Logistic Regression -Dailies 12/22/24

Linda

2024-12-23

Load the libraries

library(tidyverse)  
library(ggplot2)  
library(openxlsx)  
library(caret)  
library(fastDummies)

## Load the dataset

data<-read.xlsx(file.choose())

## Exploratory Data Analysis

Explore the dataset

head(data) # 6 first rows

## id name host\_id host\_name  
## 1 2539 Clean & quiet apt home by the park 2787 John  
## 2 2595 Skylit Midtown Castle 2845 Jennifer  
## 3 3647 THE VILLAGE OF HARLEM....NEW YORK ! 4632 Elisabeth  
## 4 3831 Cozy Entire Floor of Brownstone 4869 LisaRoxanne  
## 5 5022 Entire Apt: Spacious Studio/Loft by central park 7192 Laura  
## 6 5099 Large Cozy 1 BR Apartment In Midtown East 7322 Chris  
## neighbourhood\_group neighbourhood latitude longitude room\_type price  
## 1 Brooklyn Kensington 40.64749 -73.97237 Private room 149  
## 2 Manhattan Midtown 40.75362 -73.98377 Entire home/apt 225  
## 3 Manhattan Harlem 40.80902 -73.94190 Private room 150  
## 4 Brooklyn Clinton Hill 40.68514 -73.95976 Entire home/apt 89  
## 5 Manhattan East Harlem 40.79851 -73.94399 Entire home/apt 80  
## 6 Manhattan Murray Hill 40.74767 -73.97500 Entire home/apt 200  
## minimum\_nights number\_of\_reviews last\_review reviews\_per\_month  
## 1 1 9 43392 0.21  
## 2 1 45 43606 0.38  
## 3 3 0 NA NA  
## 4 1 270 43651 4.64  
## 5 10 9 43423 0.10  
## 6 3 74 43638 0.59  
## calculated\_host\_listings\_count availability\_365  
## 1 6 365  
## 2 2 355  
## 3 1 365  
## 4 1 194  
## 5 1 0  
## 6 1 129

summary(data) #Descriptive statistics

## id name host\_id host\_name   
## Min. : 2539 Length:48895 Min. : 2438 Length:48895   
## 1st Qu.: 9471945 Class :character 1st Qu.: 7822033 Class :character   
## Median :19677284 Mode :character Median : 30793816 Mode :character   
## Mean :19017143 Mean : 67620011   
## 3rd Qu.:29152178 3rd Qu.:107434423   
## Max. :36487245 Max. :274321313   
##   
## neighbourhood\_group neighbourhood latitude longitude   
## Length:48895 Length:48895 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:48895 Min. : 0.0 Min. : 1.00 Min. : 0.00   
## Class :character 1st Qu.: 69.0 1st Qu.: 1.00 1st Qu.: 1.00   
## Mode :character Median : 106.0 Median : 3.00 Median : 5.00   
## Mean : 152.7 Mean : 7.03 Mean : 23.27   
## 3rd Qu.: 175.0 3rd Qu.: 5.00 3rd Qu.: 24.00   
## Max. :10000.0 Max. :1250.00 Max. :629.00   
##   
## last\_review reviews\_per\_month calculated\_host\_listings\_count  
## Min. :40630 Min. : 0.010 Min. : 1.000   
## 1st Qu.:43289 1st Qu.: 0.190 1st Qu.: 1.000   
## Median :43604 Median : 0.720 Median : 1.000   
## Mean :43377 Mean : 1.373 Mean : 7.144   
## 3rd Qu.:43639 3rd Qu.: 2.020 3rd Qu.: 2.000   
## Max. :43654 Max. :58.500 Max. :327.000   
## NA's :10052 NA's :10052   
## availability\_365  
## Min. : 0.0   
## 1st Qu.: 0.0   
## Median : 45.0   
## Mean :112.8   
## 3rd Qu.:227.0   
## Max. :365.0   
##

Data cleaning

sum(is.na(data))

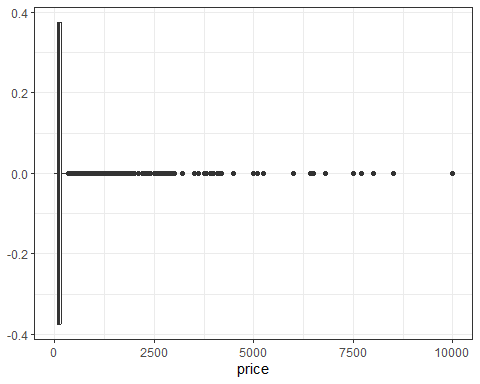
## [1] 20152

##Check missing values  
data$last\_review[is.na(data$last\_review)]<-0  
data$name[is.na(data$name)]<-'Unknown'  
data$host\_name[is.na(data$host\_name)]<-'Unkown'  
data$reviews\_per\_month[is.na(data$reviews\_per\_month)]<-mode(data$reviews\_per\_month)[1]

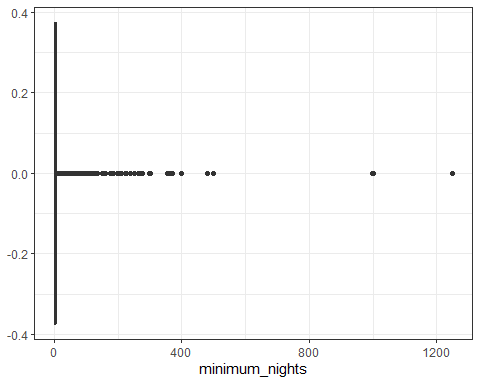
## Check the absence of na values  
colSums(is.na(data))

## 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 0   
## calculated\_host\_listings\_count availability\_365   
## 0 0

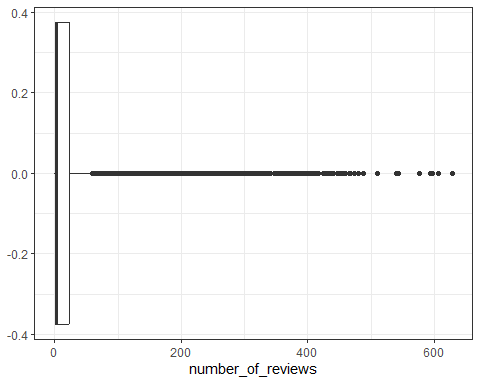
###Check for outliers in the numerical variables using boxplots  
data|>ggplot(aes(price))+geom\_boxplot()+theme\_bw()



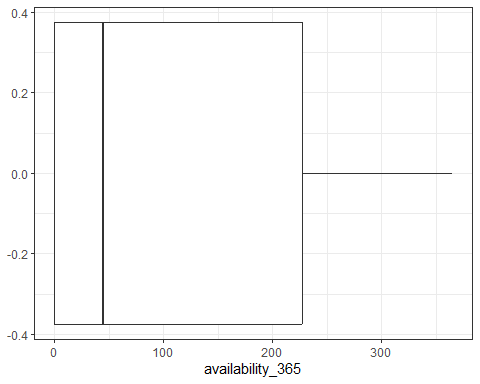
data|>ggplot(aes(minimum\_nights))+geom\_boxplot()+theme\_bw()



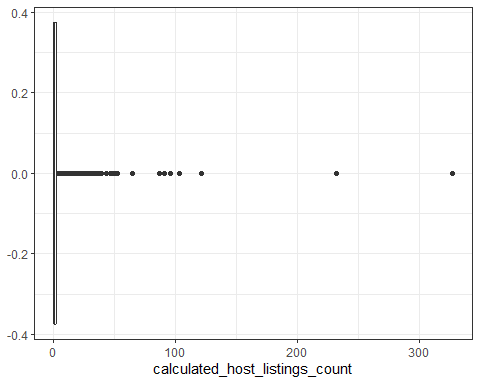
data|>ggplot(aes(number\_of\_reviews))+geom\_boxplot()+theme\_bw()



data|>ggplot(aes(availability\_365))+geom\_boxplot()+theme\_bw()



data|>ggplot(aes(calculated\_host\_listings\_count))+geom\_boxplot()+theme\_bw()



##Removing outliers  
data<-data|>group\_by(room\_type)|>  
 mutate(Q1=quantile(price,.25,na.rm=TRUE),  
 Q3=quantile(price,.75,na.rm=TRUE),  
 IQR=Q3-Q1,  
 Lower\_bound=Q1 -1.5\*IQR,  
 Upper\_bound=Q3+ 1.5\*IQR)|>  
 filter(price>=Lower\_bound & price <= Upper\_bound)|>  
 ungroup()|>  
 select(-Q1,-Q3,-IQR)  
summary(data$price)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 65.0 100.0 121.8 155.0 392.0

data<-data|>  
 mutate(Q1=quantile(minimum\_nights,0.25,na.rm=TRUE),  
 Q3=quantile(minimum\_nights,0.75,na.rm=TRUE),  
 IQR=Q3-Q1,  
 Lower\_bound=Q1-1.5\*IQR,  
 Upper\_bound=Q3+1.5\*IQR)|>  
 filter(minimum\_nights>=Lower\_bound & minimum\_nights<=Upper\_bound)|>  
 select(-Q1,-Q3,-IQR)  
summary(data$minimum\_nights)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 2.000 2.715 3.000 11.000

## Detect and Remove the duplicated rows

sum(duplicated(data))

## [1] 0

data<-data|> ###Removes the duplicated rows  
 distinct()

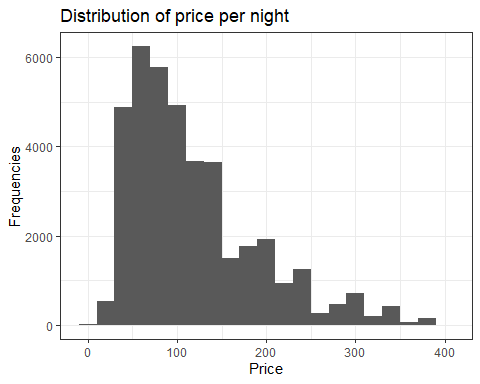
## Select the columns to use to determine what happens when the price of a listing is higher than the mean price of the listings per night using the minimum nights requirement,availability of a listing during the year, the number of reviews and room type.

data<-data|>  
 select('price','minimum\_nights','number\_of\_reviews','availability\_365','room\_type')  
head(data) ## 6 first rows

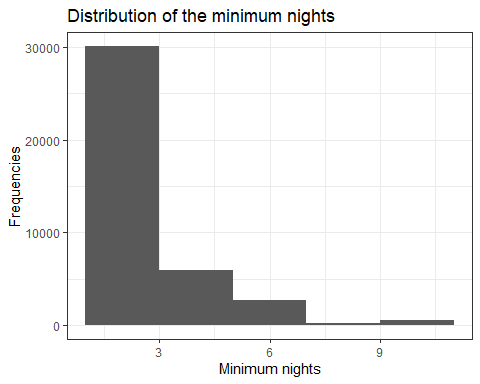
## # A tibble: 6 × 5  
## price minimum\_nights number\_of\_reviews availability\_365 room\_type   
## <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 149 1 9 365 Private room   
## 2 225 1 45 355 Entire home/apt  
## 3 150 3 0 365 Private room   
## 4 89 1 270 194 Entire home/apt  
## 5 80 10 9 0 Entire home/apt  
## 6 200 3 74 129 Entire home/apt

Histograms for numerical variables

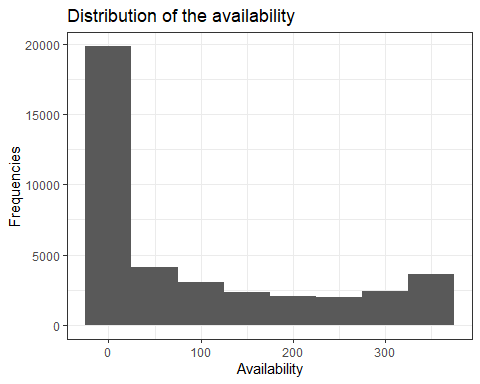
data|>ggplot(aes(price))+  
 geom\_histogram(binwidth = 20,position ='identity')+  
 theme\_bw()+  
 labs(x='Price',y='Frequencies',title='Distribution of price per night')



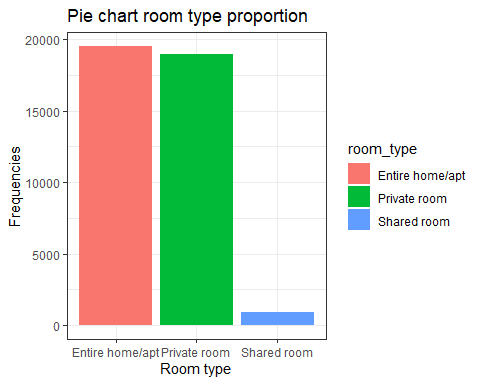
data|>ggplot(aes(minimum\_nights))+geom\_histogram(binwidth = 2)+  
 theme\_bw()+  
 labs(x='Minimum nights',y='Frequencies',title='Distribution of the minimum nights')



data|>ggplot(aes(availability\_365))+geom\_histogram(binwidth = 50 )+  
 theme\_bw()+  
 labs(x='Availability',y='Frequencies',title='Distribution of the availability ')

 ## Bar chart for the room type proportions

data|>ggplot(aes(room\_type,fill=room\_type))+geom\_bar()+  
 theme\_bw()+labs(x='Room type',y='Frequencies',title='Pie chart room type proportion')



## Converting categorical variables to numerical variables

data$price\_class<-ifelse(data$price>mean(data$price,na.rm=TRUE),1,0) #Add column for converted price values  
##data<-dummy\_cols(data, select\_columns ='room\_type' )##Create room type dummy variables  
head(data)

## # A tibble: 6 × 6  
## price minimum\_nights number\_of\_reviews availability\_365 room\_type price\_class  
## <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 149 1 9 365 Private r… 1  
## 2 225 1 45 355 Entire ho… 1  
## 3 150 3 0 365 Private r… 1  
## 4 89 1 270 194 Entire ho… 0  
## 5 80 10 9 0 Entire ho… 0  
## 6 200 3 74 129 Entire ho… 1

## Data partitioning

Splitting the dataset into train and test

set.seed(1124)#For reproducibility  
library(caret)  
index<-createDataPartition(data$price\_class,  
 p=.7,list=FALSE)  
data\_train<-data[index,]  
data\_test<-data[-index,]  
head(data\_train)

## # A tibble: 6 × 6  
## price minimum\_nights number\_of\_reviews availability\_365 room\_type price\_class  
## <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 150 3 0 365 Private r… 1  
## 2 200 3 74 129 Entire ho… 1  
## 3 79 2 430 220 Private r… 0  
## 4 79 2 118 0 Private r… 0  
## 5 150 1 160 188 Entire ho… 1  
## 6 135 5 53 6 Entire ho… 1

## Build the model

This involves training the model on the expected output

model<-glm(price\_class~minimum\_nights+number\_of\_reviews+availability\_365+room\_type,family=binomial,data=data\_train)  
summary(model)

##   
## Call:  
## glm(formula = price\_class ~ minimum\_nights + number\_of\_reviews +   
## availability\_365 + room\_type, family = binomial, data = data\_train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.8303727 0.0367923 22.569 < 2e-16 \*\*\*  
## minimum\_nights -0.0153400 0.0089745 -1.709 0.0874 .   
## number\_of\_reviews -0.0029714 0.0003782 -7.856 3.95e-15 \*\*\*  
## availability\_365 0.0017946 0.0001449 12.386 < 2e-16 \*\*\*  
## room\_typePrivate room -3.4349783 0.0390144 -88.044 < 2e-16 \*\*\*  
## room\_typeShared room -5.7188774 0.4112487 -13.906 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 36682 on 27561 degrees of freedom  
## Residual deviance: 23580 on 27556 degrees of freedom  
## AIC: 23592  
##   
## Number of Fisher Scoring iterations: 7

## Test the model

data\_test$predicted\_probabilities<-predict(model,newdata=data\_test,type='response')  
##Convert the predicted probabilities to binary of 1's and 0's with a threshold of .5  
data\_test$predicted\_class<-ifelse(data\_test$predicted\_probabilities>.5,1,0)  
  
head(data\_test)

## # A tibble: 6 × 8  
## price minimum\_nights number\_of\_reviews availability\_365 room\_type price\_class  
## <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 149 1 9 365 Private r… 1  
## 2 225 1 45 355 Entire ho… 1  
## 3 89 1 270 194 Entire ho… 0  
## 4 80 10 9 0 Entire ho… 0  
## 5 85 2 113 333 Private r… 0  
## 6 140 2 148 46 Entire ho… 1  
## # ℹ 2 more variables: predicted\_probabilities <dbl>, predicted\_class <dbl>

## Validate the model

table(prediction=data\_test$predicted\_class, Actual=data\_test$price\_class)

## Actual  
## prediction 0 1  
## 0 5480 439  
## 1 1715 4178

## Confusion matrix

To validate the model accuracy and precision

confusion\_Matrix<-confusionMatrix(table(prediction=data\_test$predicted\_class, Actual=data\_test$price\_class))  
###Get the confusion matrix  
confusion\_Matrix

## Confusion Matrix and Statistics  
##   
## Actual  
## prediction 0 1  
## 0 5480 439  
## 1 1715 4178  
##   
## Accuracy : 0.8176   
## 95% CI : (0.8106, 0.8246)  
## No Information Rate : 0.6091   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6351   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.7616   
## Specificity : 0.9049   
## Pos Pred Value : 0.9258   
## Neg Pred Value : 0.7090   
## Prevalence : 0.6091   
## Detection Rate : 0.4639   
## Detection Prevalence : 0.5011   
## Balanced Accuracy : 0.8333   
##   
## 'Positive' Class : 0   
##

##Histogram To visualize the predicted probanilities

data\_test|>ggplot(aes(predicted\_probabilities,fill=room\_type))+geom\_histogram(binwidth = .1,alpha=.5)+theme\_bw()+  
 labs(x='Predicted Probabilities',y='Frequencies',title='Distribution of the predicted probabilities')

