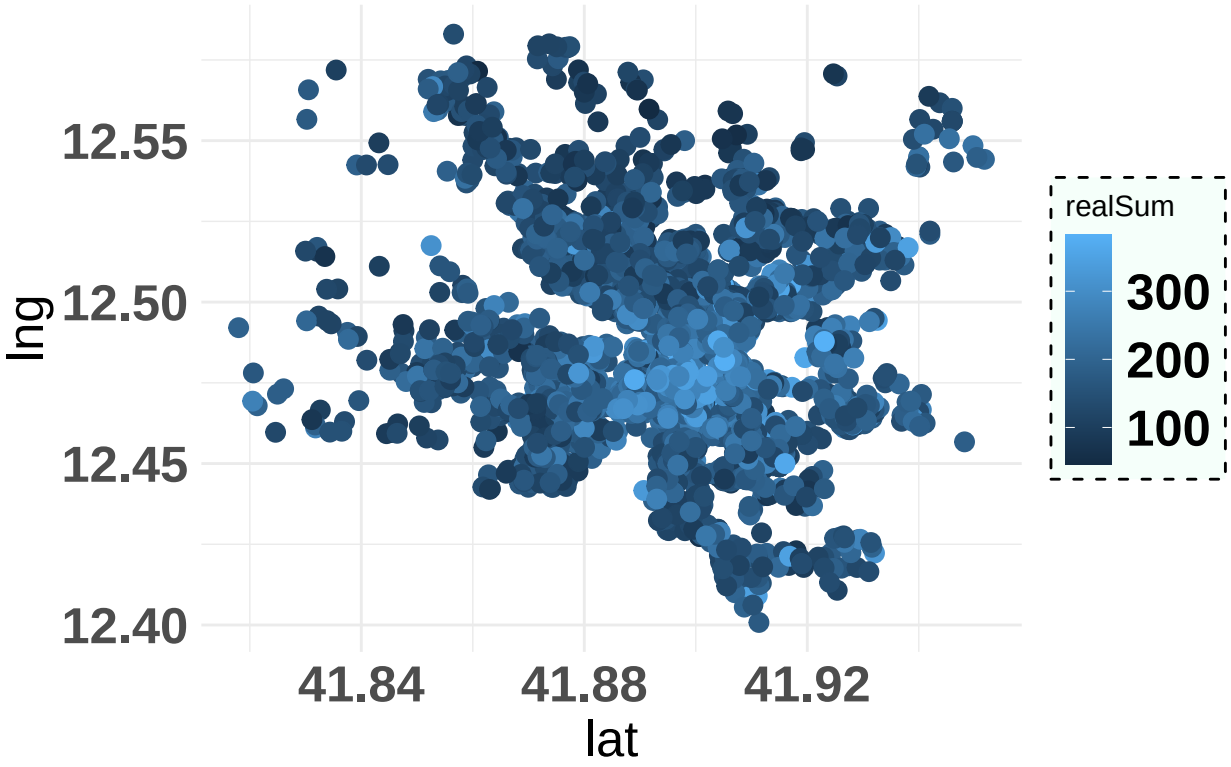
Scatterplot of Price vs Guest Satisfaction filtered by city

```
ggplot(my_data_filtered, aes(x = realSum, y = guest_satisfaction_overall,
    color = city_day)) + geom_point() + xlab("Price") + ylab("Guest Satisfaction Overall") +
    scale_color_discrete(name = "City-Day")
```

This plot implies there are good cheaper Airbnb at most cities which give higher guest satisfaction rating

Scatterplot of Prices in Rome w.r.t Latitude and Longitude during weekdays

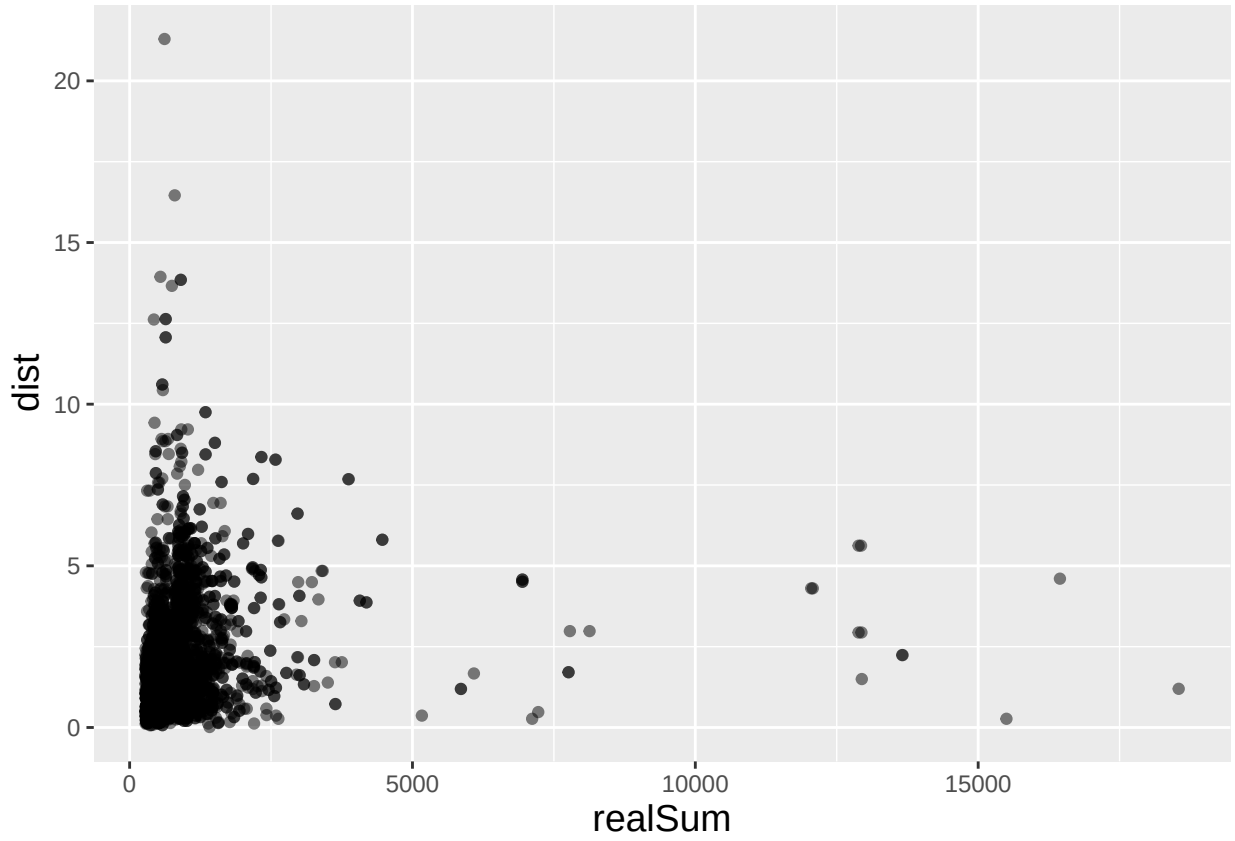
```
tema <- theme(plot.title = element_text(size = 23, hjust = 0.5),
    axis.text.x = element_text(size = 19, face = "bold"), axis.text.y = element_text(size = 19,
        face = "bold"), axis.title.x = element_text(size = 19),
    axis.title.y = element_text(size = 19), legend.text = element_text(colour = "black",
        size = 19, face = "bold"), legend.background = element_rect(fill = "#F5FFFA",
        size = 0.5, linetype = "dashed", colour = "black"))

rome_data <- my_data_filtered %>%
    subset(city_day == "rome_weekdays")

ggplot(data = rome_data, mapping = aes(x = lat, y = lng)) + theme_minimal() +
    scale_fill_identity() + geom_point(mapping = aes(color = realSum),
    size = 3) + ggtitle("") + tema
```



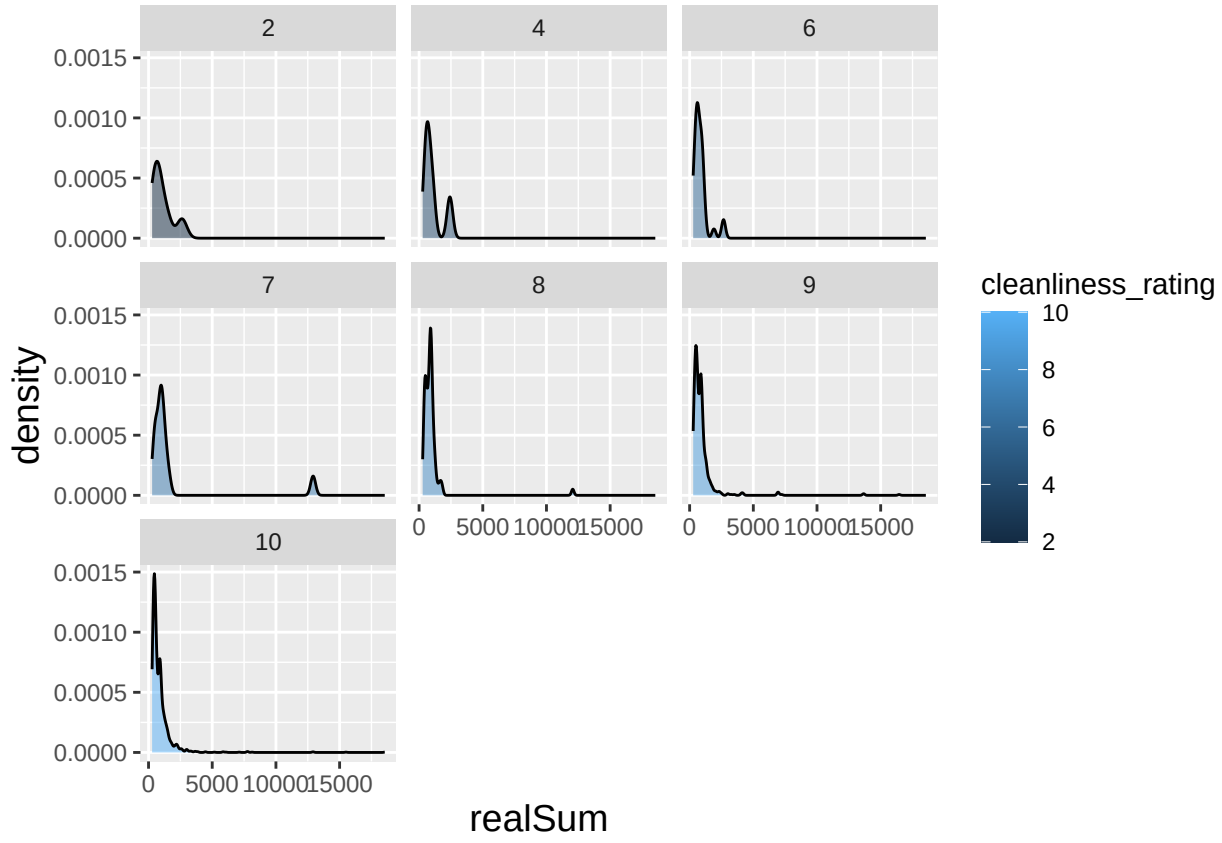
This plot is within expectations of game theory, which suggests similar types of establishments (price and hospitality) tend be in clusters.

Outlier Analysis

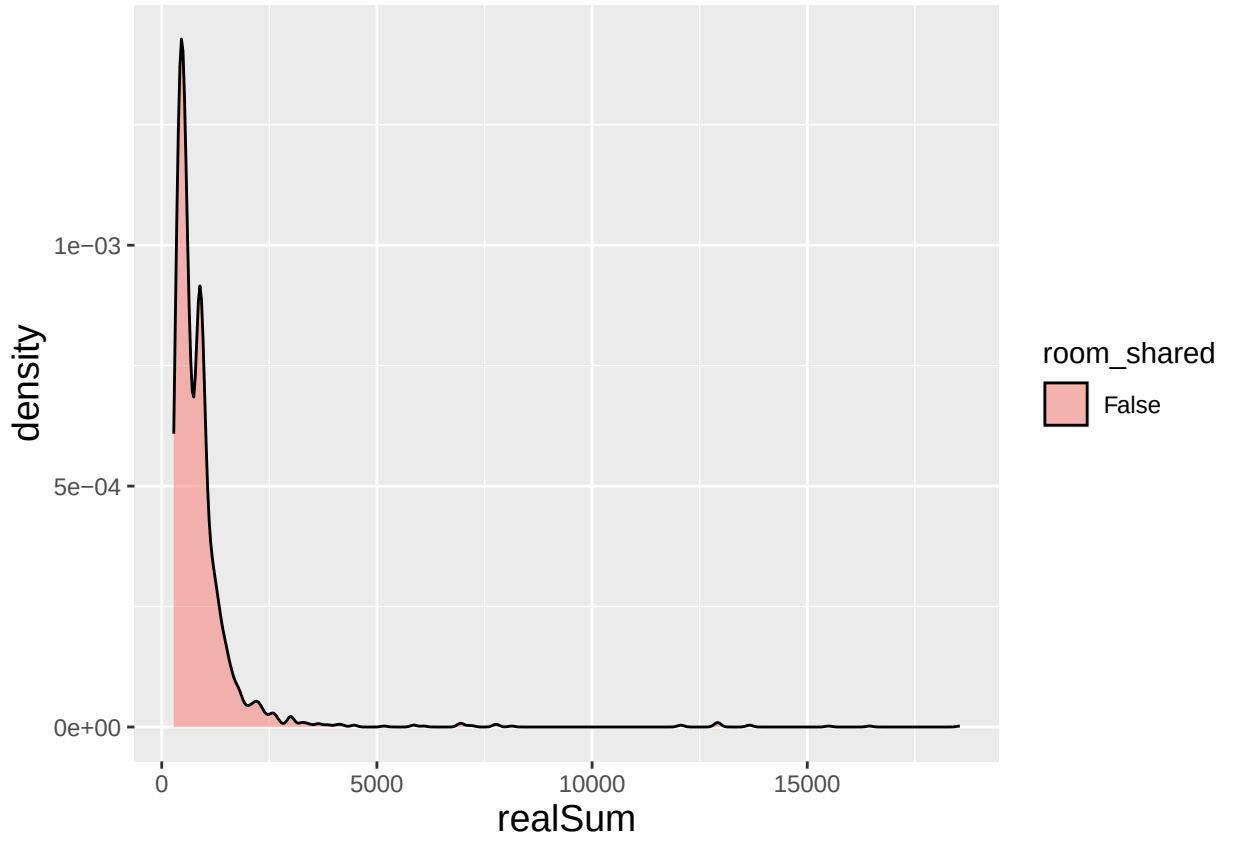
```
ggplot(my_outliers, aes(x = realSum, y = dist)) + geom_point(alpha = 0.5) +
    theme(axis.title.x = element_text(size = 14), axis.title.y = element_text(size = 14))
```



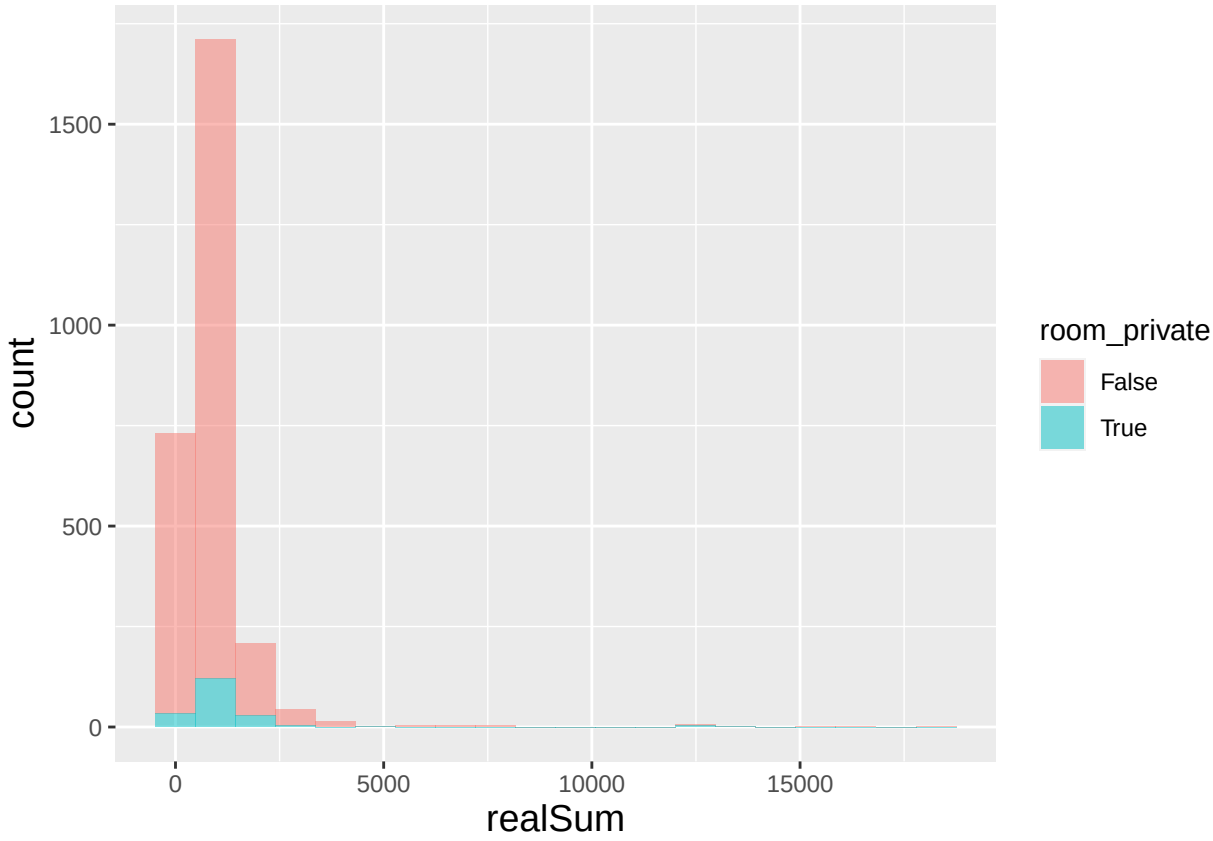

```
ggplot(my_outliers, aes(x = realSum, fill = cleanliness_rating,
    group = cleanliness_rating)) + geom_density(alpha = 0.5) +
    theme(axis.title.x = element_text(size = 14), axis.title.y = element_text(size = 14)) +
    facet_wrap(~cleanliness_rating)
```



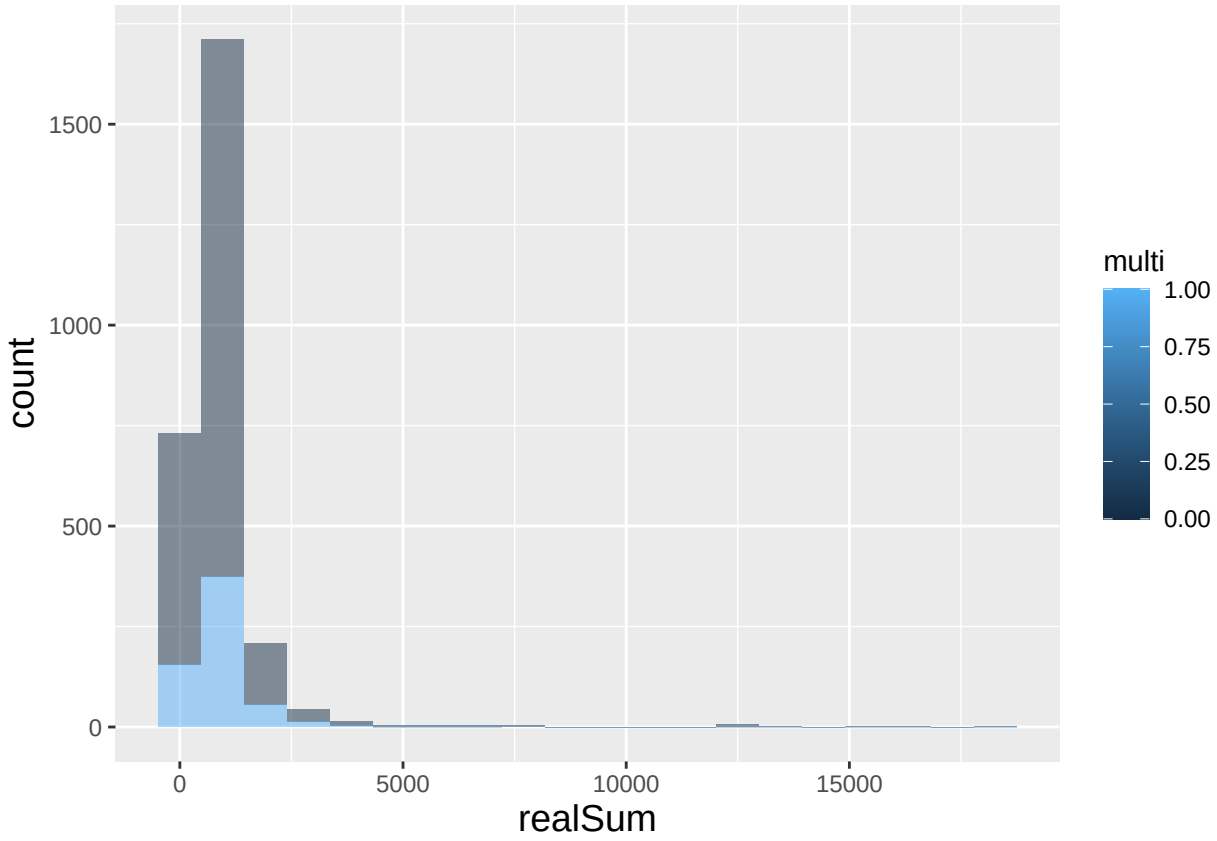

```
ggplot(my_outliers, aes(x = realSum, fill = room_shared, group = room_shared)) +
    geom_density(alpha = 0.5) + theme(axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14))
```



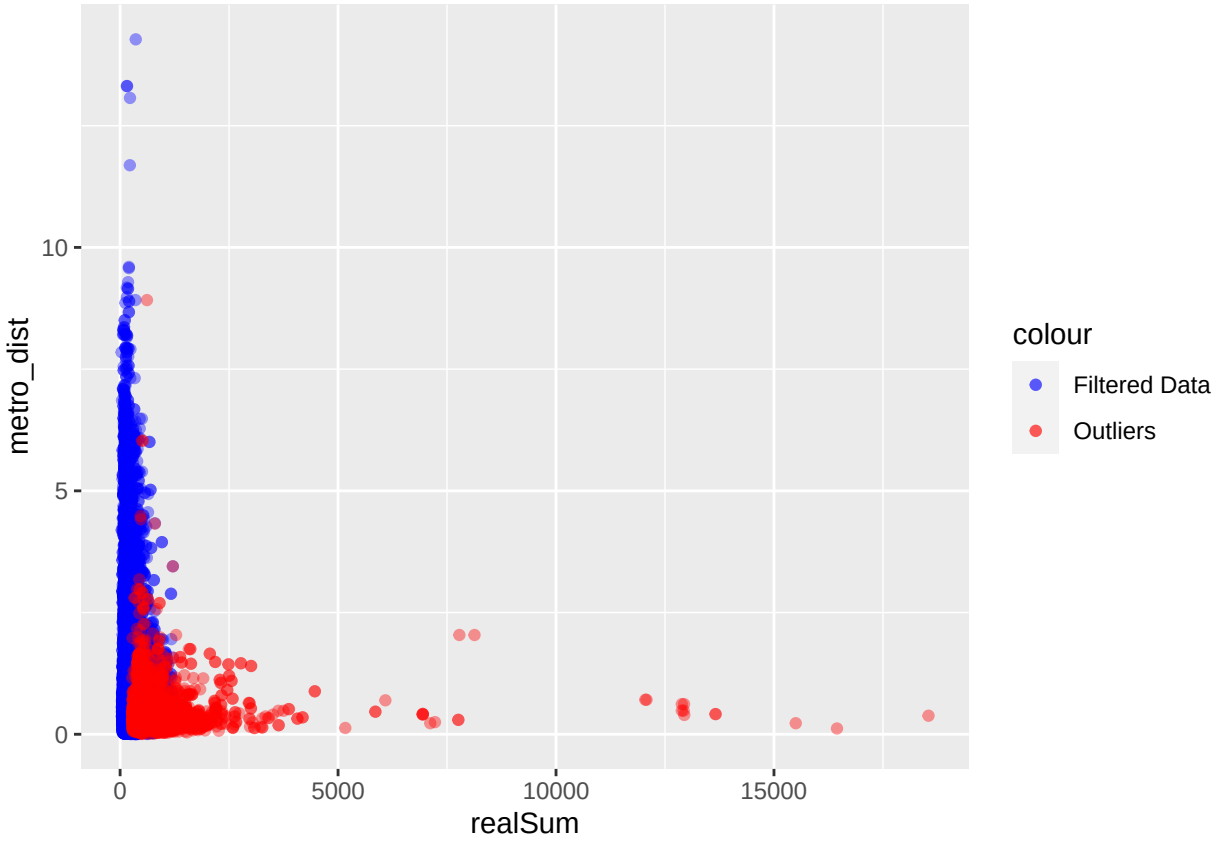

```
ggplot(my_outliers, aes(x = realSum, fill = room_private, group = room_private)) +
    geom_histogram(alpha = 0.5, bins = 20) + theme(axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14))
```



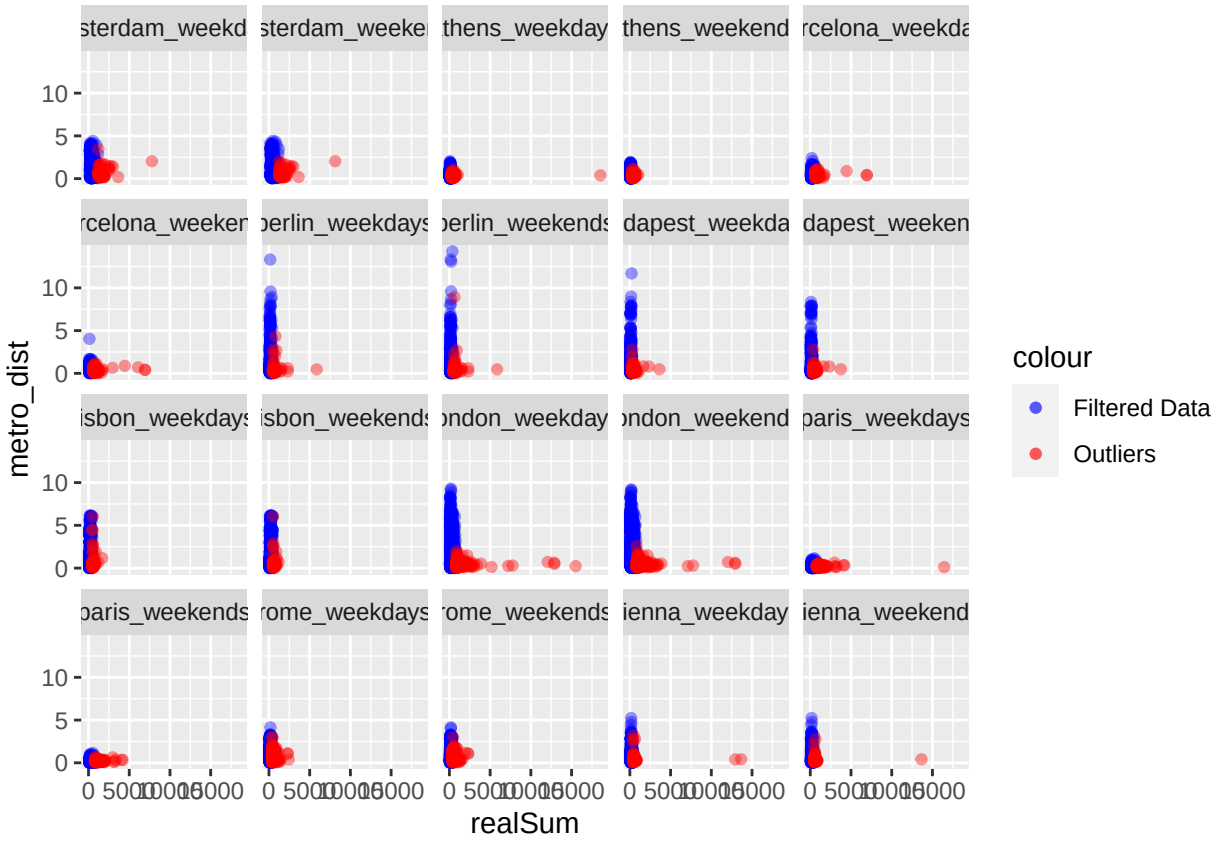

```
ggplot(my_outliers, aes(x = realSum, fill = multi, group = multi)) +
    geom_histogram(alpha = 0.5, bins = 20) + theme(axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14))
```




```
ggplot() + geom_point(data = my_data_filtered, aes(x = realSum,
    y = metro_dist, color = "Filtered Data"), alpha = 0.4) +
    geom_point(data = my_outliers, aes(x = realSum, y = metro_dist,
        color = "Outliers"), alpha = 0.4) + scale_color_manual(values = c(`Filtered Data` = "blue",
    Outliers = "red"))
```

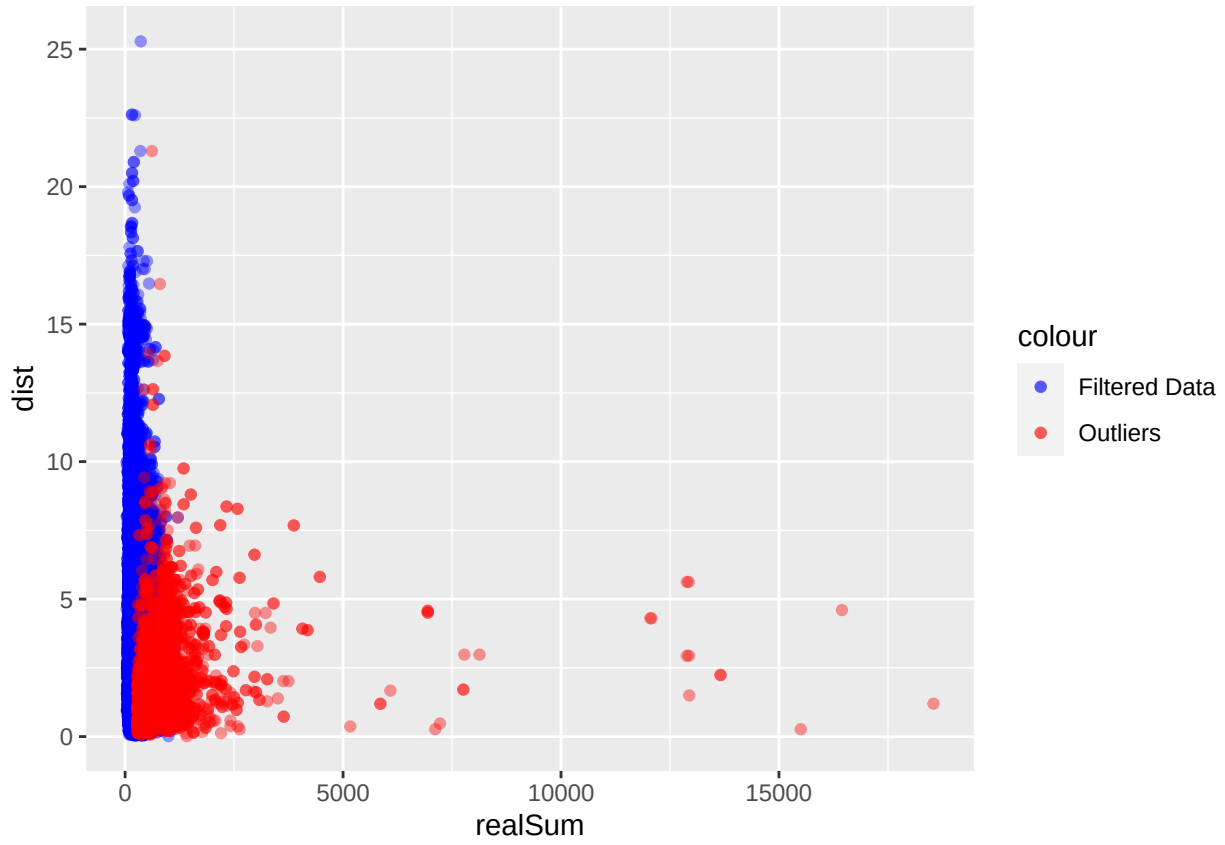



```
ggplot() + geom_point(data = my_data_filtered, aes(x = realSum,
    y = metro_dist, color = "Filtered Data"), alpha = 0.4) +
    geom_point(data = my_outliers, aes(x = realSum, y = metro_dist,
        color = "Outliers"), alpha = 0.4) + scale_color_manual(values = c(`Filtered Data` = "blue",
    Outliers = "red")) + facet_wrap(~city_day)
```

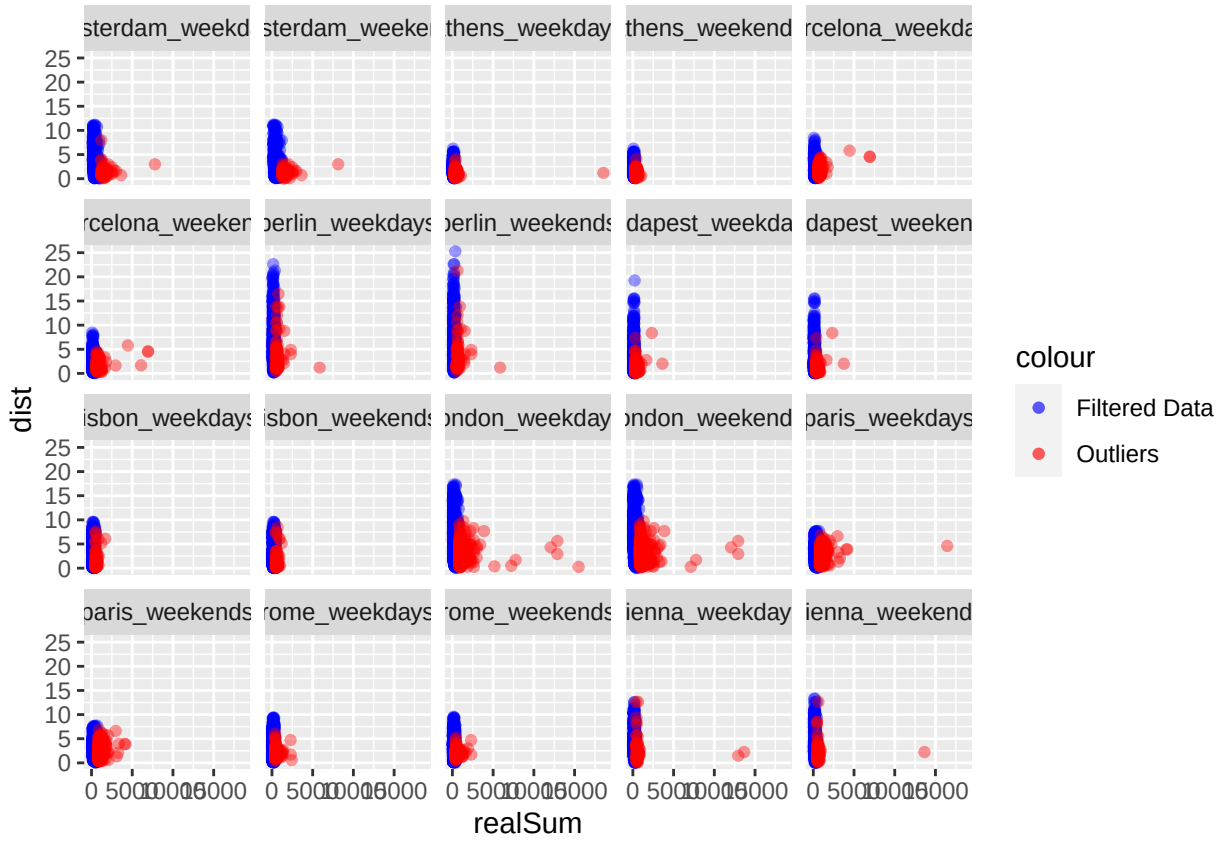


The pricey rooms are mostly near to the metro

```
ggplot() + geom_point(data = my_data_filtered, aes(x = realSum,
    y = dist, color = "Filtered Data"), alpha = 0.4) + geom_point(data = my_outliers,
    aes(x = realSum, y = dist, color = "Outliers"), alpha = 0.4) +
    scale_color_manual(values = c(`Filtered Data` = "blue", Outliers = "red"))
```

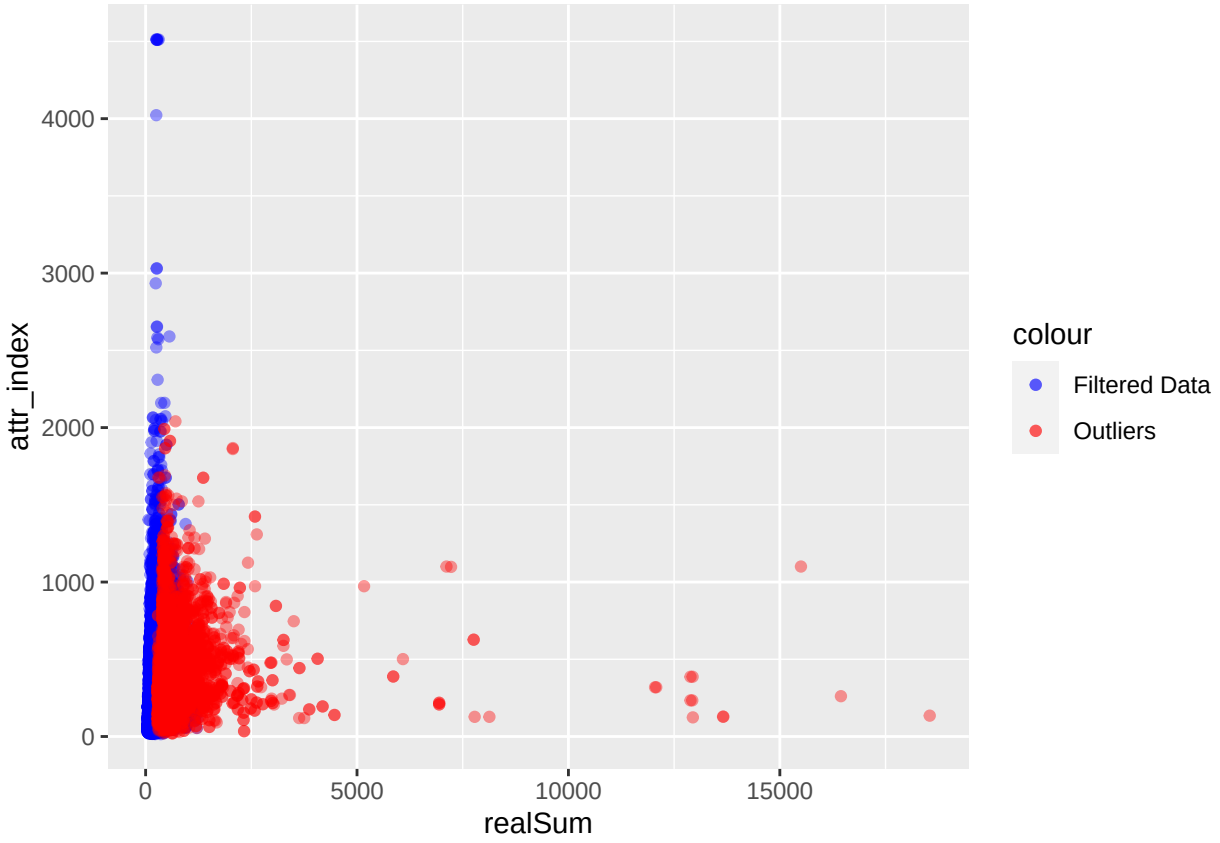



```
ggplot() + geom_point(data = my_data_filtered, aes(x = realSum,
    y = dist, color = "Filtered Data"), alpha = 0.4) + geom_point(data = my_outliers,
    aes(x = realSum, y = dist, color = "Outliers"), alpha = 0.4) +
    scale_color_manual(values = c(`Filtered Data` = "blue", Outliers = "red")) +
    facet_wrap(~city_day)
```

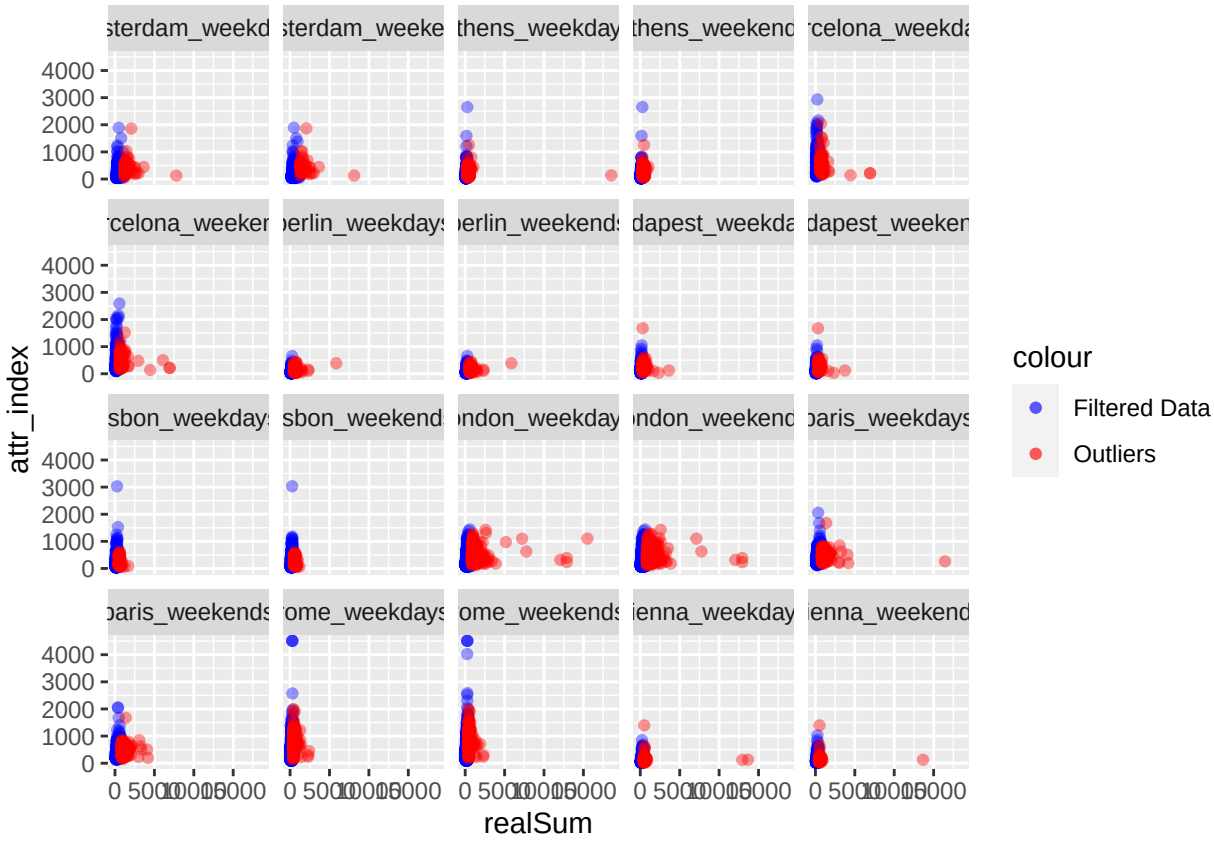


The pricey rooms are mostly near to the centre of the city and the there is some correlation.

```
ggplot() + geom_point(data = my_data_filtered, aes(x = realSum,
    y = attr_index, color = "Filtered Data"), alpha = 0.4) +
    geom_point(data = my_outliers, aes(x = realSum, y = attr_index,
        color = "Outliers"), alpha = 0.4) + scale_color_manual(values = c(`Filtered Data` = "blue",
    Outliers = "red"))
```

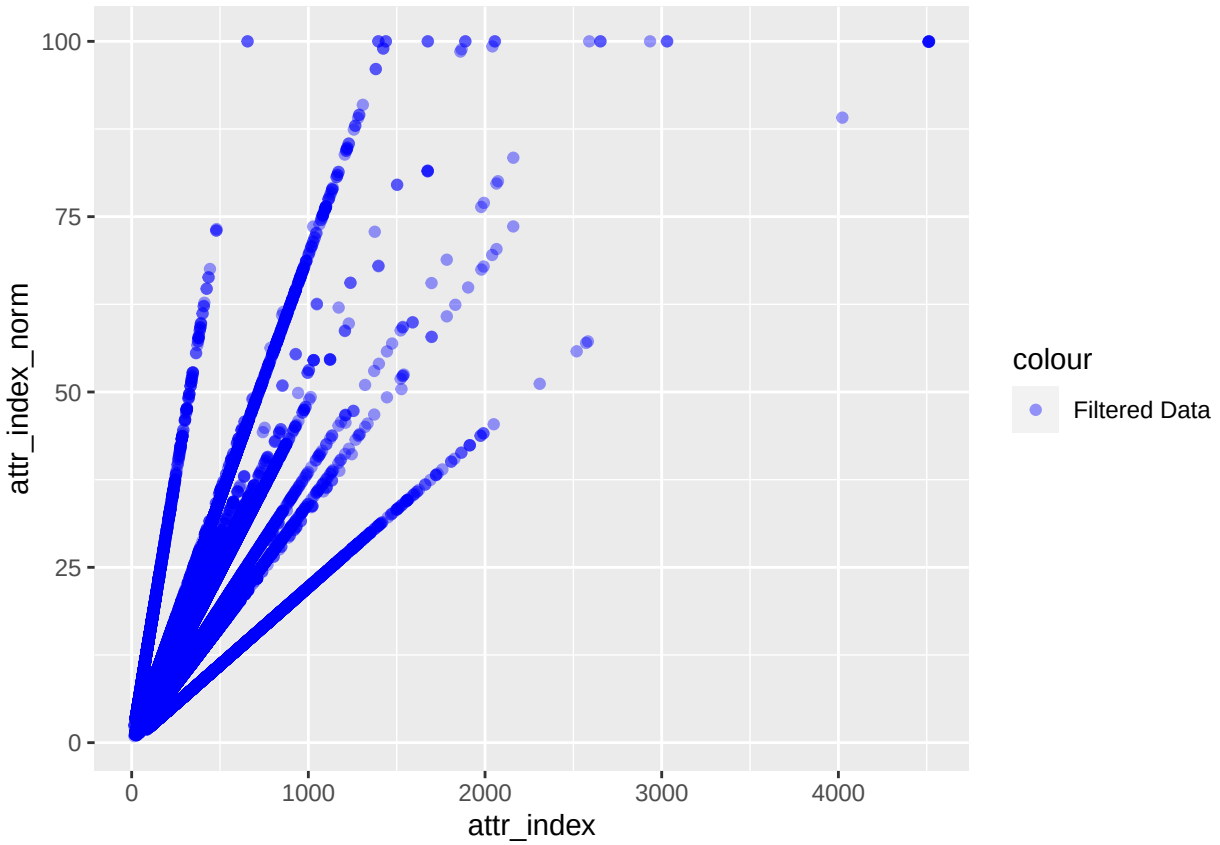



```
ggplot() + geom_point(data = my_data_filtered, aes(x = realSum,
    y = attr_index, color = "Filtered Data"), alpha = 0.4) +
    geom_point(data = my_outliers, aes(x = realSum, y = attr_index,
        color = "Outliers"), alpha = 0.4) + scale_color_manual(values = c(`Filtered Data` = "blue",
    Outliers = "red")) + facet_wrap(~city_day)
```

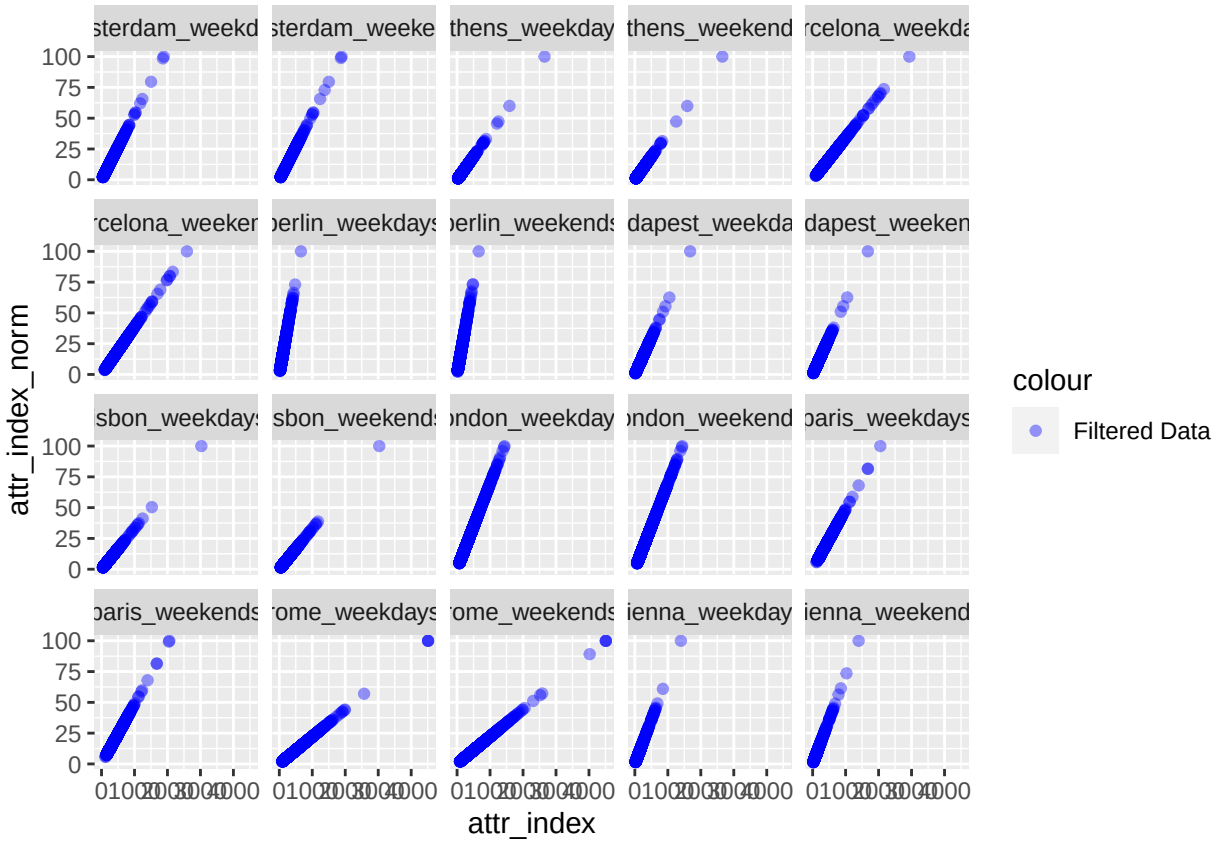


The range of values falling b/w outliers and normal data is almost same . So there isn't a relationship b/w attr_index and realSum.

```
ggplot() + geom_point(data = my_data, aes(x = attr_index, y = attr_index_norm,
    color = "Filtered Data"), alpha = 0.4) + scale_color_manual(values = c(`Filtered Data` = "blue",
    Outliers = "red"))
```

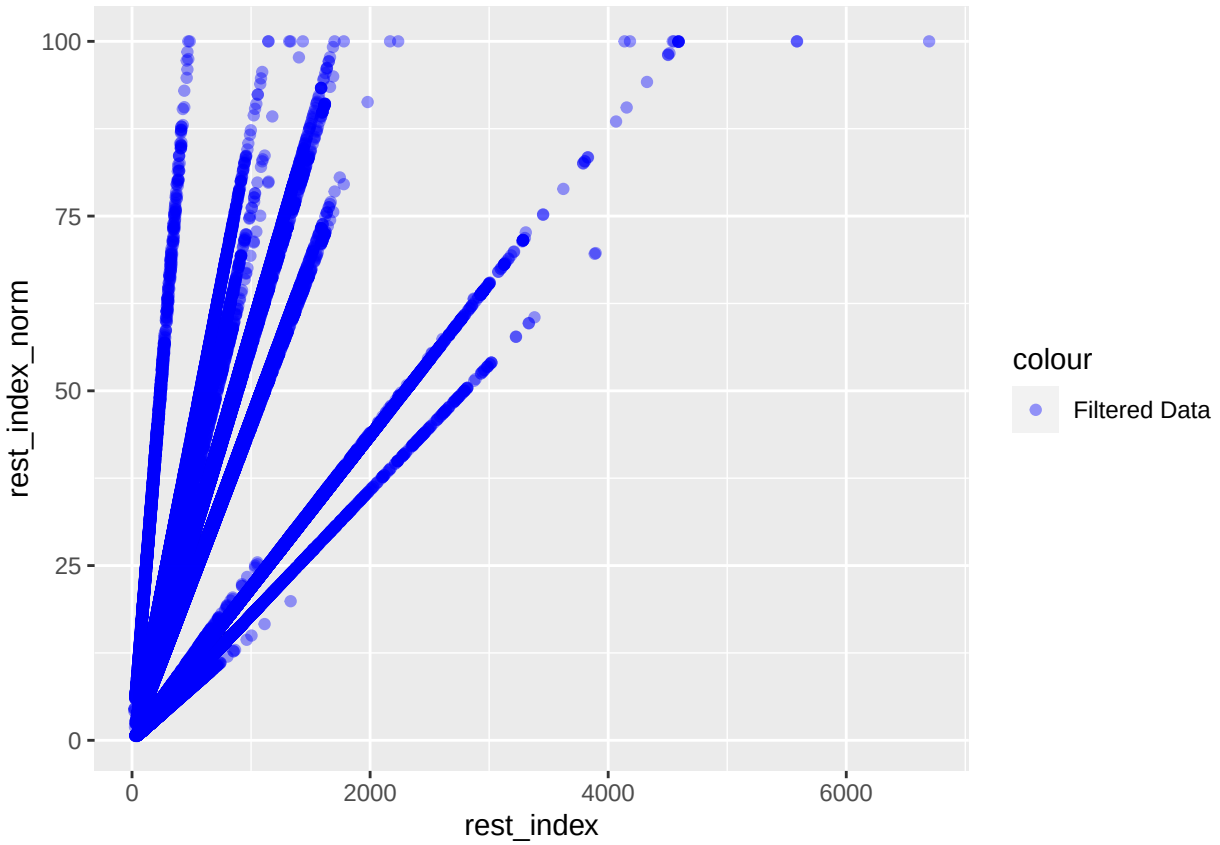



```
ggplot() + geom_point(data = my_data, aes(x = attr_index, y = attr_index_norm,
    color = "Filtered Data"), alpha = 0.4) + scale_color_manual(values = c(`Filtered Data` = "blue",
    Outliers = "red")) + facet_wrap(~city_day)
```

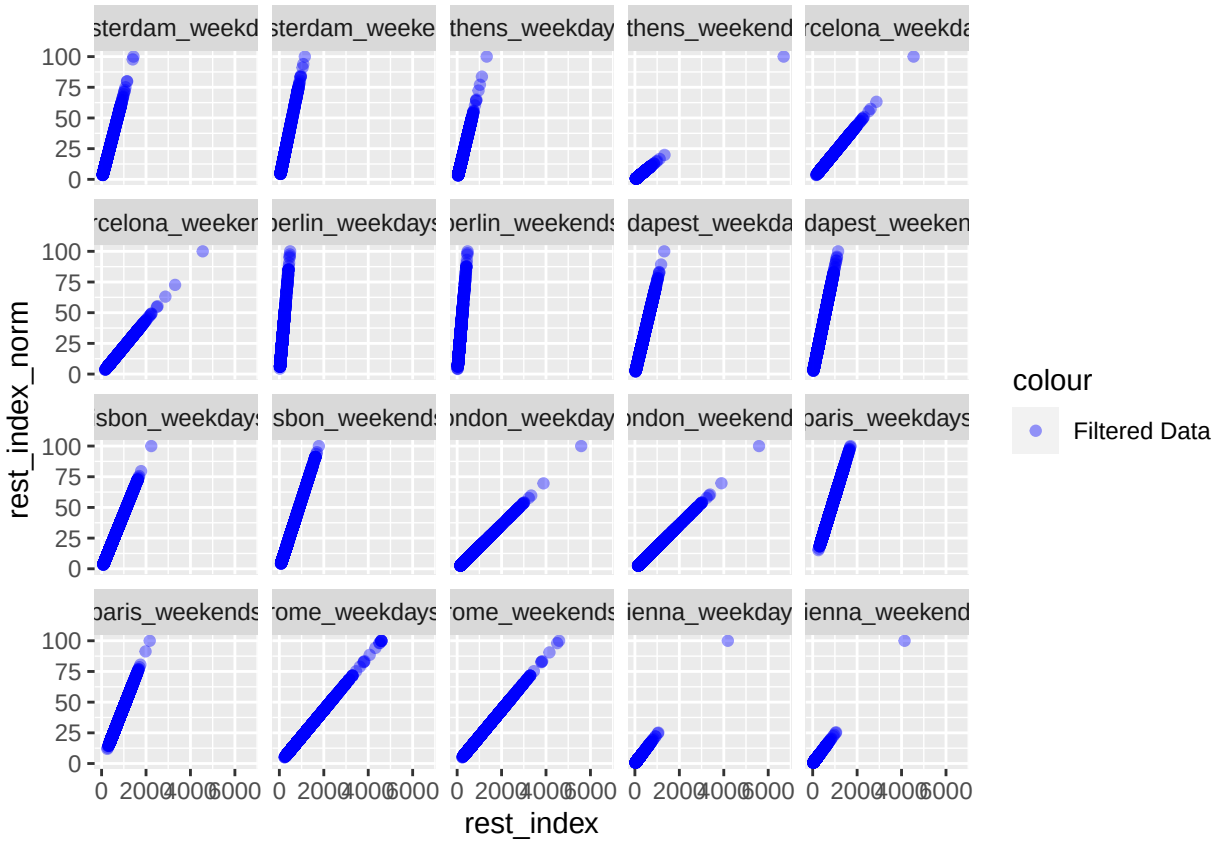


attr_index and attr_index_norm are same, attr_index_norm is just normalized attr_index

```
ggplot() + geom_point(data = my_data, aes(x = rest_index, y = rest_index_norm,
    color = "Filtered Data"), alpha = 0.4) + scale_color_manual(values = c(`Filtered Data` = "blue",
    Outliers = "red"))
```

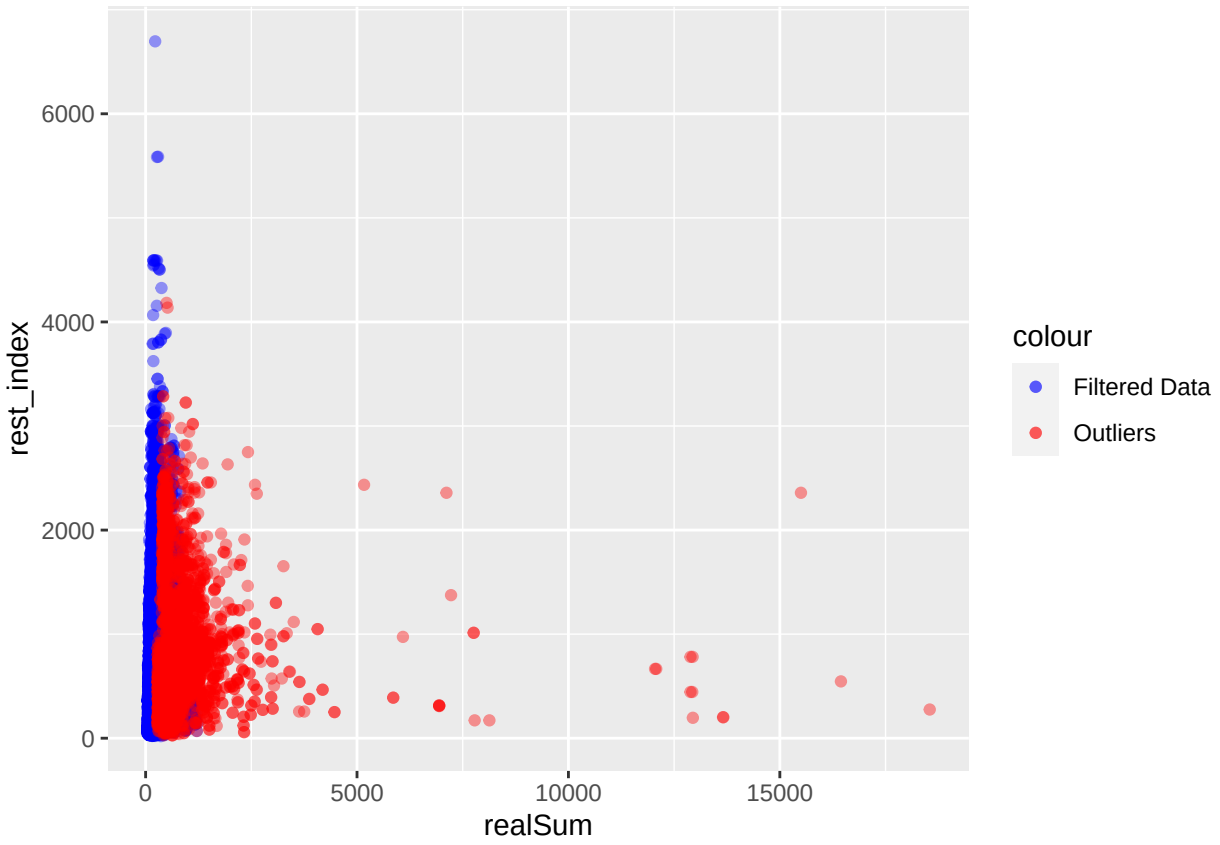



```
ggplot() + geom_point(data = my_data, aes(x = rest_index, y = rest_index_norm,
    color = "Filtered Data"), alpha = 0.4) + scale_color_manual(values = c(`Filtered Data` = "blue",
    Outliers = "red")) + facet_wrap(~city_day)
```

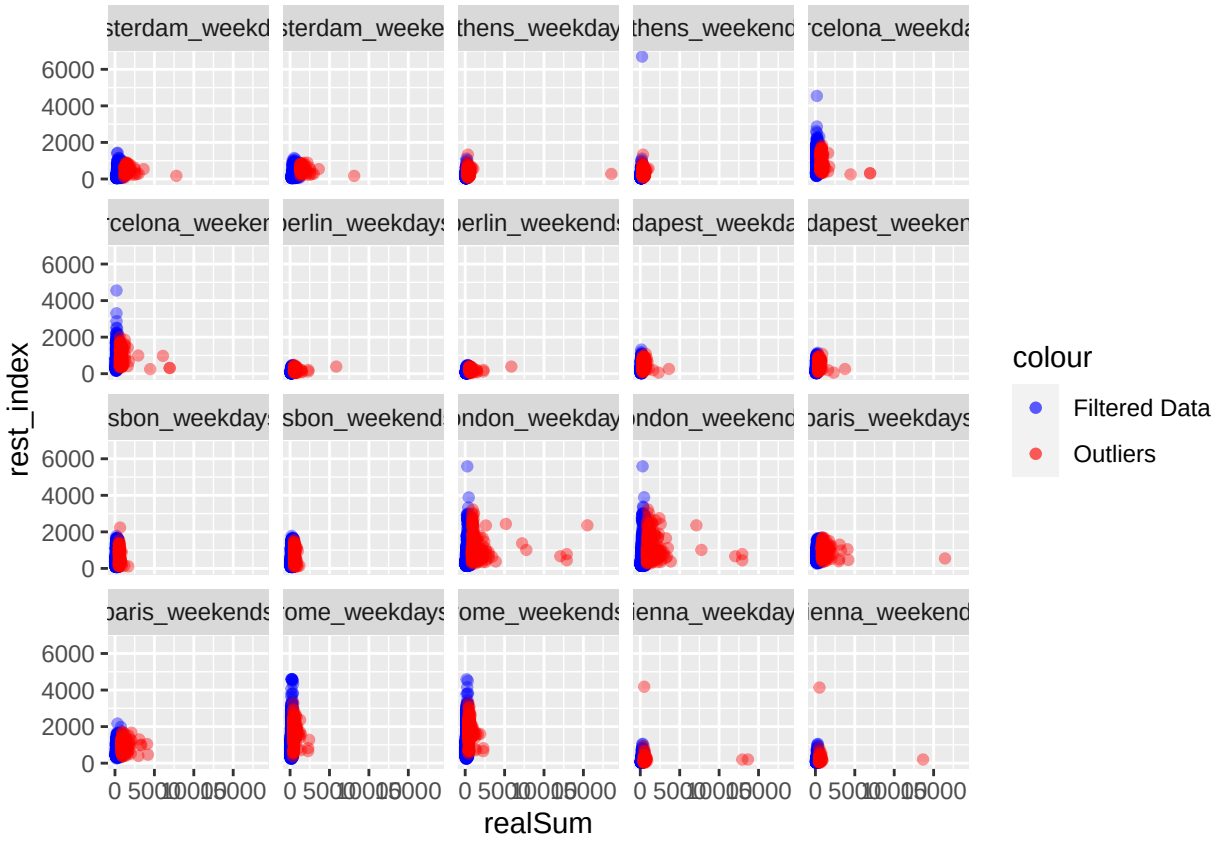


rest_index and rest_index_norm are same, rest_index_norm is just normalized rest_index

```
ggplot() + geom_point(data = my_data_filtered, aes(x = realSum,
    y = rest_index, color = "Filtered Data"), alpha = 0.4) +
    geom_point(data = my_outliers, aes(x = realSum, y = rest_index,
        color = "Outliers"), alpha = 0.4) + scale_color_manual(values = c(`Filtered Data` = "blue",
    Outliers = "red"))
```




```
ggplot() + geom_point(data = my_data_filtered, aes(x = realSum,
    y = rest_index, color = "Filtered Data"), alpha = 0.4) +
    geom_point(data = my_outliers, aes(x = realSum, y = rest_index,
        color = "Outliers"), alpha = 0.4) + scale_color_manual(values = c(`Filtered Data` = "blue",
    Outliers = "red")) + facet_wrap(~city_day)
```



There is no relationship outliers and rest_index

Percentage of Outliers.

```
# Create empty table
outliers_table <- data.frame(City_day = character(), Data_Length = numeric(),
    Percent_Outliers = numeric(), stringsAsFactors = FALSE)

# Loop through city_data and fill in table
for (city_day in unique(my_data$city_day)) {
    x = my_data[my_data$city_day == city_day, ]$realSum
    q1 <- quantile(x, 0.25)
    q3 <- quantile(x, 0.75)
    iqr <- IQR(x)
    upper_bound <- q3 + 1.5 * iqr
    lower_bound <- q1 - 1.5 * iqr
    x_no_outliers <- x[x >= lower_bound & x <= upper_bound]
    percent_outliers <- ((length(x) - length(x_no_outliers))/length(x)) *
        100

    # Add row to table
    outliers_table <- rbind(outliers_table, data.frame(City_day = city_day,
        Data_Length = length(x), Percent_Outliers = percent_outliers))
}
```

```
# Format table using kable
kable(outliers_table, format = "markdown")
```

City_day	Data_Length	Percent_Outliers
amsterdam_weekdays	1103	5.077063
amsterdam_weekends	977	5.629478
athens_weekdays	2653	5.767056
athens_weekends	2627	5.405405
barcelona_weekdays	1555	7.524116
barcelona_weekends	1278	8.059468
berlin_weekdays	1284	6.308411
berlin_weekends	1200	6.166667
budapest_weekdays	2074	5.930569
budapest_weekends	1948	5.544148
lisbon_weekdays	2857	3.360168
lisbon_weekends	2906	3.475568
london_weekdays	4614	5.353273
london_weekends	5379	5.521472
paris_weekdays	3130	6.134185
paris_weekends	3558	5.368184
rome_weekdays	4492	5.031167
rome_weekends	4535	5.005513
vienna_weekdays	1738	4.257767
vienna_weekends	1799	4.113396

Linear Regression

```
M1 <- lm(realSum ~ . - realSum - city_day, data = my_data_train)
summary(M1)
```

```
##
## Call:
## lm(formula = realSum ~ . - realSum - city_day, data = my_data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -871.8   -91.5   -23.6    45.4 18473.9
##
## Coefficients: (2 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -5.755e+02  2.995e+01 -19.212  < 2e-16 ***
## X                          -7.886e-03  1.478e-03  -5.335 9.61e-08 ***
## room_typePrivate room      -1.102e+02  4.160e+00 -26.505  < 2e-16 ***
## room_typeShared room       -1.998e+02  1.919e+01 -10.408  < 2e-16 ***
## room_sharedTrue                    NA         NA      NA       NA
## room_privateTrue                   NA         NA      NA       NA
## person_capacity             1.974e+01  1.792e+00  11.017  < 2e-16 ***
## host_is_superhostTrue      -2.224e+00  3.995e+00  -0.557 0.577722
## multi                      -7.342e-01  4.180e+00  -0.176 0.860570
## biz                         1.322e+01  4.181e+00   3.161 0.001572 **
```

```
## cleanliness_rating          5.685e+00  2.456e+00   2.315 0.020639 *
## guest_satisfaction_overall  7.461e-01  2.660e-01   2.805 0.005042 **
## bedrooms                    9.316e+01  3.226e+00  28.881  < 2e-16 ***
## dist                       -6.242e+00  1.123e+00  -5.556 2.78e-08 ***
## metro_dist                 -6.913e+00  2.360e+00  -2.929 0.003398 **
## attr_index                  8.638e-02  1.984e-02   4.354 1.34e-05 ***
## attr_index_norm             4.554e+00  3.722e-01  12.235  < 2e-16 ***
## rest_index                  9.112e-04  7.446e-03   0.122 0.902598
## rest_index_norm            -5.321e-01  1.372e-01  -3.877 0.000106 ***
## lng                        -5.587e+00  1.880e-01 -29.724  < 2e-16 ***
## lat                         1.323e+01  5.255e-01  25.170  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 310.6 on 36175 degrees of freedom
## Multiple R-squared:  0.1858, Adjusted R-squared:  0.1854
## F-statistic: 458.7 on 18 and 36175 DF,  p-value: < 2.2e-16
```

Checking for Multi Collinearity

```
alias(M1)
```

```
## Model :
## realSum ~ (X + room_type + room_shared + room_private + person_capacity +
##     host_is_superhost + multi + biz + cleanliness_rating + guest_satisfaction_overall +
##     bedrooms + dist + metro_dist + attr_index + attr_index_norm +
##     rest_index + rest_index_norm + lng + lat + city_day) - realSum -
##     city_day
##
## Complete :
##                  (Intercept) X room_typePrivate room room_typeShared room
## room_sharedTrue  0           0 0                    1
## room_privateTrue 0           0 1                    0
##                  person_capacity host_is_superhostTrue multi biz
## room_sharedTrue  0               0                     0     0
## room_privateTrue 0               0                     0     0
##                  cleanliness_rating guest_satisfaction_overall bedrooms dist
## room_sharedTrue  0                  0                          0        0
## room_privateTrue 0                  0                          0        0
##                  metro_dist attr_index attr_index_norm rest_index
## room_sharedTrue  0          0          0               0
## room_privateTrue 0          0          0               0
##                  rest_index_norm lng lat
## room_sharedTrue  0               0   0
## room_privateTrue 0               0   0
```

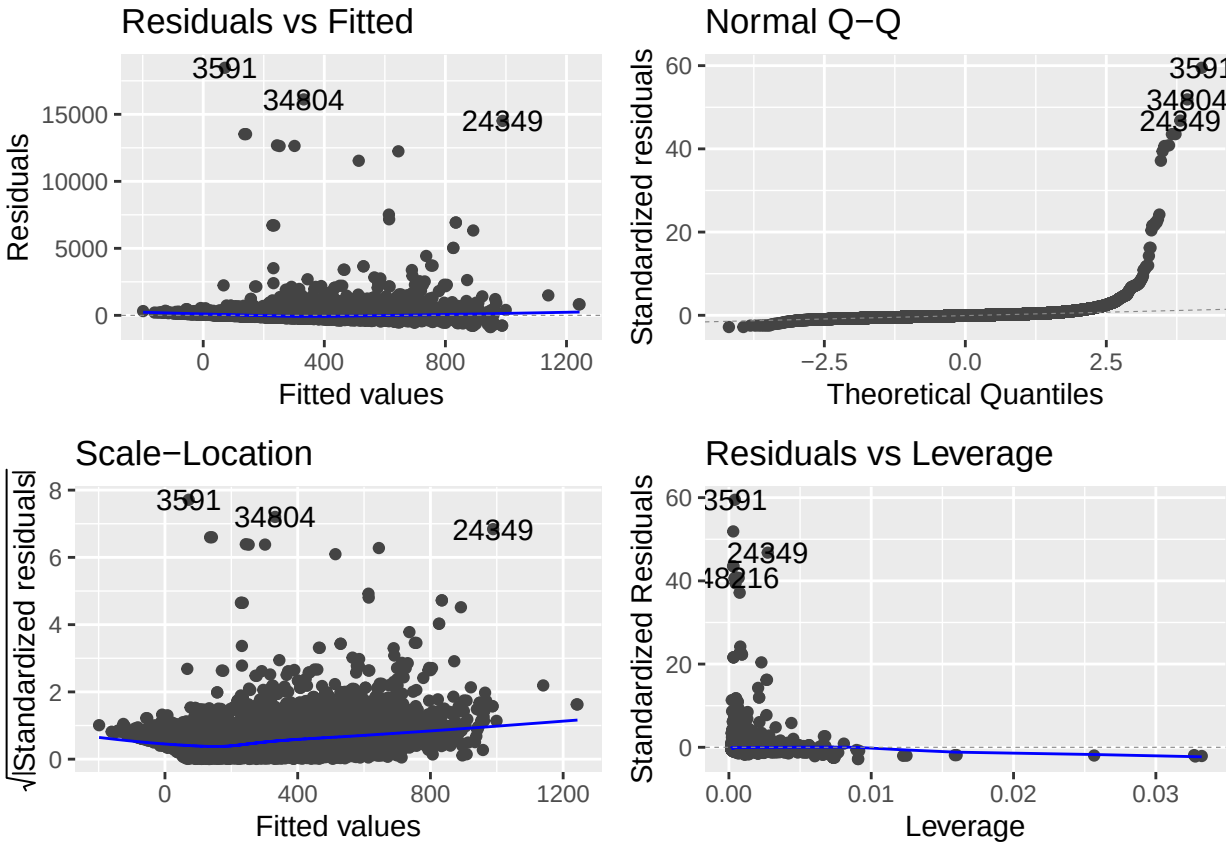
```
# Calculate the VIFs
M2 <- lm(realSum ~ . - realSum - city_day - room_type, data = my_data_train)
vif(M2)
```

```
##                          X                room_shared
```

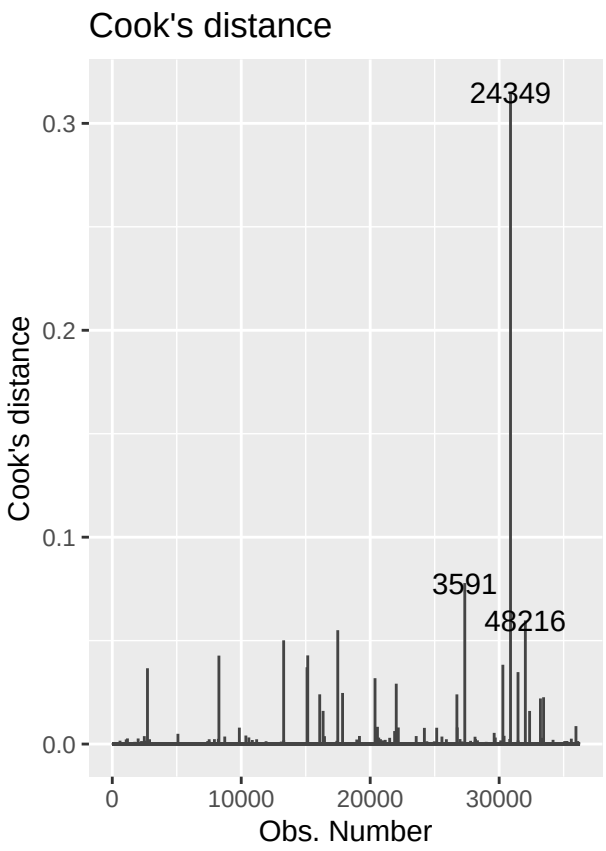
```
##                    1.215469                    1.015685
##                room_private             person_capacity
##                    1.502068                    2.034193
##           host_is_superhost                       multi
##                    1.137854                    1.355783
##                         biz           cleanliness_rating
##                    1.487035                    2.051483
## guest_satisfaction_overall                    bedrooms
##                    2.132586                    1.555173
##                        dist                 metro_dist
##                    2.775926                    1.575344
##                  attr_index             attr_index_norm
##                    7.361432                    4.959117
##                  rest_index             rest_index_norm
##                    5.139643                    2.249142
##                         lng                         lat
##                    1.270706                    2.853201
```

Autoplots

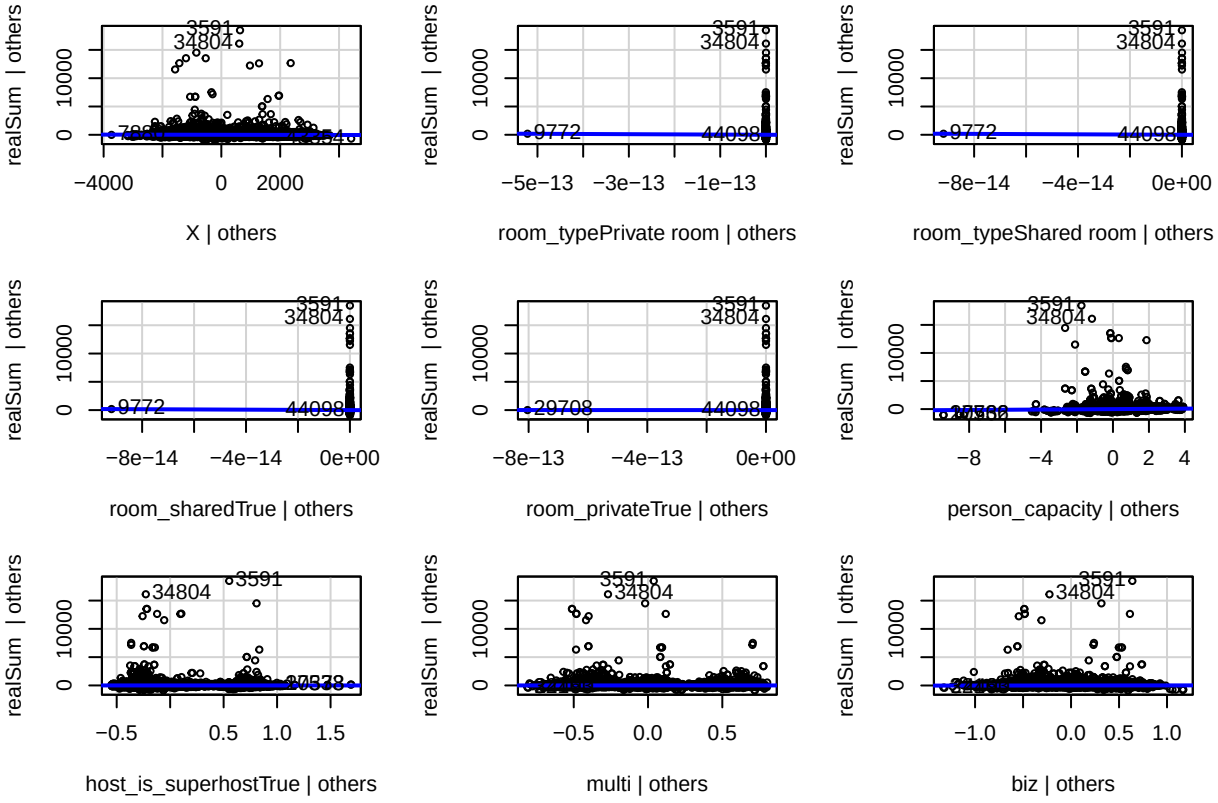
```
autoplot(M1)
```

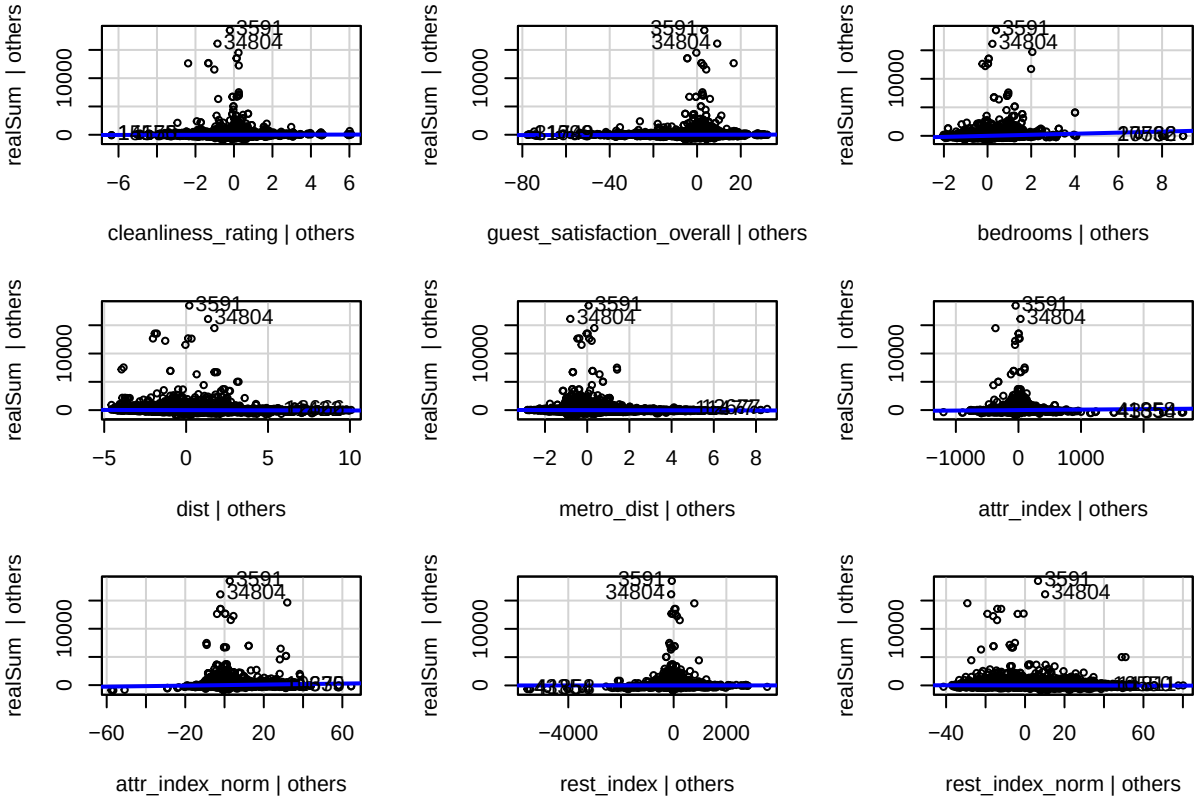


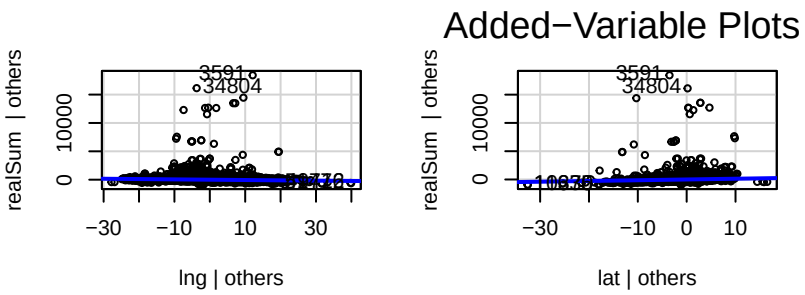
```
autoplot(M1, which = 4)
```




```
avPlots(M1)
```


realSum | others
X | others
room_typePrivate room | others
room_typeShared room | others
room_sharedTrue | others
room_privateTrue | others
person_capacity | others
host_is_superhostTrue | others
multi | others
biz | others
3591
34804
9772
44098
20708


realSum | others
cleanliness_rating | others
guest_satisfaction_overall | others
bedrooms | others
dist | others
metro_dist | others
attr_index | others
attr_index_norm | others
rest_index | others
rest_index_norm | others
3591
34804



Cooks Distance

```
cooksd <- cooks.distance(M1)
max(cooksd)
```

```
## [1] 0.3151076
```

Cooks Distance V2

```
augment_M1 = data.frame(augment(M1))
max(augment_M1$.cooksd)
```

```
## [1] 0.3151076
```

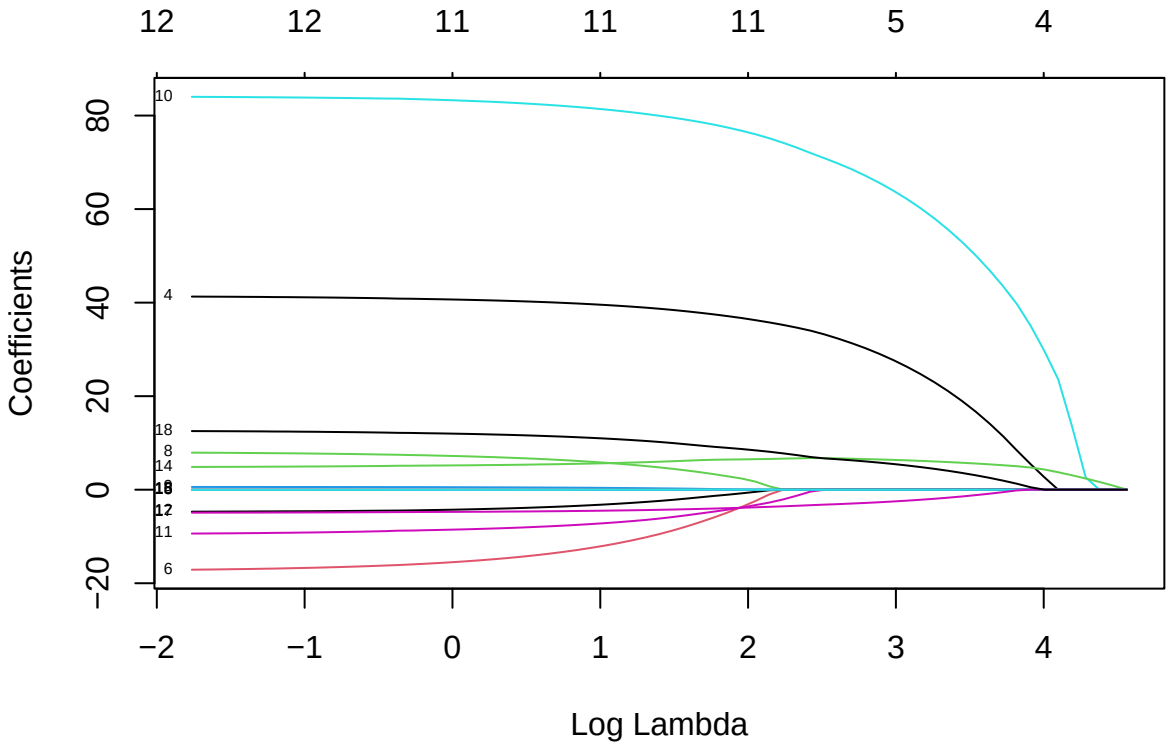
Prediction Analysis

```
my_data_test_spread <- my_data_test %>%
    spread_predictions(M1)
my_data_test_spread <- my_data_test %>%
    spread_residuals(M1)
mean(my_data_test_spread$M1)
```

```
## [1] -0.9869423
```

Lasso Regression

```
x <- as.matrix(my_data[, c("room_type", "room_shared", "room_private",
    "person_capacity", "host_is_superhost", "multi", "biz", "cleanliness_rating",
    "guest_satisfaction_overall", "bedrooms", "dist", "metro_dist",
    "attr_index", "attr_index_norm", "rest_index", "rest_index_norm",
    "lng", "lat", "city_day")])
y <- my_data$realSum
lasso.fit <- glmnet(x, y, alpha = 1)
plot(lasso.fit, xvar = "lambda", label = TRUE)
```



Model Selection

Model Training

Conclusion