Regression tree model to analyze the price trend of Airbnb

Ziqing Cheng & Ruiying Liu

2023-06-08

Table of Contents

[SUMMARY 1](#_Toc137137581)

[INTRODUCTION 2](#_Toc137137582)

[Data source and variable description 2](#_Toc137137583)

[Summary Statistics 3](#_Toc137137584)

[DATA ANALYSIS AND METHOD 4](#_Toc137137585)

[Normality of the data 4](#_Toc137137586)

[Method & model 5](#_Toc137137587)

[Regression tree 5](#_Toc137137588)

[Tree pruning 6](#_Toc137137589)

[Paris\_weekdays 6](#_Toc137137590)

[Paris\_weekends 8](#_Toc137137591)

[London\_weekdays 9](#_Toc137137592)

[London\_weekends 10](#_Toc137137593)

[Conclusion 10](#_Toc137137594)

[Citation 12](#_Toc137137595)

[APPENDIX: The Code 12](#_Toc137137596)

# SUMMARY

This analysis focused on rental variations across regions, the primary factors impacting rental rates, and the evolution of Airbnb rentals over time. The study found that geographical factors play a role in rental prices, with key factors differing between Paris and London. Factors such as person capacity, attraction index, and bedrooms influenced rental prices in Paris, while room type, bedrooms, attraction index, restaurant index, and distance were significant in London. Rental prices on weekdays and weekends showed similar trends in Paris, with higher prices on weekends. In London, weekends were influenced by leisure activities and location. The findings are valuable for real estate investors and Airbnb operators in making informed decisions.

# INTRODUCTION

## Data source and variable description

This study utilized data obtained from the following source: <https://www.kaggle.com/datasets/thedevastator/airbnb-prices-in-european-cities>. The data collection process involved gathering Airbnb rental rates in several popular European cities, including Amsterdam, Athens, Barcelona, Berlin, Budapest, Lisbon, London, Paris, Rome, Vienna. The rental rates were specifically recorded for a two-night stay accommodating two individuals. The data collection period occurred between four to six weeks prior to the intended travel dates. The prices collected encompassed the total amount payable for the accomodation, which inclued teh reservation fee and cleaning fee. Two distict datasets were prepared for each city, including offers for weekdays (Tuesday-Thursday) and weekends (Friday-Sunday) (Gyodi K., & Nawaro L., 2021). For our analysis, we have chosen the datasets for Paris and London as the focus of our research questions.

This dataset encompasses a range of attributes that pertain to rental rates, including factors such as room types, cleanliness and satisfaction ratings, bedrooms, distance from the city center, and additional variables. The dataset comprises a total of 20 variables, and the complete information regarding these variables is as follows:

*X*: index

*realSum*: the full price of accommodation for two people and two nights in EUR (Numeric) *room\_type*: the type of the accommodation (private/shared/entire home/apt) (Categorical)

*room\_shared*: dummy variable for shared rooms (Categorical)

*room\_private*: dummy variable for private rooms (Categorical)

*person\_capacity*: the maximum number of guests (Numeric)

*host\_is\_superhost*: dummy variable for superhost status (Categorical)

*multi*: dummy variable if the listing belongs to hosts with 2-4 offers (Categorical)

*biz*: dummy variable if the listing belongs to hosts with more than 4 offers (Categorical) *cleanliness\_rating*: cleanliness rating (Numeric)

*guest\_satisfaction\_overall*: overall rating of the listing (Numeric)

*bedrooms*: number of bedrooms (0 for studios) (Numeric)

*dist*: distance from city centre in km (Numeric)

*metro\_dist*: distance from nearest metro station in km (Numeric)

*attr\_index*: attraction index of the listing location (Numeric)

*attr\_index\_norm*: normalised attraction index (0-100) (Numeric)

*rest\_index*: restaurant index of the listing location (Numeric)

*attr\_index\_norm*: normalised restaurant index (0-100) (Numeric)

*lng*: longitude of the listing location (Numeric)

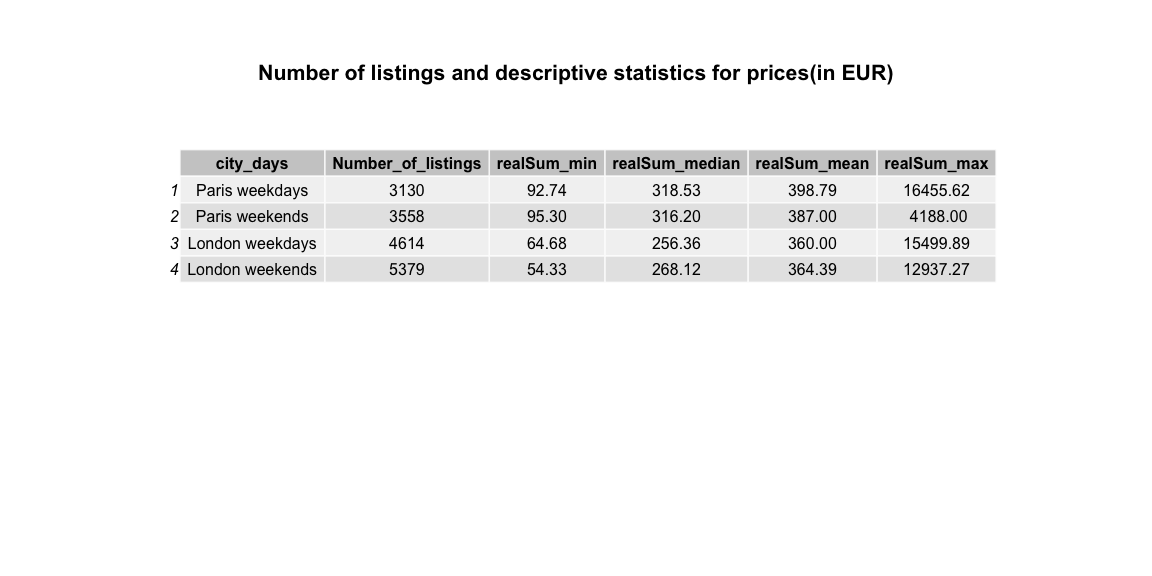
*lat*: latitude of the listing location (Numeric)

Among these, *attr\_index* and *rest\_index* are two numeric variables to evaluate potential popularity of the place Airbnb listings located. Following the work of Yang et al., (2018), *attr\_index* and *rest\_index* are measured based on the locations and numbers of reviews of attractions (e.g., sights, museums, and parks) and restaurants collected from TripAdvisor website (TripAdvisor, 2020). The attr\_index for listing j, based on K points of interest is calculated as:

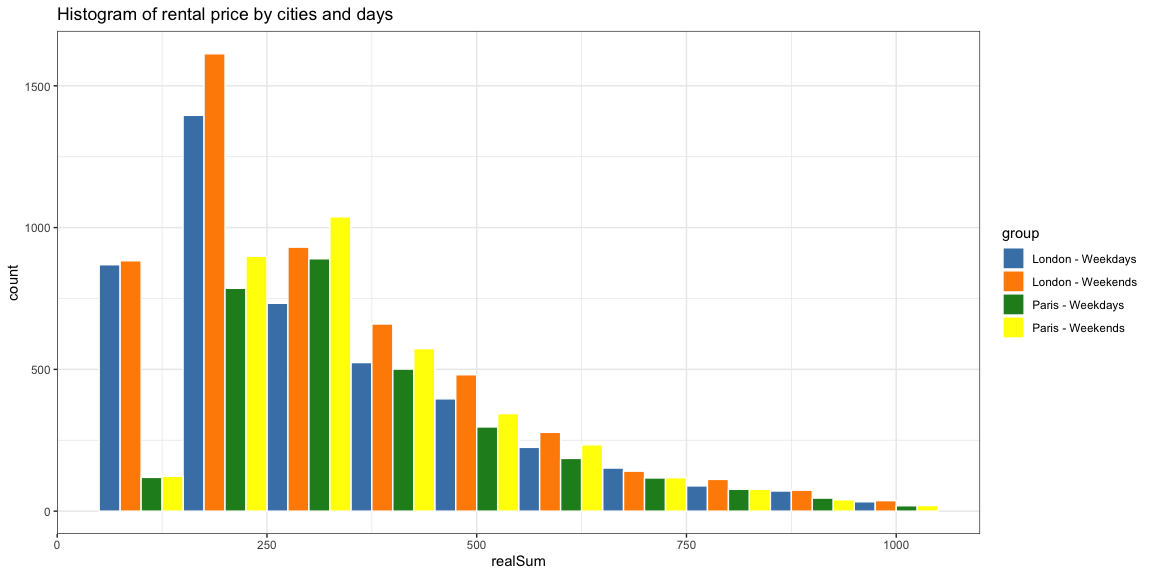
where is the number of reviews for attraction , and is the distance between the listing and point . The calculated value is divided by the maximum value in a given city and multiplied by 100; therefore, the index has a range of 0-100 in all cities. In addition to attr\_index, which is based on the venues from the Attractions category, rest\_index is considered for restaurants.(Gyodi K., & Nawaro L., 2021)

## Summary Statistics

The summary statistics are shown below:



According to the provided table, the original data consists of different sample sizes for Paris\_weekdays, Paris\_weekends, London\_weekdays and London\_weekends, with 3130, 3558, 4614 and 5379 listings respectively. The minimum realSum values across all cases fall within the range of 50 to 96 EUR, while the median value is approximately 300 EUR and the mean ranges from 350 to 400 EUR. However, there is substantial variation in the maximum values spanning from 4, 000 to 16, 000 EUR. To gain a better understanding of the rental rate distribution among various cities and days, histogram plot analysis was conducted.

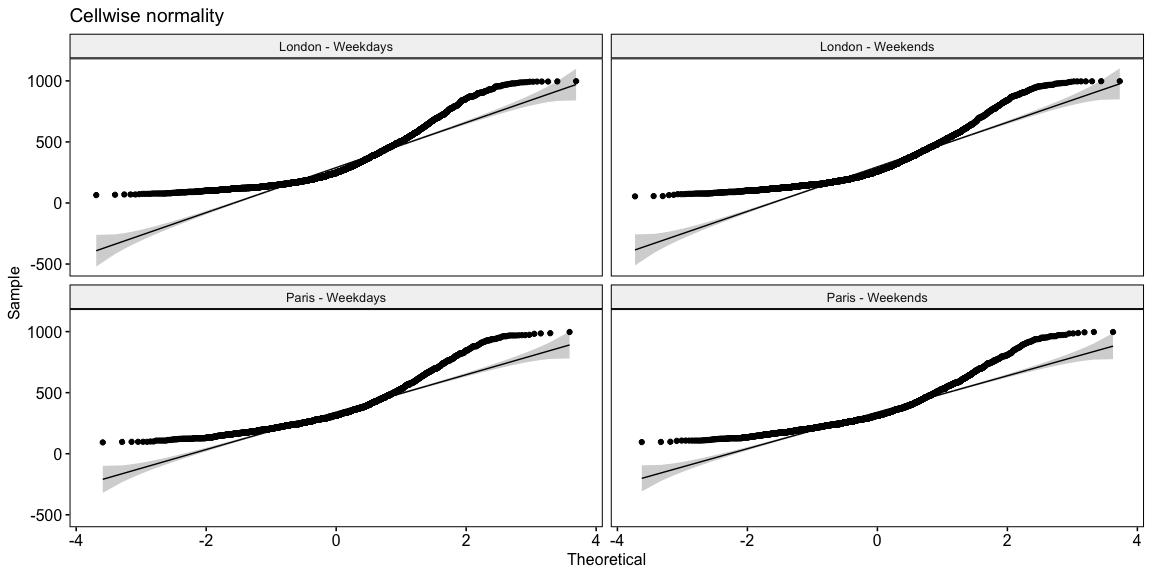
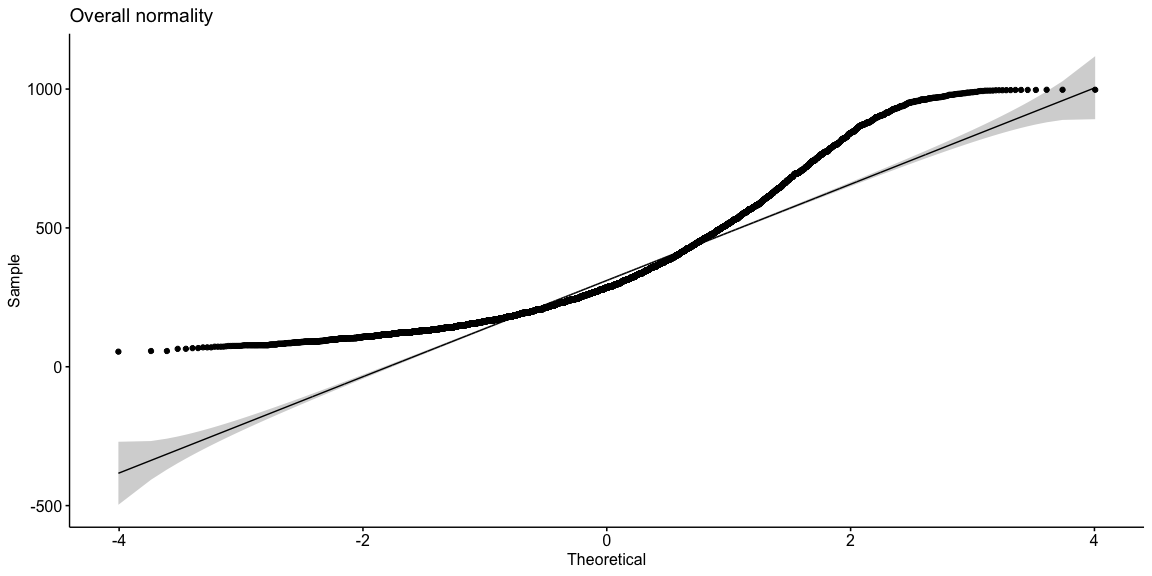


Based on the depicted plot above, the distribution patterns of rental rates across various cities and days exhibit similarity, with a significant majority of listings having a realSum value of less than 1, 000. Only a small proportion, specifically 479 listings, have a realSum value exceeding 1, 000. This constitutes a minor portion when compared to the total number of listings, which is 16, 681. To mitigate potential errors in data collection and account for any unknown factors that might contribute to higher values, we will concentrate on the listings with a realSum below 1, 000 as our primary focus.

# DATA ANALYSIS AND METHOD

## Normality of the data

Linear regression is commonly employed method for modelling the relationship between a scalar response and one or more explanatory variables. it is widely used in predicting rental prices. However, the assumption of linearity in the model necessitates that the data conforms to the normality requirement.



Normality analysis was performed, but based on the plots presented above, indicate that the realSum values, whether observed overall or on a cellwise basis, do not conform to the assumption of normality. Therefore, Linear regression approach is not suitable for construction a predictive model using these data due to the violation the normality assumption in the distribution of realSum values.

## Method & model

### Regression tree

Regression trees are a type of decision tree, which is a hierarchical decision support model tht utilizes a tree-like structure to represent decisions and their potential consequences. Unlike linear regression, regression trees are nonlinear predictive models and do not necessarily rely on the assumption of normality. The flowchart-like structure of a decision tree includes internal nodes that represent attribute tests, and leaf nodes that indicate class labels or decisions made after considering all attributes.

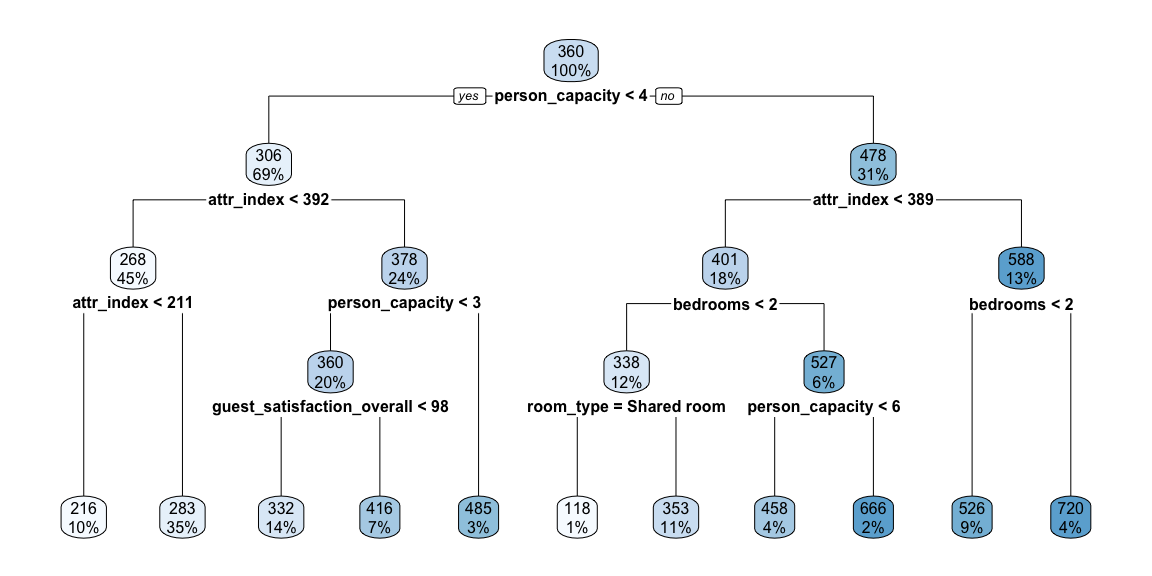
Decision can be classified into two main types: regression trees and classification trees. Regression trees are used when the target variable can take continuous values. For example, regression trees can be utilized predict the price of a house, the length of stay for a patient in a hospital, or an individual’s salary.

### Tree pruning

Tree pruning is a data compression technique employed to decrease the size of decision trees by eliminating non-critical and redundant sections of the tree that are used for classifying instances. By reducing the complexity of the final classifier, pruning helps improve predictive accuracy by mitigating overfitting.

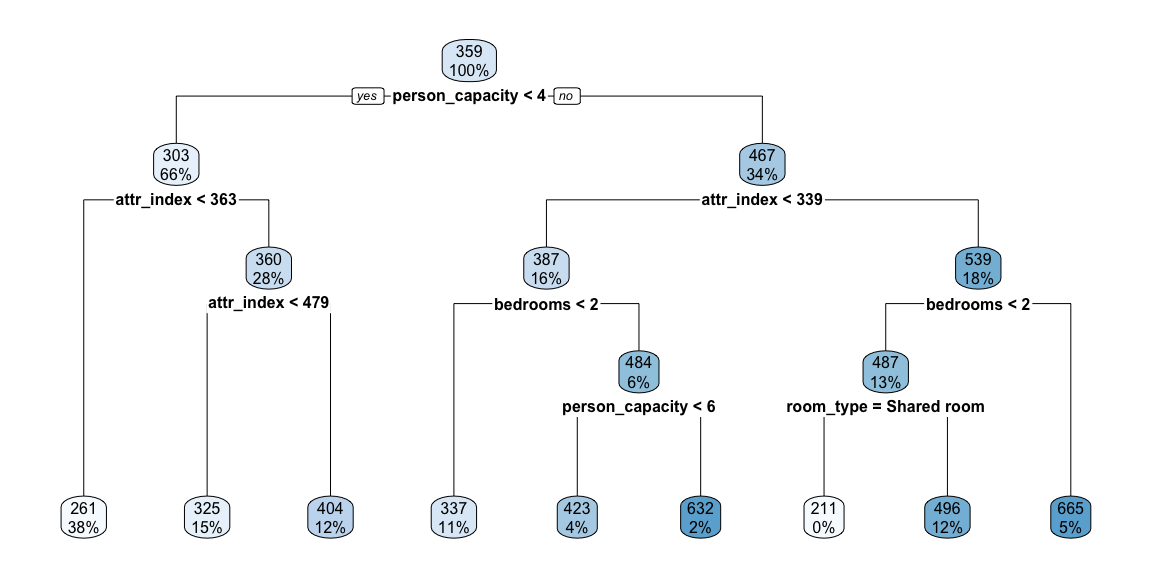
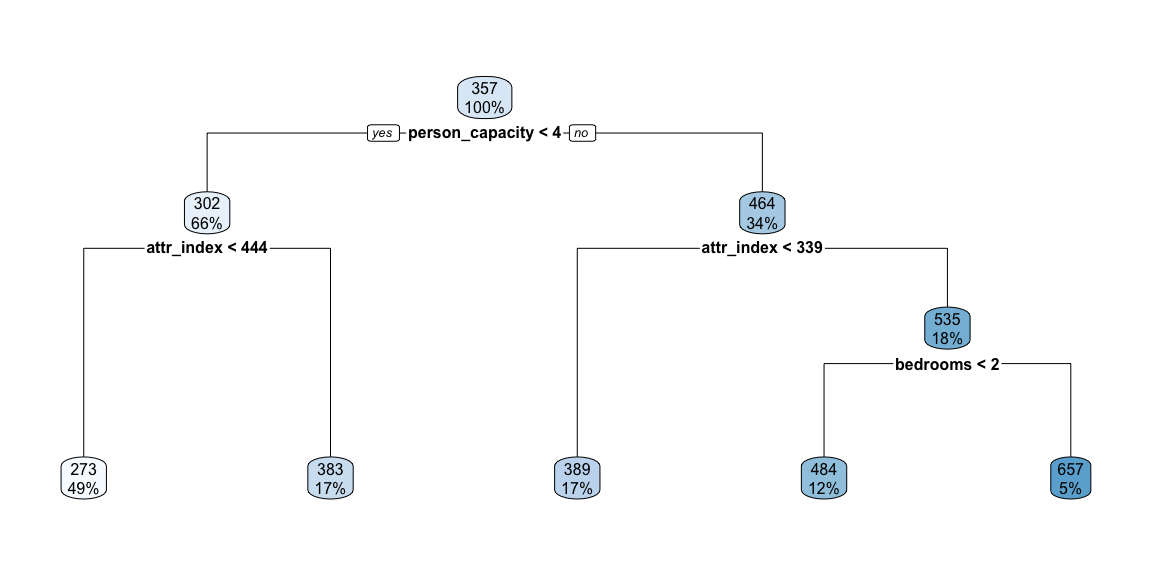
In the following analysis, the regression tree approach is employed to generate an overall model for each city on weekdays or weekends using the training data. Subsequently, the tree pruning method is applied to simplify the tree models based on the validation data, utilizing the criterion of minimizing the mean squared error.

### Paris\_weekdays

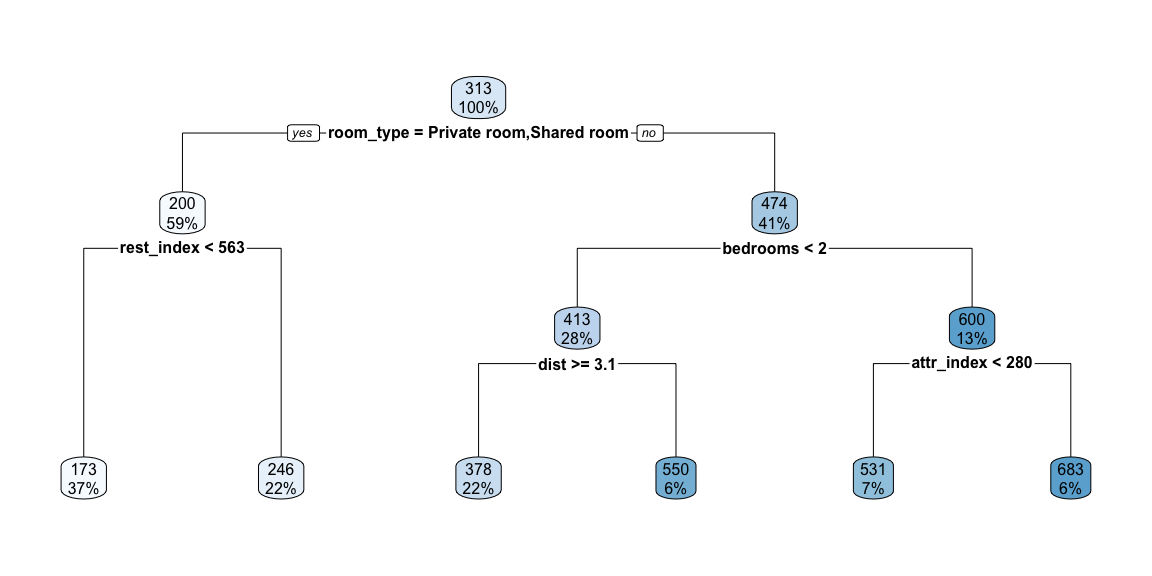
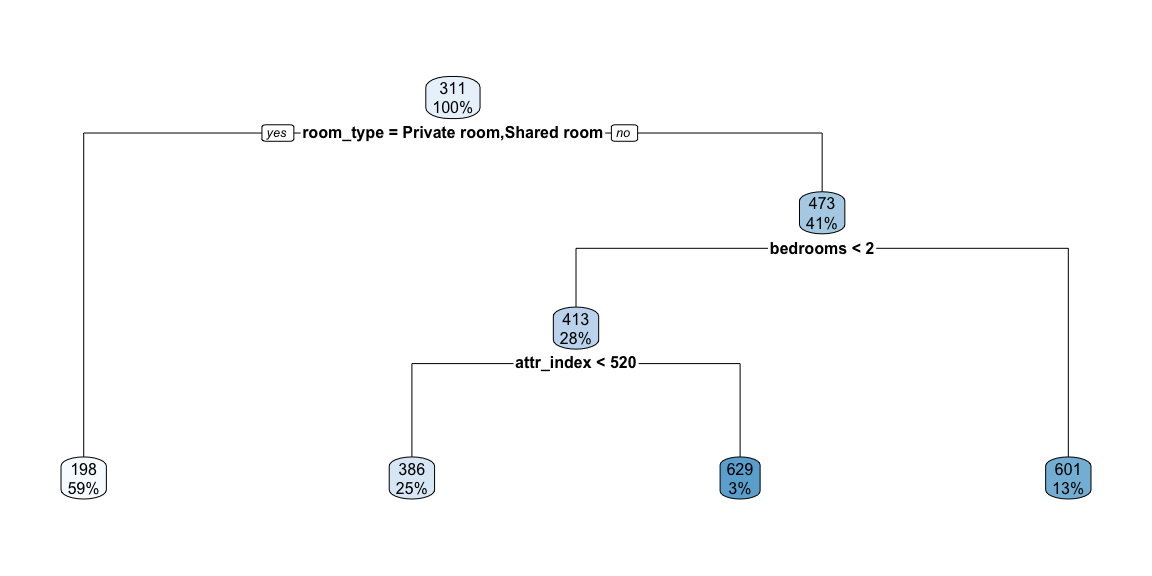
A picture containing text, screenshot, diagram, line

Description automatically generated

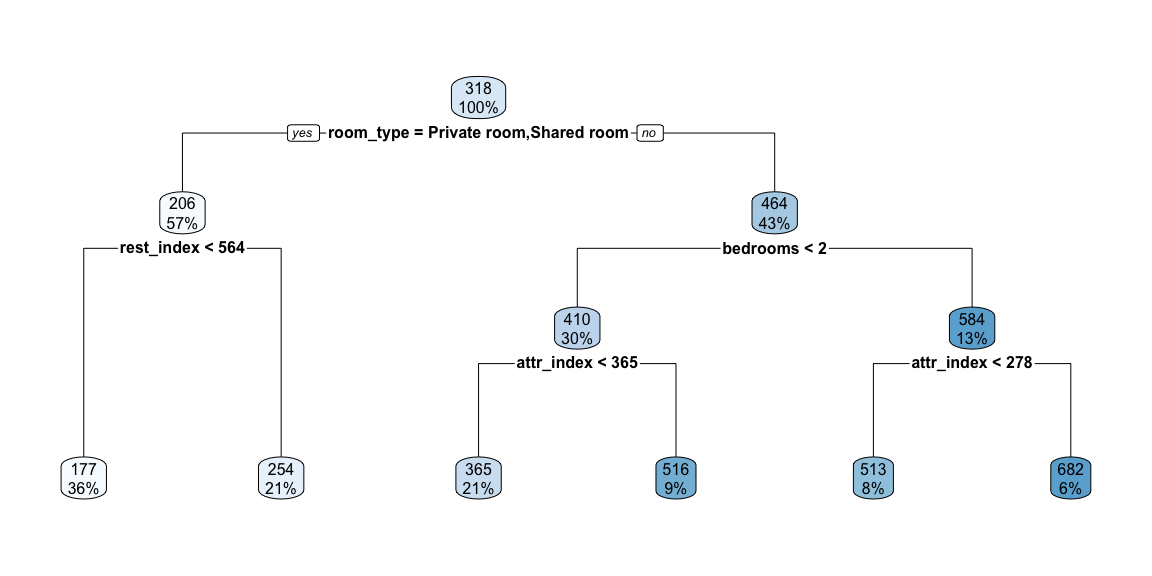
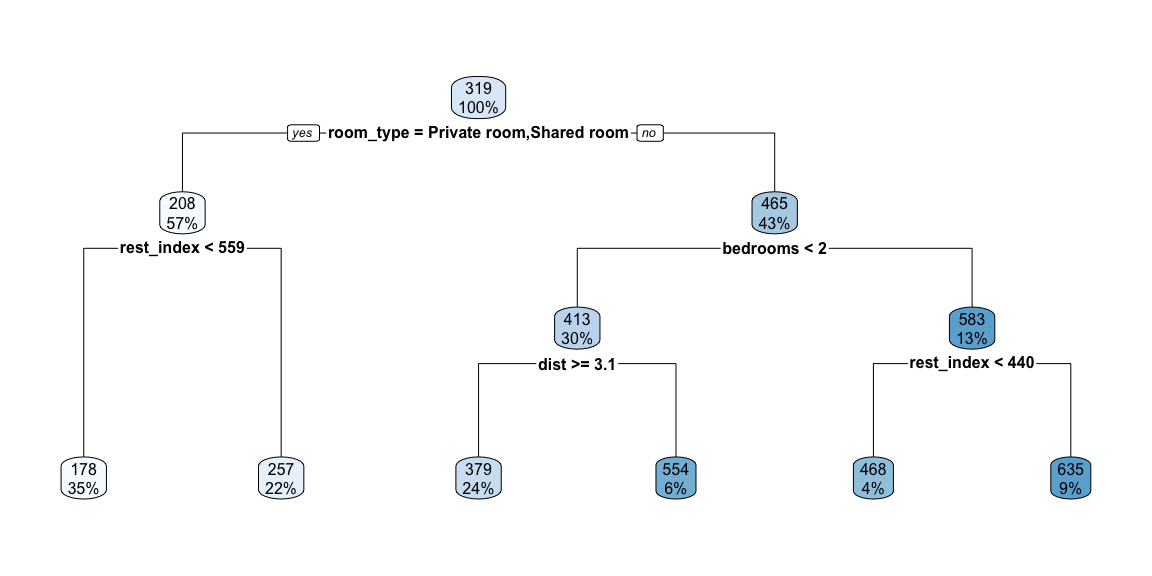
### Paris\_weekends

### London\_weekdays

### London\_weekends

# Conclusion

The key factors influencing the rental rates of the two cities are different. For Paris: person\_capacity, attr\_index, and bedrooms numbers are the critical variables for prediction of rental price. However, for London: room\_type, bedroom numbers, attr\_index, rest\_index and dist are the critical variables for prediction of rental price.

1. Paris\_weekdays:

The first split is . For less than 4, if less than 392, the predict rental price is 268 EUR, if more than 392, the predict rental price is 378 EUR; for more than 4, if less than 389, and number less than 2,the predict rental price is 338 EUR, if less than 389, and bedroom number more than 2, the predict rental price is 527 EUR; if more than 398, the predict rental price is 588 EUR.

Based on this result, we can figure out the trend that during weekdays, the Airbnb rental prices are around 268 to 588 EUR, and it is growing as the room size getting bigger and increases.

1. Paris\_weekends:

The first split is . For less than 4, if less than 444, the predict rental price is 273 EUR, if more than 444, the predict rental price is 383 EUR; for more than 4, if less than 339, the predict rental price is 389 EUR; if more than 339 and number less than 2,the predict rental price is 484 EUR, if more than 339, and number more than 2, the predict rental price is 657 EUR.

Similar to the rental price in Paris during weekdays, the trend during weekend is: the Airbnb rental prices are around 273 to 657 EUR, overall, it is higher compared with the price during weekdays. And it is also growing as the room size getting bigger and increases.

1. London\_weekdays:

The first split is . For is private or shared room, the predict rental price is 198 EUR; for is entire home/apt, if number less than 2, and less than 520, the predict rental price is 386 EUR, if number less than 2, and less than 520, the predict rental price is 629 EUR; if number more than 2, the predict rental price is 601 EUR.

Based on this result, we can figure out the trend that during weekdays, the Airbnb rental prices are around 198 to 601 EUR. The entire home/apt is more costly as well as room size and increases.

1. London\_weekends:

The first split is . For is private or shared room, if is less than 559, the predict rental price is 178 EUR; if is more than 559, the predict rental price is 257 EUR. For is entire home/apt, if number less than 2, and more than or equal to 3.1, the predict rental price is 379 EUR; if number less than 2, and less than 3.1, the predict rental price is 554 EUR; if number more than 2, and less than 440, the predict rental price is 468 EUR; if number more than 2, and more than 440, the predict rental price is 635 EUR.

It shows significant differences compared with the rental price during weekdays. The Airbnb rental prices are around 178 to 635 EUR.Unlike it during weekdays, the rental price during weekends in London is more relying on and , the higher value of and more close to the center, the predict rental price goes higher, indicating that the difference of human activity between weekdays and weekends in London have an significant impact on the rental price.

# Citation

Gyodi, K., & Nawaro, L.(2021). Determinants of Airbnb prices in European cities: A spatial econometrics approach. Tourism Management 86(2021) 104319 <http://doi.org/10/1016/j.tourman.2021.104319>

TripAdvisor.(2020). About tripadvisor. Available online at: <Https://tripadvisor.mediaroom.com/us-about-us>. (Accessed 19 January 2020)

Yang, Y., Mao, Z., & Tang, J.(2018). Understanding guest satisfaction with urban hotel location. Journal of Travel Research, 57, 243-259. <https://doi.org/10.3390/su9091635>

# APPENDIX: The Code

knitr::opts\_chunk$set(fig.align="center",  
 fig.height=6,  
 fig.width=12,  
 warning = FALSE,  
 message = FALSE,  
 comment = NA,  
 echo=FALSE)  
#install.packages("ISLR")  
library(tidyverse)# dplyr::tibble  
library(knitr) #allows you to create Appendix with all\_labels()  
library(tidyverse)  
library(ISLR) #Auto dataset  
theme\_set(theme\_bw()) #sets default ggplot output style  
library(ggplot2)  
library(rpart)  
library(rpart.plot)  
dat.paris.weekdays<-read.csv("paris\_weekdays.csv") # 3130 entries  
dat.paris.weekends<-read.csv("paris\_weekends.csv") # 3558 entries  
dat.london.weekdays<-read.csv("london\_weekdays.csv") # 4614 entries  
dat.london.weekends<-read.csv("london\_weekends.csv") # 5,379 entries  
#View summary statistics  
summary(dat.paris.weekdays)  
summary(dat.paris.weekends)  
summary(dat.london.weekdays)  
summary(dat.london.weekends)  
city\_days <- c("Paris weekdays","Paris weekends", "London weekdays", "London weekends")  
Number\_of\_listings <- c(3130, 3558, 4614, 5379)  
realSum\_min <- c(92.74, 95.3, 64.68, 54.33)  
realSum\_median <- c(318.53, 316.2, 256.36, 268.12)  
realSum\_mean <- c(398.79, 387.0, 360, 364.39)  
realSum\_max <- c(16455.62, 4188, 15499.89, 12937.27)  
data\_summary <- data.frame(city\_days, Number\_of\_listings, realSum\_min, realSum\_median, realSum\_mean, realSum\_max)  
library(gridExtra)  
library(grid)  
title <- textGrob("Number of listings and descriptive statistics for prices(in EUR)", gp = gpar(fontsize = 16, fontface = "bold"))  
table <- tableGrob(data\_summary, theme = ttheme\_default() )  
grid.arrange(title, table, nrow = 4)  
#View summary statistics  
dat.realSum <- rbind(  
 transform(dat.paris.weekdays$realSum, group = "Paris - Weekdays"),  
 transform(dat.paris.weekends$realSum, group = "Paris - Weekends"),  
 transform(dat.london.weekdays$realSum, group = "London - Weekdays"),  
 transform(dat.london.weekends$realSum, group = "London - Weekends")  
) # 16681 entries  
colnames(dat.realSum)[1] <- "realSum"  
library(ggplot2)  
  
ggplot(dat.realSum, aes(x = X\_data, fill = group)) +  
 geom\_histogram(stat = "bin", binwidth = 500, color = "white",position = "dodge") +  
 labs(title = "Histogram of realSum by cities and days") +  
 scale\_fill\_manual(values = c("steelblue", "darkorange", "forestgreen", "yellow"))   
#View summary statistics  
dat.realSum <- dat.realSum%>%filter(realSum <1000) # 16202 entries  
  
ggplot(dat.realSum, aes(x = realSum, fill = group)) +  
 geom\_histogram(stat = "bin", binwidth = 100, color = "white",position = "dodge") +  
 labs(title = "Histogram of rental price by cities and days") +  
 scale\_fill\_manual(values = c("steelblue", "darkorange", "forestgreen", "yellow"))   
library(ggpubr)  
ggqqplot(dat.realSum$realSum) + labs(title = "Overall normality")  
ggqqplot(dat.realSum,x="realSum", facet.by = c("group")) + labs(title = "Cellwise normality")  
# paris.weekdays  
## filter the samples realSum less than 1000  
dat.paris.weekdays <- dat.paris.weekdays%>%filter(dat.paris.weekdays$realSum < 1000) # 3038 entries  
## remove the index column "X"  
dat.paris.weekdays <- dat.paris.weekdays[, -which(names(dat.paris.weekdays)=="X")]  
  
n<-nrow(dat.paris.weekdays)  
  
set.seed(12323)  
tvt<-sample(rep(0:2,c(round(n\*.2),round(n\*.2),n-2\*round(n\*.2))),n)  
#mean(dat.paris.weekdays$realSum[tvt==0])  
#mean(dat.paris.weekdays$realSum[tvt==1])  
#mean(dat.paris.weekdays$realSum[tvt==2])  
#mean(dat.paris.weekdays$realSum)  
  
dat.paris.weekdays.train<-dat.paris.weekdays[tvt==2,] # 1822 entries  
dat.paris.weekdays.valid<-dat.paris.weekdays[tvt==1,] # 608 entries  
dat.paris.weekdays.test<-dat.paris.weekdays[tvt==0,] # 608 entries  
# regression tree   
model.paris.weekdays <- rpart(realSum ~., dat.paris.weekdays.train, method = "anova")  
# summary(model.paris.weekdays)  
# visualization of the regression tree model  
rpart.plot(model.paris.weekdays)  
  
best.cp <- NULL   
best.error <- Inf   
for (cp in seq(0.01, 0.04, by = 0.01)) {  
 model <- rpart(formula = realSum ~ ., data = dat.paris.weekdays.train, method = "anova", cp = cp)  
 validation.predictions <- predict(model, newdata = dat.paris.weekdays.valid)  
 validation.error <- mean((validation.predictions - dat.paris.weekdays.valid$realSum)^2)  
   
 if (validation.error < best.error) {  
 best.cp <- cp  
 best.error <- validation.error  
 }  
}  
final\_model <- prune(model, cp = best.cp)  
#summary(final\_model)  
rpart.plot(final\_model)  
  
test\_predictions <- predict(final\_model, newdata = dat.paris.weekdays.test)  
mse\_test <- mean((test\_predictions - dat.paris.weekdays.test$realSum)^2)  
#mse\_test # 20752.29  
  
test\_predictions.ori <- predict(model.paris.weekdays, newdata = dat.paris.weekdays.test)  
mse\_test.ori <- mean((test\_predictions.ori - dat.paris.weekdays.test$realSum)^2)  
#mse\_test.ori # 20752.29  
  
# paris.weekends  
## filter the samples realSum less than 1000  
dat.paris.weekends <- dat.paris.weekends%>%filter(realSum < 1000) # 3466 entries  
## remove the index column "X"  
dat.paris.weekends <- dat.paris.weekends[, -which(names(dat.paris.weekends)=="X")]  
  
n<-nrow(dat.paris.weekends)  
  
set.seed(12323)  
tvt<-sample(rep(0:2,c(round(n\*.2),round(n\*.2),n-2\*round(n\*.2))),n)  
#mean(dat.paris.weekends$realSum[tvt==0])  
#mean(dat.paris.weekends$realSum[tvt==1])  
#mean(dat.paris.weekends$realSum[tvt==2])  
#mean(dat.paris.weekends$realSum)  
  
dat.paris.weekends.train<-dat.paris.weekends[tvt==2,] # 2080 entries  
dat.paris.weekends.valid<-dat.paris.weekends[tvt==1,] # 693 entries  
dat.paris.weekends.test<-dat.paris.weekends[tvt==0,] # 693 entries  
  
  
# regression tree   
model.paris.weekends <- rpart(realSum ~., dat.paris.weekends, method = "anova")  
#summary(model.paris.weekends)  
# visualization of the regression tree model  
rpart.plot(model.paris.weekends)  
  
best.cp <- NULL   
best.error <- Inf   
for (cp in seq(0.01, 0.03, by = 0.01)) {  
 model <- rpart(formula = realSum ~ ., data = dat.paris.weekends.train, method = "anova", cp = cp)  
 validation.predictions <- predict(model, newdata = dat.paris.weekends.valid)  
 validation.error <- mean((validation.predictions - dat.paris.weekends.valid$realSum)^2)  
   
 if (validation.error < best.error) {  
 best.cp <- cp  
 best.error <- validation.error  
 }  
}  
final\_model <- prune(model, cp = best.cp)  
#summary(final\_model)  
rpart.plot(final\_model)  
  
test\_predictions <- predict(final\_model, newdata = dat.paris.weekends.test)  
mse\_test <- mean((test\_predictions - dat.paris.weekends.test$realSum)^2)  
#mse\_test # 17324.72  
  
test\_predictions.ori <- predict(model.paris.weekends, newdata = dat.paris.weekends.test)  
mse\_test.ori <- mean((test\_predictions.ori - dat.paris.weekends.test$realSum)^2)  
#mse\_test.ori # 14680.37  
  
# london.weekdays  
## filter the samples realSum less than 1000  
dat.london.weekdays <- dat.london.weekdays%>%filter(realSum < 1000) # 4488 entries  
## remove the index column "X"  
dat.london.weekdays <- dat.london.weekdays[, -which(names(dat.london.weekdays)=="X")]  
  
n<-nrow(dat.london.weekdays)  
  
set.seed(12323)  
tvt<-sample(rep(0:2,c(round(n\*.2),round(n\*.2),n-2\*round(n\*.2))),n)  
#mean(dat.london.weekdays$realSum[tvt==0])  
#mean(dat.london.weekdays$realSum[tvt==1])  
#mean(dat.london.weekdays$realSum[tvt==2])  
#mean(dat.london.weekdays$realSum)  
  
dat.london.weekdays.train<-dat.london.weekdays[tvt==2,] # 2692 entries  
dat.london.weekdays.valid<-dat.london.weekdays[tvt==1,] # 898 entries  
dat.london.weekdays.test<-dat.london.weekdays[tvt==0,] # 898 entries  
  
  
# regression tree   
model.london.weekdays <- rpart(realSum ~., dat.london.weekdays, method = "anova")  
#summary(model.london.weekdays)  
# visualization of the regression tree model  
rpart.plot(model.london.weekdays)  
  
best.cp <- NULL   
best.error <- Inf   
for (cp in seq(0.01, 0.03, by = 0.01)) {  
 model <- rpart(formula = realSum ~ ., data = dat.london.weekdays.train, method = "anova", cp = cp)  
 validation.predictions <- predict(model, newdata = dat.london.weekdays.valid)  
 validation.error <- mean((validation.predictions - dat.london.weekdays.valid$realSum)^2)  
   
 if (validation.error < best.error) {  
 best.cp <- cp  
 best.error <- validation.error  
 }  
}  
final\_model <- prune(model, cp = best.cp)  
#summary(final\_model)  
rpart.plot(final\_model)  
  
test\_predictions <- predict(final\_model, newdata = dat.london.weekdays.test)  
mse\_test <- mean((test\_predictions - dat.london.weekdays.test$realSum)^2)  
#mse\_test # 16622.35  
  
test\_predictions.ori <- predict(model.london.weekdays, newdata = dat.london.weekdays.test)  
mse\_test.ori <- mean((test\_predictions.ori - dat.london.weekdays.test$realSum)^2)  
#mse\_test.ori # 15281.41  
  
  
# london.weekends  
## filter the samples realSum less than 1000  
dat.london.weekends <- dat.london.weekends%>%filter(realSum < 1000) # 5210 entries  
## remove the index column "X"  
dat.london.weekends <- dat.london.weekends[, -which(names(dat.london.weekends)=="X")]  
  
n<-nrow(dat.london.weekends)  
  
set.seed(12323)  
tvt<-sample(rep(0:2,c(round(n\*.2),round(n\*.2),n-2\*round(n\*.2))),n)  
#mean(dat.london.weekends$realSum[tvt==0])  
#mean(dat.london.weekends$realSum[tvt==1])  
#mean(dat.london.weekends$realSum[tvt==2])  
#mean(dat.london.weekends$realSum)  
  
dat.london.weekends.train<-dat.london.weekends[tvt==2,] # 3126 entries  
dat.london.weekends.valid<-dat.london.weekends[tvt==1,] # 1042 entries  
dat.london.weekends.test<-dat.london.weekends[tvt==0,] # 1042 entries  
# regression tree   
model.london.weekends <- rpart(realSum ~., dat.london.weekends, method = "anova")  
#summary(model.london.weekends)  
# visualization of the regression tree model  
rpart.plot(model.london.weekends)  
  
best.cp <- NULL   
best.error <- Inf   
for (cp in seq(0.01, 0.02, by = 0.01)) {  
 model <- rpart(formula = realSum ~ ., data = dat.london.weekends.train, method = "anova", cp = cp)  
 validation.predictions <- predict(model, newdata = dat.london.weekends.valid)  
 validation.error <- mean((validation.predictions - dat.london.weekends.valid$realSum)^2)  
   
 if (validation.error < best.error) {  
 best.cp <- cp  
 best.error <- validation.error  
 }  
}  
final\_model <- prune(model, cp = best.cp)  
#summary(final\_model)  
rpart.plot(final\_model)  
  
test\_predictions <- predict(final\_model, newdata = dat.london.weekends.test)  
mse\_test <- mean((test\_predictions - dat.london.weekends.test$realSum)^2)  
#mse\_test # 16622.35  
  
test\_predictions.ori <- predict(model.london.weekends, newdata = dat.london.weekends.test)  
mse\_test.ori <- mean((test\_predictions.ori - dat.london.weekends.test$realSum)^2)  
#mse\_test.ori # 15281.41