Kyoto AirBnB Analysis

Marina Huang

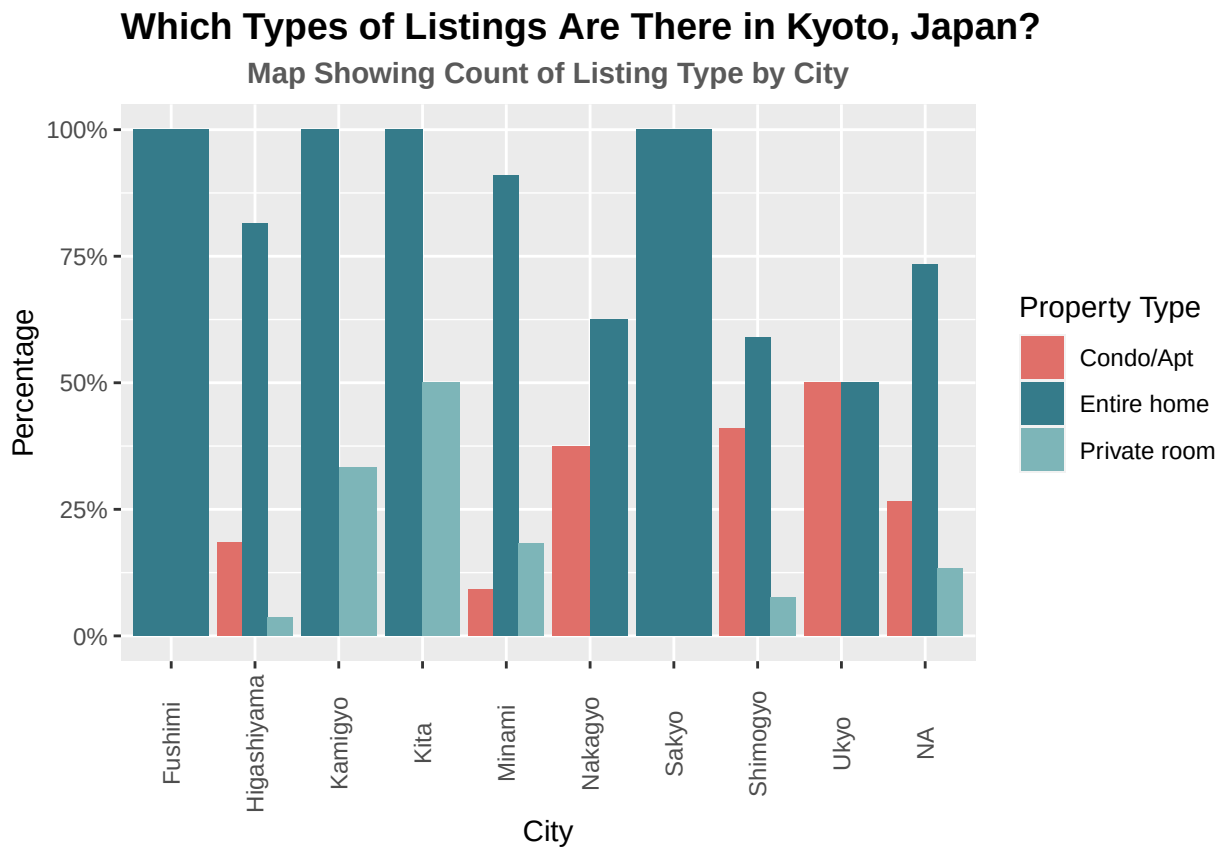
2023-04-20

Exploratory Data Analysis

Which Type of Listings Are There in the Cities?

We do an analysis to find out the type of listings that are common to a particular city.

```
property_df <- airbnb %>%
  group_by(city, property) %>%
  summarize(Freq = n())

total_property <- airbnb %>%
  filter(property %in% c("Entire home", "Condo/Apt", "Private Room")) %>%
  group_by(city) %>%
  summarize(sum = n())

property_ratio <- merge(property_df, total_property, by = "city")

property_ratio <- property_ratio %>%
  mutate(ratio = Freq/sum)

ggplot(property_ratio, aes(x = city, y = ratio, fill = property)) +
  geom_bar(position = "dodge", stat = "identity") +
  xlab("City") + ylab("Count") +
  scale_fill_discrete(name = "Property Type") +
  scale_y_continuous(labels = scales::percent) +
  ggtitle("Which Types of Listings Are There in Kyoto, Japan?",
          subtitle = "Map Showing Count of Listing Type by City") +
          theme(plot.title = element_text(face = "bold", size = 14)) +
          theme(plot.subtitle = element_text(face = "bold",
                                             color = "grey35", hjust = 0.5)) +
          theme(plot.caption = element_text(color = "grey69")) +
          scale_color_gradient(low = "#d3cbcb", high = "#852eaa") +
          scale_fill_manual("Property Type", values = c("#e06f69","#357b8a", "#7db5b8",
                                                        "#59c6f3", "#f6c458")) +
          xlab("City") + ylab("Percentage") +
          theme(axis.text.x = element_text(angle = 90,
                                           vjust = 0.5,
                                           hjust = 0.5))
```



Observations

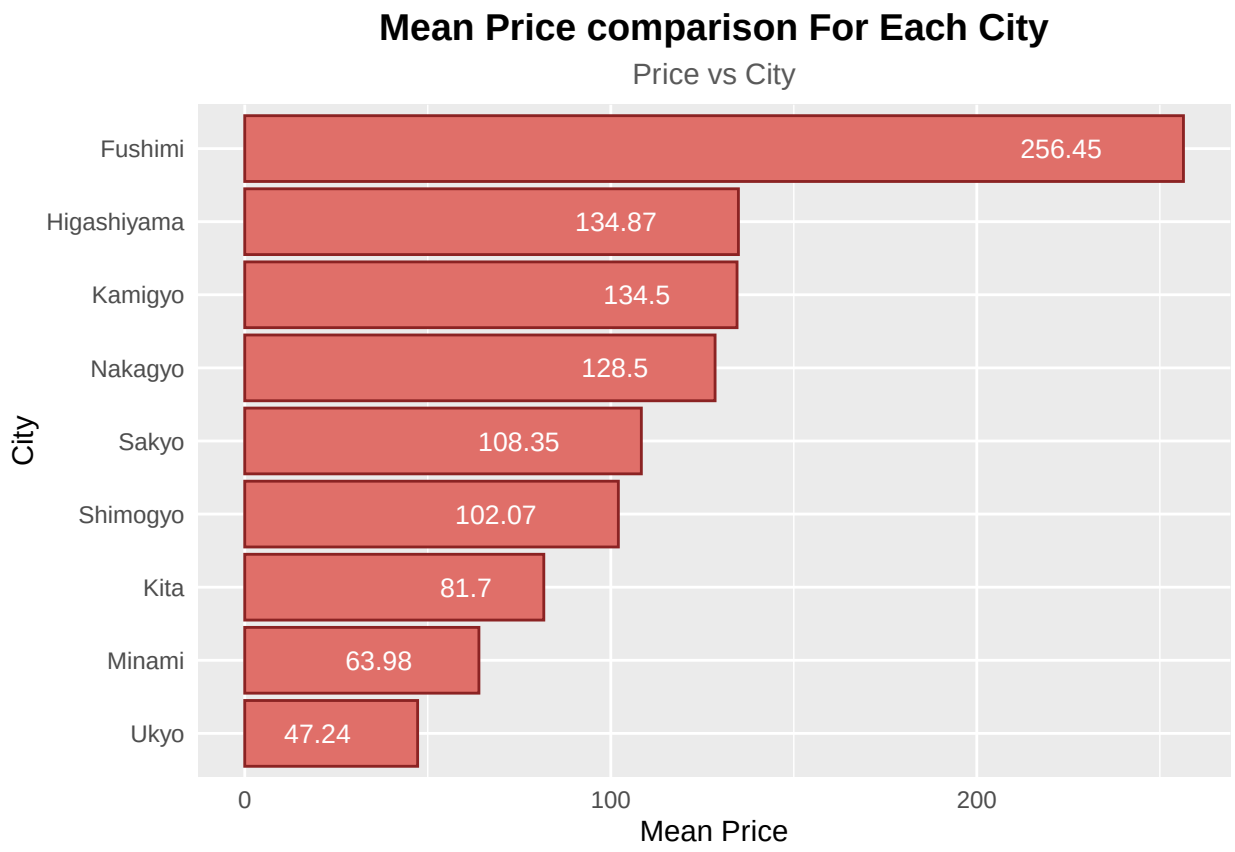
1. Entire homes is the most common listing type in all cities except Ukyo, where Entire homes and Condo/Apt are equally the most common (one of each).
2. There are about equal numbers of Condo/Apt and Private Rooms.

Mean Price Comparison For Each City Group

Obtain the average prices of listings in every city.

```
airbnb %>%
  filter(!(is.na(city))) %>%
  filter(!(city == "Unknown")) %>%
  group_by(city) %>%
  summarize(mean_price = mean(per_night, na.rm = TRUE)) %>%
  ggplot(aes(x = reorder(city, mean_price), y = mean_price, fill = city)) +
  geom_col(stat = "identity", color = "brown4", fill = "#e06f69") +
  coord_flip() +
  theme_gray() +
  labs(x = "City", y = "Price") +
  geom_text(aes(label = round(mean_price, digit = 2)),
            hjust = 2.0, color = "white", size = 3.5) +
  ggtitle("Mean Price comparison For Each City", subtitle = "Price vs City") +
```

```
xlab("City") +
ylab("Mean Price") +
theme(legend.position = "none",
      plot.title = element_text(color = "black",
                                size = 14, face = "bold", hjust = 0.5),
      plot.subtitle = element_text(color = "grey35", hjust = 0.5),
      axis.title.y = element_text(),
      axis.title.x = element_text(),
      axis.ticks = element_blank())
```



Observations

1. Average price of listings is the highest for Fushimi (256.45 USD) followed by Higashiyama (134.87 USD), which is only 0.37 USD more expensive than Kamigyo (134.50 USD).
2. There was only 1 listing in Fushimi, and that should be considered an outlier. If this single listing for Fushimi were to be neglected, Higashiyama and Kamigyo would be the top 2 in average prices.
3. Ukyo has the cheapest listings with an average price of 87.5 USD.

Mean Price Comparison For Each Room Type

Obtain the average prices of listings by accommodation type.

```r
airbnb %>%
  filter(!(is.na(property))) %>%
  filter(!(property == "Unknown")) %>%
  group_by(property) %>%
  summarise(mean_price = mean(per_night, na.rm = TRUE)) %>%
  ggplot(aes(x = reorder(property, mean_price), y = mean_price, fill = property)) +
  geom_col(stat = "identity", color = "brown4", fill = "#e06f69") +
  coord_flip() +
  theme_gray() +
  labs(x = "Accomodation Type", y = "Price") +
  geom_text(aes(label = round(mean_price, digit = 2)),
            hjust = 2.0, color = "white", size = 3.5) +
  ggtitle("Mean Price comparison For Each Accomodation Type",
          subtitle = "Price vs Accomodation Type") +
  xlab("Accomodation Type") +
  ylab("Mean Price") +
  theme(legend.position = "none",
        plot.title = element_text(color = "black",
                                  size = 14, face = "bold", hjust = 0.5),
        plot.subtitle = element_text(color = "grey35", hjust = 0.5),
        axis.title.y = element_text(),
        axis.title.x = element_text(),
        axis.ticks = element_blank())
```

Observation

1. Average price is highest for Entire Homes, followed by Condo/Apt, which is expected since entire homes tend to have larger rooms and multiple stories.

Modelling

Data Splitting

We will split the data into Training set and Testing sets in the ratio of 70:30 so that we can use the testing set to validate our model. Training set will be 70% of the original data. We will use the test data set in the future for testing and prediction purposes. In order to remove the outliers, we are filtering the `airbnb` data by removing the extreme values of price from both sides (10% from both the ends). They would make the predictive models significantly weaker.

Removing Outliers:

```
airbnb_filtered <- airbnb %>%
  filter(per_night < quantile(airbnb$per_night, 0.9) & per_night >
           quantile(airbnb$per_night, 0.1))

#row 86 had "Kyoto" under "reviews", deleting.
airbnb_filtered[86, "reviews"] <- NA
airbnb_filtered <- airbnb_filtered %>% drop_na(reviews)

#change type for "rating" and "reviews" to numeric
airbnb_filtered$rating <- as.double(airbnb_filtered$rating)
airbnb_filtered$reviews <- as.double(airbnb_filtered$reviews)
```

Creating Train and Test Sets:

```
set.seed(123456)
colnames(airbnb)[1] = "id"
airbnb_filtered <- airbnb_filtered %>% mutate(id = row_number())
airbnb_train <- airbnb_filtered %>% sample_frac(0.7)
airbnb_test <- anti_join(airbnb_filtered, airbnb_train, by = "id")
```

Check if Train Set + Test Set = Total Number of Observations in the Original Data Set:

```
nrow(airbnb_train) + nrow(airbnb_test) == nrow(airbnb_filtered)
```

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

Observations

1. The resulting training data set has 121 observations and testing data set has 52 observations.

Summary of Variables Excluded:

In our model, we won't be considering the below variables and the reasons in the summary below.

- id: Unique Identifier, not relevant to the study
- name: Identifier, not relevant to the study
- type: Redundant, we are already taking property in our study
- badge_desc: It describes the badges on a listing, will unnecessarily complicate our model
- checkin_until: Most listings do not have this variable, and is not necessary. Checkin_from is.

- host: Identifier, not relevant to the study
- url: Unique Identifier, not relevant to the study
- host_response: Redundant, almost all variables are 100%

Hence, we try to predict the **price** of the AirBnBs using the remaining covariates:

- property
- city
- superhost
- guests
- bedrooms
- beds
- baths
- rating
- reviews
- badge
- checkin_from
- checkout
- CO_alarm
- free_cancellation
- per_night
- total_price

Model Building Process: Linear Regression Model

We will build our initial linear model using all the variables that we have selected for the study.

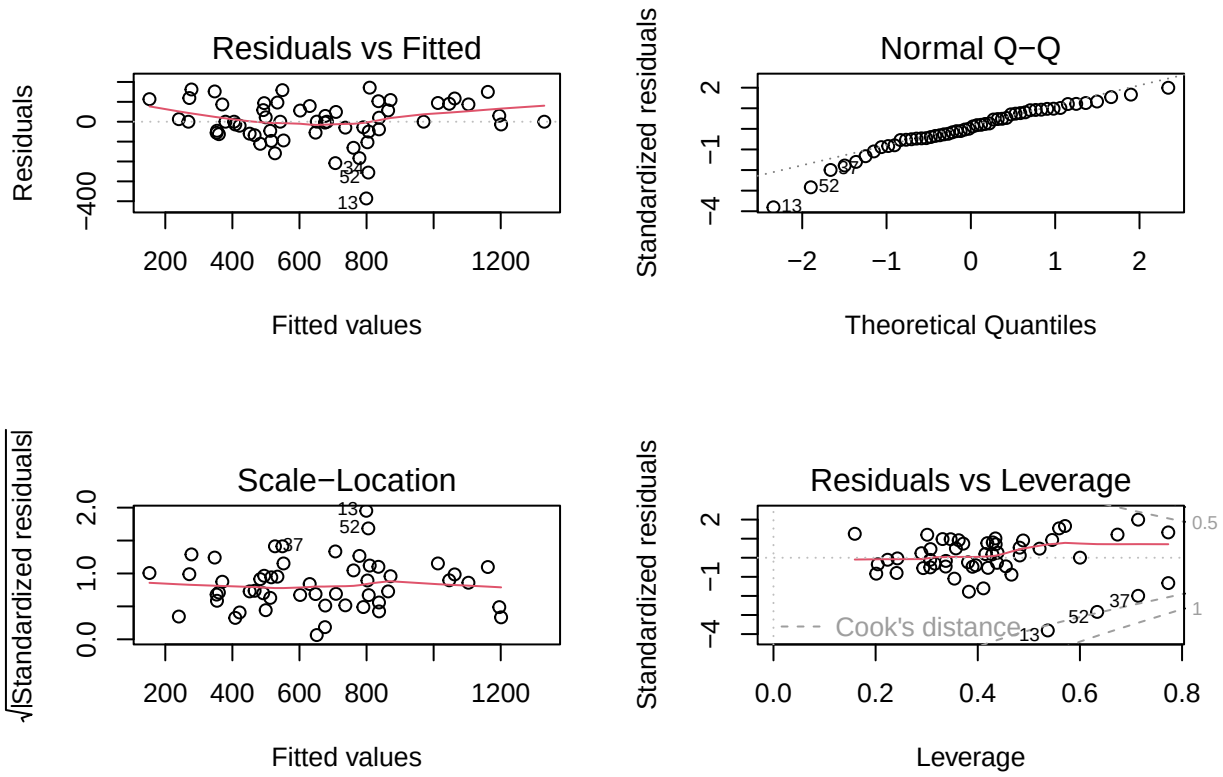
```
airbnb_model_1 <- lm(total_price ~  property + city + superhost +
                       guests + rating + reviews + badge + checkin_from +
                       checkout + CO_alarm + free_cancellation +
                       per_night, data = airbnb_train)

summary_model_1 <- summary(airbnb_model_1)
rmse_1 <- summary_model_1$sigma
r_sq_1 <- summary_model_1$r.squared
adj_r_sq_1 <- summary_model_1$adj.r.squared

# summary_model_1
```

Model 1 Observations:

- RMSE: 148.92
- R-squared: 0.865
- Adjusted R-squared: 0.739

Plot of the Linear Regression Model

```
par(mfrow = c(2,2))
plot(airbnb_model_1)
```



Observations

1. Residuals vs. Fitted values shows that some of the dots "stand out" at the bottom center of the graph, indicating that there are some outliers. Most values are hovering over zero, which does not show a constant variance around X. The equal variance assumption is not satisfied. Otherwise, the graph would "bounce randomly" around the 0 line which suggests that the relationship is linear is reasonable.

2. QQ Plot shows a 45 degree line, meaning that Normality assumptions are met. The bottom part of the line is slightly skewed, so it may be considered that the data set is skewed left.

Conclusion