STAT-FINAL-PROJECT.R

Bhargav Dasari

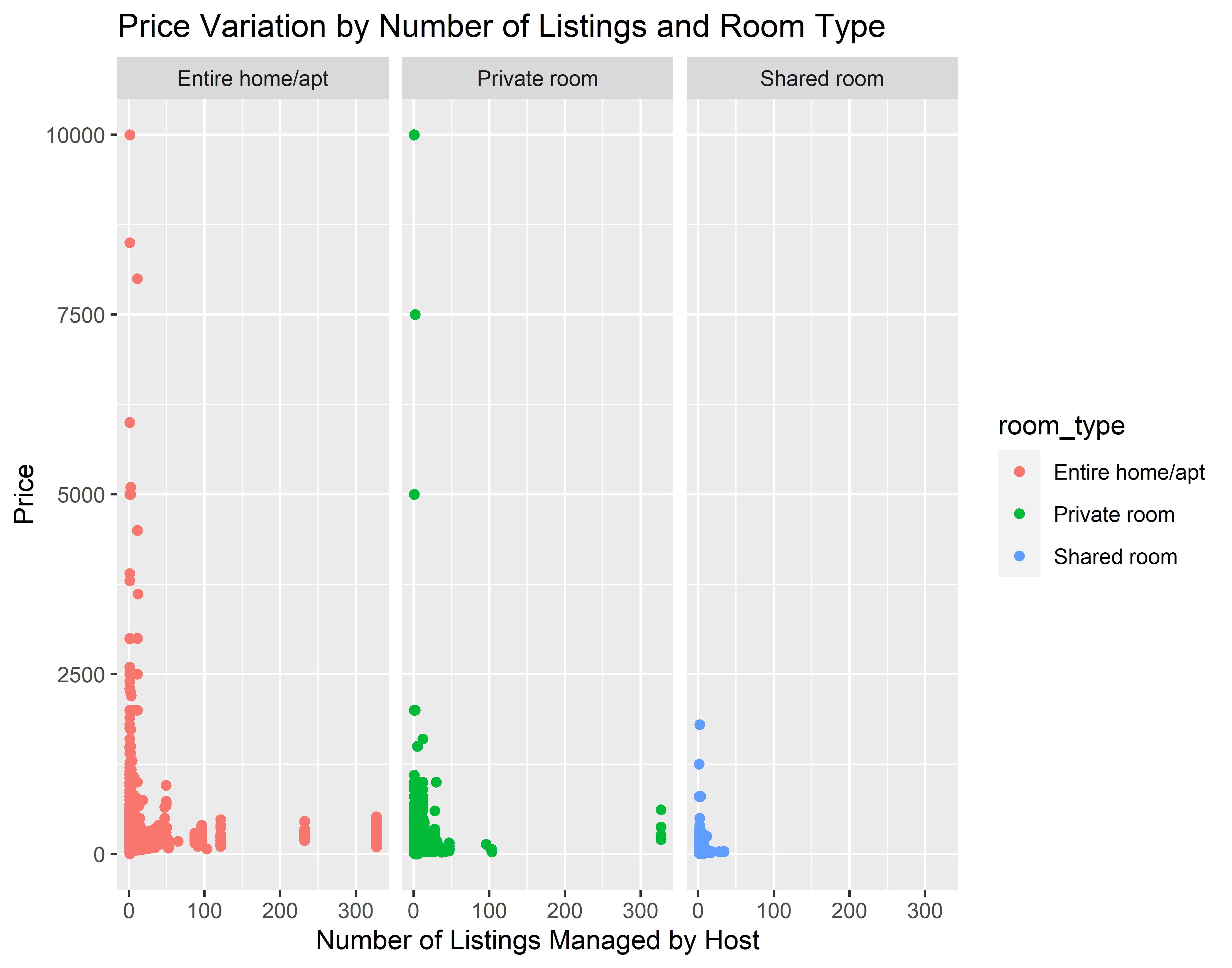
2024-05-01

# Load required libraries  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Read the dataset  
airbnb\_data <- read.csv("C:/Users/Bhargav Dasari/Documents/SPRING-2024/STAT-515/project/cleaned\_data.csv")

# Visualize the relationship between number of listings managed by a host and price variation across different room types  
ggplot(airbnb\_data, aes(x = calculated\_host\_listings\_count, y = price, color = room\_type)) +  
 geom\_point() +  
 facet\_wrap(~ room\_type) +  
 labs(x = "Number of Listings Managed by Host", y = "Price", title = "Price Variation by Number of Listings and Room Type")



# Linear regression to quantify the relationship  
lm\_model <- lm(price ~ calculated\_host\_listings\_count \* room\_type, data = airbnb\_data)  
summary(lm\_model)

##   
## Call:  
## lm(formula = price ~ calculated\_host\_listings\_count \* room\_type,   
## data = airbnb\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -194.9 -47.6 -21.6 11.4 9916.7   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) 194.65262 1.34844  
## calculated\_host\_listings\_count 0.23328 0.03769  
## room\_typePrivate room -111.74346 2.00683  
## room\_typeShared room -122.18015 8.31816  
## calculated\_host\_listings\_count:room\_typePrivate room 0.12059 0.15136  
## calculated\_host\_listings\_count:room\_typeShared room -2.17688 1.05792  
## t value Pr(>|t|)   
## (Intercept) 144.354 < 2e-16 \*\*\*  
## calculated\_host\_listings\_count 6.190 6.09e-10 \*\*\*  
## room\_typePrivate room -55.682 < 2e-16 \*\*\*  
## room\_typeShared room -14.688 < 2e-16 \*\*\*  
## calculated\_host\_listings\_count:room\_typePrivate room 0.797 0.4256   
## calculated\_host\_listings\_count:room\_typeShared room -2.058 0.0396 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 188.5 on 38837 degrees of freedom  
## Multiple R-squared: 0.08386, Adjusted R-squared: 0.08374   
## F-statistic: 711 on 5 and 38837 DF, p-value: < 2.2e-16

# Build Poisson regression model  
poisson\_model <- glm(number\_of\_reviews ~ calculated\_host\_listings\_count + room\_type, data = airbnb\_data, family = poisson)  
  
# Summarize the model  
summary(poisson\_model)

##   
## Call:  
## glm(formula = number\_of\_reviews ~ calculated\_host\_listings\_count +   
## room\_type, family = poisson, data = airbnb\_data)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.3949421 0.0013482 2518.07 <2e-16 \*\*\*  
## calculated\_host\_listings\_count -0.0111227 0.0001199 -92.80 <2e-16 \*\*\*  
## room\_typePrivate room 0.0522275 0.0018922 27.60 <2e-16 \*\*\*  
## room\_typeShared room -0.2190670 0.0073282 -29.89 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 1954366 on 38842 degrees of freedom  
## Residual deviance: 1932166 on 38839 degrees of freedom  
## AIC: 2094724  
##   
## Number of Fisher Scoring iterations: 6

# Preprocess the data  
# Encode categorical variables  
airbnb\_data <- airbnb\_data %>%  
 mutate(location\_encoded = as.factor(neighbourhood\_group),  
 room\_type\_encoded = as.factor(room\_type))  
  
# Handle missing values if any  
  
# Split the data into train and test sets  
set.seed(123)  
train\_indices <- sample(1:nrow(airbnb\_data), 0.8 \* nrow(airbnb\_data))  
train\_data <- airbnb\_data[train\_indices, ]  
test\_data <- airbnb\_data[-train\_indices, ]  
  
# Train regression model  
regression\_model <- lm(number\_of\_reviews ~ location\_encoded + room\_type\_encoded + price, data = train\_data)  
summary(regression\_model)

##   
## Call:  
## lm(formula = number\_of\_reviews ~ location\_encoded + room\_type\_encoded +   
## price, data = train\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -34.20 -26.08 -19.54 4.12 594.18   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 33.460891 1.880139 17.797 < 2e-16 \*\*\*  
## location\_encodedBrooklyn -2.777958 1.880566 -1.477 0.140   
## location\_encodedManhattan -4.319758 1.885814 -2.291 0.022 \*   
## location\_encodedQueens 1.401967 1.998881 0.701 0.483   
## location\_encodedStaten Island 1.649350 3.559207 0.463 0.643   
## room\_type\_encodedPrivate room 0.325691 0.587427 0.554 0.579   
## room\_type\_encodedShared room -7.967898 1.895489 -4.204 2.63e-05 \*\*\*  
## price -0.007853 0.001556 -5.046 4.54e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 48.44 on 31066 degrees of freedom  
## Multiple R-squared: 0.003292, Adjusted R-squared: 0.003067   
## F-statistic: 14.66 on 7 and 31066 DF, p-value: < 2.2e-16

# Predictions on test set  
predictions <- predict(regression\_model, newdata = test\_data)  
  
# Evaluate model performance  
RMSE <- sqrt(mean((test\_data$number\_of\_reviews - predictions)^2))  
MAE <- mean(abs(test\_data$number\_of\_reviews - predictions))  
  
RMSE

## [1] 46.71403

MAE

## [1] 30.57147

# Load required library  
library(randomForest)

## Warning: package 'randomForest' was built under R version 4.3.3

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

# Encode categorical variables  
airbnb\_data$location\_encoded <- as.factor(airbnb\_data$neighbourhood\_group)  
airbnb\_data$room\_type\_encoded <- as.factor(airbnb\_data$room\_type)  
  
# Handle missing values if any  
  
# Split the data into train and test sets  
set.seed(123)  
train\_indices <- sample(1:nrow(airbnb\_data), 0.8 \* nrow(airbnb\_data))  
train\_data <- airbnb\_data[train\_indices, ]  
test\_data <- airbnb\_data[-train\_indices, ]  
  
# Train Random Forest model  
rf\_model <- randomForest(number\_of\_reviews ~ location\_encoded + room\_type\_encoded + price, data = train\_data, ntree = 500)  
print(rf\_model)

##   
## Call:  
## randomForest(formula = number\_of\_reviews ~ location\_encoded + room\_type\_encoded + price, data = train\_data, ntree = 500)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 1  
##   
## Mean of squared residuals: 2314.65  
## % Var explained: 1.67

# Predictions on test set  
rf\_predictions <- predict(rf\_model, newdata = test\_data)  
  
# Evaluate model performance  
rf\_rmse <- sqrt(mean((test\_data$number\_of\_reviews - rf\_predictions)^2))  
rf\_mae <- mean(abs(test\_data$number\_of\_reviews - rf\_predictions))  
  
print(paste("Random Forest RMSE:", rf\_rmse))

## [1] "Random Forest RMSE: 46.5043157596358"

print(paste("Random Forest MAE:", rf\_mae))

## [1] "Random Forest MAE: 30.3279656644863"

# Feature Importance  
varImpPlot(rf\_model)

