

R project

Goodness Rex Nze-Igwe
2024-12-11

Data Cleaning and Restructuring

Load the Dataset

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library(ggplot2)
library(tidy)
library(caret)

## Loading required package: lattice

path <- "~/Downloads/AB_NYC_2019.csv"

# Load the dataset
data <- read.csv(path)

# View the structure of the data
str(data)

## 'data.frame':   48895 obs. of  16 variables:
## $ id                : int  2539 2595 3647 3831 5022 5099 5121 5178 5203 5238 ...
## $ name              : chr   "Clean & quiet apt home by the park" "Skylit Midtown Castle" "THE VILL
AGE OF HARLEM....NEW YORK !" "Cozy Entire Floor of Brownstone" ...
## $ host_id          : int  2787 2845 4632 4869 7192 7322 7356 8967 7490 7549 ...
## $ host_name        : chr   "John" "Jennifer" "Elisabeth" "LisaRoxanne" ...
## $ 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 ...
## $ number_of_reviews : int   9 45 0 270 9 74 49 430 118 160 ...
## $ last_review      : chr   "2018-10-19" "2019-05-21" "" "2019-07-05" ...
## $ reviews_per_month : num  0.21 0.38 NA 4.64 0.1 0.59 0.4 3.47 0.99 1.33 ...
## $ calculated_host_listings_count : int  6 2 1 1 1 1 1 1 4 ...
## $ availability_365  : int  365 355 365 194 0 129 0 220 0 188 ...
```

Handle Missing Values

```
# Check for missing values
colSums(is.na(data))

##           id           name
##           0           0
##      host_id      host_name
##           0           0
## neighbourhood_group neighbourhood
##           0           0
##      latitude      longitude
##           0           0
##      room_type      price
##           0           0
## minimum_nights  number_of_reviews
##           0           0
##      last_review  reviews_per_month
##           0          10052
## calculated_host_listings_count  availability_365
##           0           0

# Impute missing values for 'reviews_per_month' with 0
data$reviews_per_month[is.na(data$reviews_per_month)] <- 0

# Drop rows with missing 'name' or 'host_name'
data <- data %>% drop_na(name, host_name)

# Verify no missing values remain
colSums(is.na(data))

##           id           name
##           0           0
##      host_id      host_name
##           0           0
## neighbourhood_group neighbourhood
##           0           0
##      latitude      longitude
##           0           0
##      room_type      price
##           0           0
## minimum_nights  number_of_reviews
##           0           0
##      last_review  reviews_per_month
##           0           0
## calculated_host_listings_count  availability_365
##           0           0
```

Remove Duplicates and Outliers

```
# Remove duplicates
data <- data %>% distinct()

# Remove outliers in the 'price' column
data <- data %>% filter(price > 0 & price < quantile(price, 0.99))

# Summary statistics after cleaning
summary(data)

##           id           name           host_id      host_name
## Min.   : 2539   Length:48392   Min.    : 2438   Length:48392
## 1st Qu.: 9475404 Class :character 1st Qu.: 7820478 Class :character
## Median :19674832 Mode :character  Median : 30808664 Mode :character
## Mean   :19015189              Mean   : 67573396
## 3rd Qu.:29132638              3rd Qu.:107434423
## Max.   :36487245              Max.    :274321313
## neighbourhood_group neighbourhood      latitude      longitude
## Length:48392   Length:48392   Min.    :40.50   Min.    : -74.24
## Class :character Class :character 1st Qu.:40.69   1st Qu.: -73.98
## Mode :character Mode :character  Median :40.72   Median : -73.96
##              Mean :40.73   Mean   : -73.95
##              3rd Qu.:40.76   3rd Qu.: -73.94
##              Max.    :40.91   Max.    : -73.71
##      room_type      price      minimum_nights  number_of_reviews
## Length:48392   Min.    : 10.0   Min.    : 1.000   Min.    : 0.00
## Class :character 1st Qu.: 69.0   1st Qu.: 1.000   1st Qu.: 1.00
## Mode :character  Median :105.0   Median : 3.000   Median : 5.00
##              Mean :137.3   Mean   : 6.981   Mean   : 23.42
##              3rd Qu.:175.0   3rd Qu.: 5.000   3rd Qu.: 24.00
##              Max.    :795.0   Max.    :1250.000   Max.    :629.00
## last_review  reviews_per_month  calculated_host_listings_count
## Length:48392   Min.    : 0.000   Min.    : 1.000
## Class :character 1st Qu.: 0.040   1st Qu.: 1.000
## Mode :character  Median : 0.380   Median : 1.000
##              Mean : 1.097   Mean   : 7.181
##              3rd Qu.: 1.600   3rd Qu.: 2.000
##              Max.    :58.500   Max.    :327.000
## availability_365
## Min.    : 0
## 1st Qu.: 0
## Median : 44
## Mean   :112
## 3rd Qu.:224
## Max.    :365
```

Data Restructuring

Transforming Variables

```
# Create a new column for price category
data <- data %>% mutate(price_category = case_when(
  price < 60 ~ "Low",
  price < 120 ~ "Medium",
  TRUE ~ "High"
))

# View the updated structure
glimpse(data)

## Rows: 48,392
## Columns: 17
## $ id                <int> 2539, 2595, 3647, 3831, 5022, 5099, 512...
## $ name              <chr> "Clean & quiet apt home by the park", "...
## $ host_id          <int> 2787, 2845, 4632, 4869, 7192, 7322, 735...
## $ host_name        <chr> "John", "Jennifer", "Elisabeth", "LisaR...
## $ neighbourhood_group <chr> "Brooklyn", "Manhattan", "Manhattan", "...
## $ neighbourhood    <chr> "Kensington", "Midtown", "Harlem", "Cli...
## $ latitude         <dbl> 40.64749, 40.75362, 40.80902, 40.68514,...
## $ longitude        <dbl> -73.97237, -73.98377, -73.94190, -73.95...
## $ room_type        <chr> "Private room", "Entire home/apt", "Pri...
## $ price            <int> 149, 225, 150, 89, 80, 200, 60, 79, 79,...
## $ minimum_nights   <int> 1, 1, 3, 1, 10, 3, 45, 2, 2, 1, 5, 2, 4,...
## $ number_of_reviews <int> 9, 45, 0, 270, 9, 74, 49, 430, 118, 160,...
## $ last_review      <chr> "2018-10-19", "2019-05-21", "", "2019-0...
## $ reviews_per_month <dbl> 0.21, 0.38, 0.00, 4.64, 0.10, 0.59, 0.4...
## $ calculated_host_listings_count <int> 6, 2, 1, 1, 1, 1, 1, 1, 4, 1, 1, 3, ...
## $ availability_365 <int> 365, 355, 365, 194, 0, 129, 0, 220, 0, ...
## $ price_category   <chr> "High", "High", "High", "Medium", "Medi...
```

Data Visualization

Visualize Price Distribution

```
ggplot(data, aes(x = price)) +
  geom_histogram(binwidth = 60, fill = "green", color = "black") +
  labs(title = "Distribution of Prices", x = "Price", y = "Count") +
  theme_minimal()

## Distribution of Prices
## Count
## Price
```

Room Type by Neighborhood Group

```
# Calculate average price by neighborhood
avg_price <- data %>%
  group_by(neighbourhood_group) %>%
  summarize(avg_price = mean(price))

ggplot(data, aes(x = neighbourhood_group, fill = room_type)) +
  geom_bar(position = "dodge") +
  labs(title = "Room Types by Neighborhood Group", x = "Neighborhood Group", y = "Count", fill = "Room Type") +
  theme_minimal()

## Room Types by Neighborhood Group
## Count
## Neighborhood Group
## Room Type
## Entire home/apt
## Private room
## Shared room
```

Average Price by Neighborhood

```
avg_price <- data %>%
  group_by(neighbourhood_group) %>%
  summarize(avg_price = mean(price)) %>%
  mutate(category = if_else(rank(-avg_price) <= 8, neighbourhood, "Other"))

ggplot(avg_price, aes(x = reorder(category, avg_price), y = avg_price)) +
  geom_bar(stat = "identity", fill = "purple") +
  coord_flip() +
  labs(title = "Average Price by Neighborhood (Top 8+ Other)", x = "Neighborhood", y = "Average Price") +
  theme_minimal()

## Average Price by Neighborhood (Top 8+ Other)
## Neighborhood
## Average Price
```

Prediction Model

Prepare Data for Machine Learning

```
library(caret)

# Select relevant columns and encode categorical variables
data_ml <- data %>% select(price, minimum_nights, number_of_reviews, reviews_per_month, calculated_host_listings_count, availability_365, neighbourhood_group, room_type)
data_ml <- data_ml %>% mutate(
  neighbourhood_group = as.factor(neighbourhood_group),
  room_type = as.factor(room_type)
)

# Split the data into training and testing sets
set.seed(123)
train_index <- createDataPartition(data_ml$price, p = 0.8, list = FALSE)
train_data <- data_ml[train_index, ]
test_data <- data_ml[-train_index, ]
```

Train a Linear Regression Model

```
train_data <- na.omit(train_data) ## making sure there is no missing values in the trained data

# Train the model
model <- train(price ~ ., data = train_data, method = "lm")

# Summary of the model
summary(model)

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -221.27  -45.21  -13.38   19.66   685.37
##
## Coefficients:
## (Intercept)                Estimate Std. Error t value Pr(>|t|)
## minimum_nights             -3.194e-01  2.231e-02  -14.313 < 2e-16 ***
## number_of_reviews           -1.377e-01  1.187e-02  -11.579 < 2e-16 ***
## reviews_per_month          -9.877e-01  3.333e-01  -2.963 0.003049 **
## calculated_host_listings_count 1.124e-01  1.342e-02   8.374 < 2e-16 ***
## availability_365            1.120e-01  3.457e-03  32.385 < 2e-16 ***
## neighbourhood_groupBrooklyn  2.619e+01  2.898e+00   9.036 < 2e-16 ***
## neighbourhood_groupManhattan 6.775e+01  2.902e+00  23.343 < 2e-16 ***
## neighbourhood_groupQueens    1.064e+01  3.071e+00   3.464 0.000533 ***
## neighbourhood_groupStaten Island -9.747e+01  5.558e+00  -111.544 < 2e-16 ***
## room_typePrivate room       -1.224e+02  8.738e-01 -111.544 < 2e-16 ***
## room_typeShared room        -1.224e+02  2.802e+00 -43.696 < 2e-16 ***
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83.02 on 38702 degrees of freedom
## Multiple R-squared:  0.3454, Adjusted R-squared:  0.3453
## F-statistic: 1857 on 11 and 38702 DF, p-value: < 2.2e-16
```

Evaluate the Model

```
# Predict on test data
predictions <- predict(model, newdata = test_data)

# Calculate the Root Mean Squared Error
rmse <- sqrt(mean((test_data$price - predictions)^2))

# Calculate the Mean Absolute Error
mae <- mean(abs(test_data$price - predictions))

# Calculate the Mean Absolute Percentage Error
mape <- mean(abs((test_data$price - predictions) / test_data$price)) * 100

# Print results
list(RMSE = rmse, MAE = mae, MAPE = mape)
```

```
## RMSE
## [1] 85.08069
##
## SMAE
## [1] 53.05916
##
## SMAPE
## [1] 44.27837
```

Conclusion

This analysis focused on cleaning and restructuring the Airbnb_NYC_2019 dataset, visualizing key patterns with ggplot2, and developing a machine learning model to predict prices. Future improvements could involve experimenting with more sophisticated models or incorporating additional features.