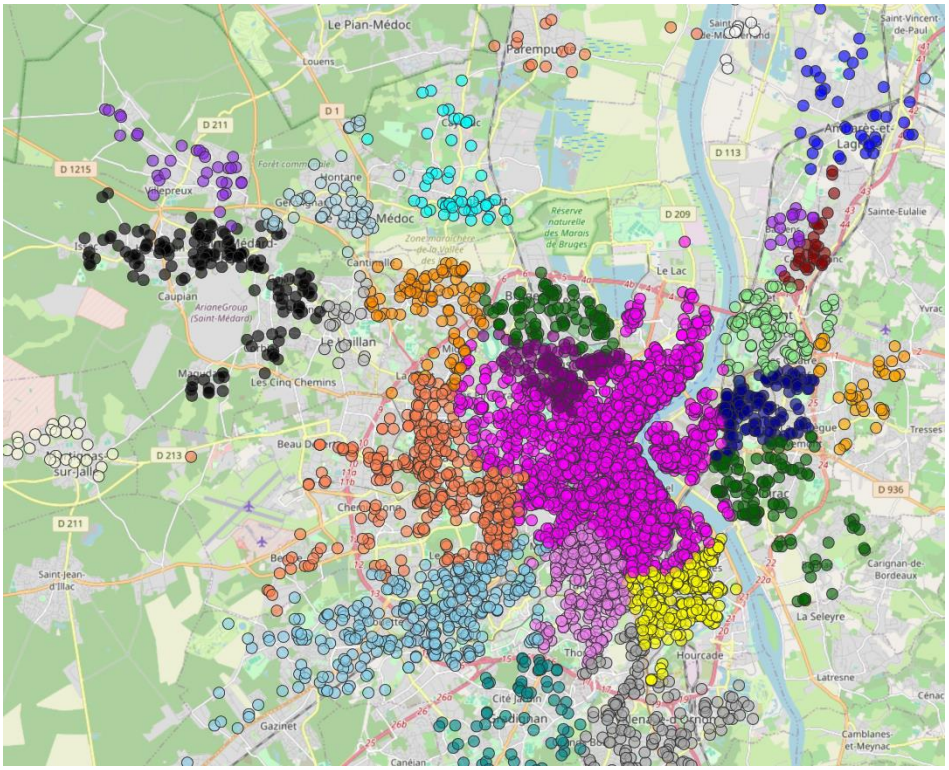
INSTITUTO UNIVERSITÁRIO DE LISBOA

Relatório

2º Semestre Ano Letivo 23/24

Unidade Curricular de Introdução a Modelos Dinâmicos

Docentes Diana Aldea Mendes/Conceição Figueiredo



Airbnb's em Bourdeaux

2º ano – LCD – CDB1/CDB2

Grupo 18

Diogo Aqueu – 110705
Eduardo Parracho – 111423
Gabriel Matos – 110907
Gonçalo Girão – 111515
Tomé Marques – 110966

Lisboa, 12 de abril de 2024

ÍNDICE

Introdução

No âmbito da unidade curricular de Introdução a Modelos Dinâmicos, foi sugerido, como projeto, a realização de uma previsão dos preços de mercado de AL's / hotéis que, no caso, pertencem ao mercado da plataforma AirBNB. Para a criação deste modelo recorreu-se ao software R.

A realização deste projeto foi efetuada através duma base de dados, "listings.csv", na qual se encontram dados sobre a cidade Bordeaux. Na informação que esta base de dados contém encontra-se um conjunto de variáveis que servirão para criar um modelo de previsão dos preços dos AL's / hotéis em Bordeaux. Esta base de dados possui 18 variáveis e cerca de 11 854 linhas e para melhor entender a variável target será benéfico compreender também as restantes variáveis que a acompanham.

Compreensão do tema/Contexto

O Airbnb é um serviço online onde é possível reservar acomodações de forma aos hóspedes conseguirem aproveitar ao máximo a sua estadia.

A base de dados utilizada para este trabalho foi retirada do site 'InsideAirbnb'. Este site tem como objetivo disponibilizar bases de dados sobre o impacto dos Airbnb's nos vários países e cidades ao longo de um ano. Foi-nos atribuída a cidade de Bordeaux.

Bordéus é uma cidade emblemática situada no sudoeste de França com cerca de 260 352 habitantes, é conhecida pela sua arquitetura e cultura que atrai visitante de todo o mundo. Possui uma rica história que remonta à idade Média, além da sua arquitetura Bordéus também é conhecida pela sua gastronomia e vinícola de renome.

Em termos de turismo, Bordéus, é dos destinos mais solicitado em França, sendo mundialmente conhecida pelas suas vinhas circundantes que oferecem uma experiência enológica única aos seus visitantes, a sua arquitetura foi ainda mais publicitada quando em 2007 foi classificada património mundial da UNESCO.

A plataforma AirBNB tem possibilitado a chegada de todo o tipo de turistas, dando a escolher o quanto querem experienciar da cultura da cidade, podendo ficar no coração da cidade ou nas casas rústicas perdidas pelas vinhas.

A economia da cidade, obviamente, gira muito à volta do turismo. Sendo que o ponto alto é alcançado na época do verão, contudo muitos consideram que a primavera é a estação em que Bordéus realmente mostra toda a sua beleza.

Em suma é uma cidade bastante apelativa ao turismo pela sua história, paisagens e arquitetura, além de disponibilizar variados tipos de experiência, os quais possuem ainda mais ênfase na plataforma AirBNB que oferece todos os tipos de alojamento.

Compreensão dos dados

Para começar e com o objetivo de aumentar a perceção sobre a base de dados, procurou-se estudar o significado de cada variável da forma como estavam inicialmente dispostas.

Variável	Descrição
id	Identificador único de cada Airbnb
name	Nome do Airbnb
host_id	Identificador único de cada proprietário de Airbnb's
host_name	Nome do proprietário do Airbnb
neighbourhood_group	Grupo de bairro onde o Airbnb se encontra
neighbourhood	Bairro onde o Airbnb se encontra
latitude	Latitude do Airbnb
longitude	Longitude do Airbnb
room_type	Tipo de quarto ou casa do Airbnb
price	Preço do Airbnb
minimum_nights	Número de noites mínimas de estadia no Airbnb
number_of_reviews	Número de avaliações do Airbnb
last_review	Data da última avaliação do Airbnb
reviews_per_month	Número de avaliações num mês do Airbnb
calculated_host_listings_count	Número de Airbnb's que o proprietário possui
availability_365	Disponibilidade do Airbnb num ano
number_of_reviews_ltm	Número de avaliações num ano do Airbnb
license	Número de licença do Airbnb

Esta compreensão sucede de uma classificação das variáveis.

Qualitativas:

- Name;
- Host_name;
- Neighbourhood;
- Neighbourhood_group;
- Room_type;
- Last_review;

Quantitativas:

- Id;
- Host_id;

- Latitude;
- Longitude;
- Price;
- Minimum_nights;
- Number_of_reviews;
- Reviews_per_month;
- Calculated_host_listing_count;
- Availability_365;
- Number_of_reviews_ltm;

Limpeza/Tratamento dos dados

Após a importação e a leitura da base de dados no Software R procedeu-se à limpeza da mesma, começou-se por sumarizar os dados, para ter uma ideia mais geral dos dados que estão a ser analisados:

```
> summary(dados) # Estatística descritiva básica das variáveis numéricas
       id                  name              host_id            host_name
 Min.   :2.229e+05   Length:11854       Min.   :    30374   Length:11854
 1st Qu.:2.336e+07   Class :character   1st Qu.: 28179296   Class :character
 Median :4.568e+07   Mode  :character   Median : 72691486   Mode  :character
 Mean   :3.182e+17                      Mean   :139894047
 3rd Qu.:7.612e+17                      3rd Qu.:199067229
 Max.   :1.046e+18                      Max.   :550766188

 neighbourhood_group neighbourhood         latitude       longitude
 Length:11854        Length:11854       Min.   :44.75   Min.   :-0.8321
 Class :character    Class :character   1st Qu.:44.82   1st Qu.:-0.6026
 Mode  :character    Mode  :character   Median :44.84   Median :-0.5773
                                        Mean   :44.84   Mean   :-0.5879
                                        3rd Qu.:44.86   3rd Qu.:-0.5651
                                        Max.   :45.02   Max.   :-0.4643

  room_type             price       minimum_nights  number_of_reviews
 Length:11854       Min.   :  13   Min.   :  1.0   Min.   :   0.00
 Class :character   1st Qu.:  55   1st Qu.:  1.0   1st Qu.:   2.00
 Mode  :character   Median :  82   Median :  2.0   Median :   8.00
                    Mean   : 115   Mean   : 43.4   Mean   :  30.69
                    3rd Qu.: 130   3rd Qu.:  5.0   3rd Qu.:  29.00
                    Max.   :5300   Max.   :999.0   Max.   :1837.00
                    NA's   :3692
 last_review        reviews_per_month calculated_host_listings_count
 Length:11854       Min.   : 0.010    Min.   : 1.000
 Class :character   1st Qu.: 0.180    1st Qu.: 1.000
 Mode  :character   Median : 0.600    Median : 1.000
                    Mean   : 1.171    Mean   : 3.485
                    3rd Qu.: 1.510    3rd Qu.: 2.000
                    Max.   :61.920    Max.   :75.000
                    NA's   :1913
 availability_365 number_of_reviews_ltm   license
 Min.   :  0.0    Min.   :  0.000       Length:11854
 1st Qu.:  0.0    1st Qu.:  0.000       Class :character
 Median : 59.0    Median :  2.000       Mode  :character
 Mean   :119.2    Mean   :  8.195
 3rd Qu.:251.0    3rd Qu.:  9.000
 Max.   :365.0    Max.   :521.000
```

Assim como a função str() para identificar a estrutura de cada variável:

```
> str(dados) #Tipos de todas as variáveis/ estrutura da base de dados
```

```
'data.frame':      11854 obs. of  18 variables:
 $ id                            : num  222887 457640 247452 482102 500193 ...
 $ name                          : chr  "Rental unit in Bordeaux · ★4.78 · 2 bedrooms · 3 bed
s · 1 bath" "Rental unit in Talence · ★4.86 · 2 bedrooms · 2 beds · 1 bath" "Rental unit in S
aint-Médard-en-Jalles · ★4.83 · 2 bedrooms · 2 beds · 1 bath" "Townhouse in Le Bouscat · ★4.
77 · 6 bedrooms · 7 beds · 2 baths" ...
 $ host_id                       : int  1156398 2274580 959918 2387430 2468244 1156398 115639
8 1697156 2680968 1847986 ...
 $ host_name                     : chr  "Suzanna" "Christine" "Krista" "Frederic" ...
 $ neighbourhood_group           : chr  "Bordeaux" "Talence" "Saint-Mdard-en-Jalles" "Le Bous
cat" ...
 $ neighbourhood                 : chr  "Bordeaux Sud" "Talence" "Saint-Mdard-en-Jalles" "Le
Bouscat" ...
 $ latitude                      : num  44.8 44.8 44.9 44.9 44.8 ...
 $ longitude                     : num  -0.566 -0.599 -0.727 -0.596 -0.562 ...
 $ room_type                     : chr  "Entire home/apt" "Entire home/apt" "Entire home/apt"
"Entire home/apt" ...
 $ price                         : int  192 120 95 243 100 150 189 81 220 93 ...
 $ minimum_nights                : int  3 3 2 2 2 3 3 1 2 5 ...
 $ number_of_reviews             : int  83 84 65 444 294 53 131 474 78 61 ...
 $ last_review                   : chr  "2023-12-03" "2023-08-31" "2023-10-01" "2023-12-04" .
..
 $ reviews_per_month             : num  0.57 0.61 0.46 3.2 2.15 0.47 0.92 3.33 0.57 0.43 ...
 $ calculated_host_listings_count: int  4 1 1 1 2 4 4 2 1 2 ...
 $ availability_365              : int  281 159 175 0 318 282 219 353 251 294 ...
 $ number_of_reviews_ltm         : int  24 7 10 56 30 13 36 44 6 8 ...
 $ license                       : chr  "3306300031048" "" "" "" ...
```

Valores omissos em “price” e “reviews_per_month”

Denotou-se que existem valores omissos na variável alvo 'price' e optou-se por substituir as ditas observações pelo valor da mediana da variável.

O mesmo processo foi concretizado com a variável “reviews_per_month” de modo a tentar eliminar o menor número de observações possíveis.

Da seguinte forma:

```
mediana <- median(dados$price,na.rm=TRUE)  #Substituir NAs pelo valor da mediana
dados$price[which(is.na(dados$price))] <- mediana

mediana1 <- median(dados$reviews_per_month,na.rm=TRUE)
dados$reviews_per_month[which(is.na(dados$reviews_per_month))] <- mediana1
```

Coluna *Name*

Através dos outputs obtidos relativos às duas funções utilizadas identificou-se de imediato a coluna “name”, pois continha muitos outros atributos escondidos dentro da mesma. O procedimento passou então por replicar o dataset e criar novas colunas de atributos que estavam contidos em “name”, para posteriormente poderem ser analisadas individualmente para o estudo.

```
dados_name <- dados[c(2)] # Coluna name tem 4 elementos
dados_name
dados_name <- cbind(dados_name,  # Adicionar as 5 novas colunas
                    Name = NA,
                    Stars = NA,
                    nbedrooms = NA,
                    nbeds = NA,
                    nbaths = NA)
```

Figura 1 - Inserção de novas colunas

Foi também necessário substituir alguns caracteres desconhecidos que estavam contidos na coluna para proceder mais facilmente à posterior identificação de cada parte do código.

```
dados_name$Name <- sub("·.*", "", dados_name[,1])
```

Figura 2 - Substituição dos caracteres

Retiraram-se os 4 elementos para cada uma das novas colunas:

```
elemento_estrela <- gsub(".*★([^·]*).*", "\\1", dados_name[,1])
elemento_estrela <- ifelse(grepl("★", dados_name[,1]), elemento_estrela, NA)
dados_name$Stars <- elemento_estrela

elemento_quarto <- gsub(".*\\b(\\d+\\s*(?:bedroom|bedrooms))\\b.*|.*\\b(Studio)\\b.*", "\\1\\2", dados_name[,1])
elemento_quarto <- ifelse(grepl("\\b(\\d+\\s*(?:bedroom|Studio|bedrooms))\\b|\\b(Studio)\\b", dados_name[,1]), elemento_quarto, NA)
dados_name$nbedrooms <- elemento_quarto

elemento_cama <- gsub(".*\\b(\\d+\\s*beds?\\b).*", "\\1", dados_name[,1])
elemento_cama <- ifelse(grepl("\\b(\\d+\\s*beds?\\b)", dados_name[,1]), elemento_cama, NA)
dados_name$nbeds<- elemento_cama

ultima_parte <- sub(".* · ([^·]+)$", "\\1", dados_name[,1])
nbaths <- ifelse(grepl("\\b(bath|baths)\\b", ultima_parte), ultima_parte, NA)
dados_name$nbaths <- nbaths
```

Figura 3-- Remoção dos elementos

```
dados2 <- cbind(dados_name,dados,by="name") # Juntar o dataframe inicial com o df mais limpo
dados2 <- subset(dados2, select = -c(1, 8)) # Apagar as colunas chamadas name
```

Figura 4 - Junção de dataframes

Após este processo verificou-se que o que constava na coluna "name" final, era apenas o nome de cada tipo de alojamento seguido do bairro onde se encontrava (algo que também já constava numa outra coluna do dataset). Foi então retirada a informação do bairro da coluna "name" e esta continha agora apenas 27 categorias.

```
> head(dados2$Name)
[1] "Rental unit in Bordeaux "               "Rental unit in Talence "
[3] "Rental unit in Saint-Médard-en-Jalles " "Townhouse in Le Bouscat "
[5] "Rental unit in Bordeaux "               "Rental unit in Bordeaux "
```

Figura 5 - 6 primeiras linhas da coluna Name

```r
# Extrair parte da string até "in" e substituir na coluna "name"
extrair_nome <- function(string) {
  if (grepl("in", string)) {
    return(str_trim(str_extract(string, ".*?\\bin")))
  } else {
    return(string)
  }
}

dados_n2$Name <- sapply(dados_n2$Name, extrair_nome)
extrair_nome2 <- function(string) {
  novo_nome <- gsub("\\sin", "", string)
  return(trimws(novo_nome))
}

# Aplicar a função à coluna "name"
dados_n2$Name <- sapply(dados_n2$Name, extrair_nome2)
str(dados_n2)
```

Figura ~~6~~ - Código utilizado para extrair apenas o tipo de alojamento

Coluna *Stars*

Depois de criada a coluna stars que dava uma avaliação ao alojamento, foram retiradas as designações às colunas que continham "new" como avaliação, assim como posteriormente classificada como variável numérica.

```r
# Remover linhas onde a coluna "Stars" é "new"
dados_n2 <- dados_n2[dados_n2$Stars != "New", ]
dados_n2 <- subset(dados_n2, Stars != "")

# Agora converter a coluna "Stars" para numérica
dados_n2$Stars <- as.numeric(dados_n2$Stars)
```

Figura 7 - Alterações na coluna Stars

Coluna *nbeds* e *nbedrooms*

Para estas duas variáveis, foram também depois transformadas em numéricas. Para nbedrooms, para o valor "studio" foi considerado como 0 quartos, e substituído então pelo valor 0.

```
dados_n2$nbeds <- as.integer(gsub("[^0-9]", "", dados_n2$nbeds)) #transformar variável "nbeds" em int
dados_n2$nbedrooms <- ifelse(dados_n2$nbedrooms == "Studio", 0, dados_n2$nbedrooms) #Substituir Studio por "0"
dados_n2$nbedrooms <- as.integer(gsub("[^0-9]", "", dados_n2$nbedrooms)) #Transformar em INT
```

Figura 8 - Alterações nas colunas nbeds e nbedrooms

Coluna *room_type*

Para a coluna room_type, foi do entendimento do grupo que seria benéfico esta variável categórica ser traduzida para uma variável categórica, mas codificada numericamente, para ser possível ser utilizada num modelo futuro pois tinha apenas 4 possíveis categorias, como se pode verificar na figura 10 (tabela de frequências).

```
table(dados_n2$room_type) #Tabela de frequência de acordo com o tipo de quarto
# Criação de variável númerica para room_type
dados_n2$room_type_cod = ifelse(dados_n2$room_type == 'Entire home/apt', 1,
                                ifelse(dados_n2$room_type == 'Hotel room', 2,
                                       ifelse(dados_n2$room_type == 'Private room', 3, 4)))
```

Figura 9 - Código de tabela de frequências e alterações na variável room_type

```
> table(dados_n2$room_type) #Tabela de frequência de acordo com o tipo de quarto

Entire home/apt      Hotel room    Private room     Shared room
           6734              14            1824              27
```

Figura 10 - Tabela de frequências de tipo de quarto

Coluna *nbaths*

Um processo semelhante ao usado na variável anterior foi utilizado nesta, sendo que a única distinção ocorre na conversão da variável para numérica, já que esta ocorreu numa nova coluna. O principal objetivo é reduzir o número de categorias da variável aglomerando em apenas nove, ou seja, apenas os valores inteiros (0, 1, 2, 3, 4, 5, 6, 7 e 8) o número de casas de banho, com o intuito de ter uma melhor interpretação e análise da variável.

```
dados_n2$nbaths <- gsub("0 shared baths|0 baths", "0", dados_n2$nbaths) #substitui os valores "0 shared baths" por "0 baths"
dados_n2$baths_int <- dados_n2$nbaths
dados_n2$baths_int <- gsub("Half-bath|Private half-bath|Shared half-bath|1 private bath|1 shared bath|1 bath", "1", dados_n2$baths_int
dados_n2$baths_int <- gsub("1.5 baths|1.5 shared baths|2 shared baths|2 baths", "2", dados_n2$baths_int)
dados_n2$baths_int <- gsub("2.5 baths|2.5 shared baths|3 shared baths|3 baths", "3", dados_n2$baths_int)
dados_n2$baths_int <- gsub("3.5 baths|4 shared baths|4 baths", "4", dados_n2$baths_int)
dados_n2$baths_int <- gsub("4.5 baths|5 baths", "5", dados_n2$baths_int)
dados_n2$baths_int <- gsub("5.5 baths|6 baths|5.5", "6", dados_n2$baths_int)
dados_n2$baths_int <- gsub("7 baths", "7", dados_n2$baths_int)
dados_n2$baths_int <- gsub("7.5 baths|8 baths|7.5", "8", dados_n2$baths_int)
dados_n2$baths_int <- as.integer(dados_n2$baths_int)
```

Figura 11 - Alterações na coluna nbaths

Valores omissos após tratamento dos dados

Posteriormente ao tratamento dos dados, e da obtenção de uma base de dados com todo o processo de tratamento anteriormente explicado, decidiu-se remover as observações que tinham campos omissos, da seguinte maneira:

```
> dados_n2 <- na.omit(dados2) #Sem valores omissos
> nrow(dados_n2)
[1] 8599
> nrow(dados2) - nrow(dados_n2)
[1] 3255
```

Figura 12 - Nº de linhas após remover NAs

Resultaram assim 8599 linhas após o tratamento final de dados, sendo eliminadas 3255 linhas.

Estudo das Variáveis

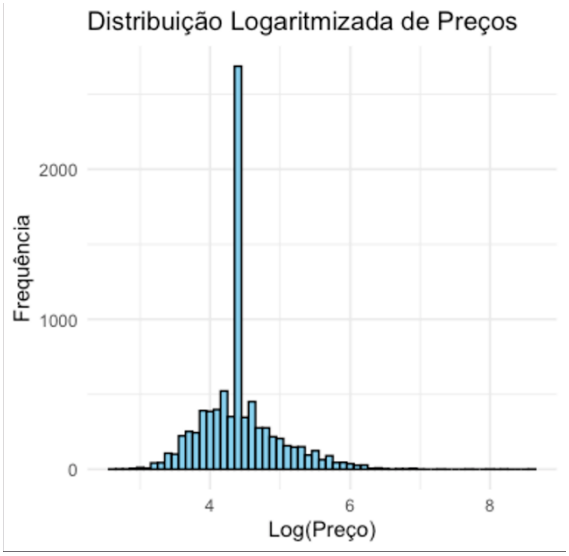
Estatísticas da variável target

Com a ajuda da biblioteca ‘ggplot2’, plotaram-se dois gráficos (um com os valores logaritmizados outro não), para ganhar uma noção da distribuição dos valores da variável 'price’ das distintas observações em relação às suas respetivas frequências, obtendo o seguinte gráfico logaritmizado:

```
# Criar o histograma usando ggplot2
dados_n2$log_price <- log(dados_n2$price)
ggplot(dados_n2, aes(x = log_price)) +
  geom_histogram(binwidth = 0.1, fill = "skyblue", color = "black") +
  labs(title = "Distribuição Logaritmizada de Preços", x = "Log(Preço)", y =
         "Frequência") + theme_minimal()
```

Figura 13 - Código de construção do histograma

Output:



Como se verifica a variável em estudo contém alguns “outliers”, sendo as inúmeras barras pequenas pouco visíveis localizadas mais à direita os valores mais extremos.

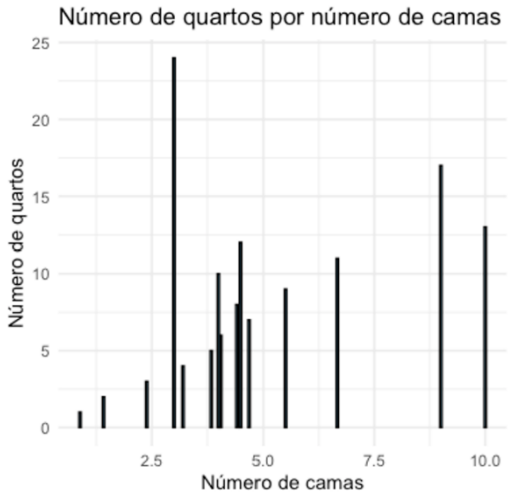
Gráficos entre as variáveis

Para obter uma ideia mais ampla de como uma variável podia ser disposta em função de outra(s) da base de dados, decidiu-se visualizar diversos gráficos de barras (com biblioteca ggplot) em que se relacionaram as várias variáveis da base de dados entre elas (duas a duas), com o intuito de visualizar a distribuição das observações de uma variável ‘x’ em função dos valores da variável ‘y’, como por exemplo:

```
#Comparar a variável nbeds e nbedrooms
dados_agrupados <- aggregate(nbedrooms ~ nbeds, data = dados_n2, FUN = mean)
ggplot(dados_agrupados, aes(x = nbedrooms, y = nbeds)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(title = "Número de quartos por número de camas", x = "Número de camas",
       y = "Número de quartos") + theme_minimal()
```

Figura 143 - Código de construção do gráfico de barras

Output:



Pode-se concluir que à medida que o número de quartos aumenta, o número de camas segue uma distribuição linearmente dependente, o que já era de esperar mais quartos por alojamento/Hotel a tendência geral é ter mais camas também, permite também identificar alguns “outliers” como é o caso neste exemplo da maior barra, em que o número de quartos aumenta exponencialmente para um número de camas ainda reduzido de aproximadamente 3 camas, o que não é comum.

Correlação entre as variáveis

Com o objetivo ainda de estudar as variáveis presentes na base de dados, criou-se uma matriz de correlação de Pearson entre todos os pares de variáveis apenas entre as variáveis

numéricas, o que é importante estudar para evitar multicolinearidade entre as variáveis explicativas e ao mesmo tempo utilizar as que melhor representam a variável de estudo.

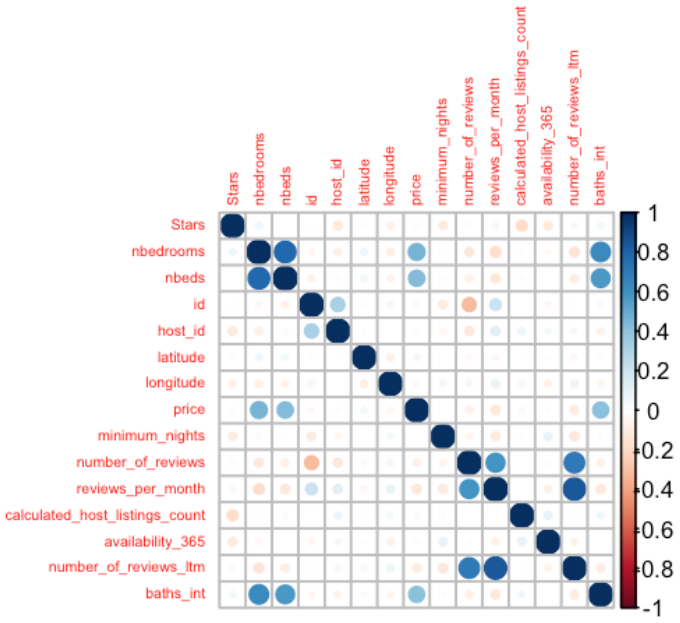


Figura 15 - Matriz de correlação

Pode-se concluir então que as variáveis mais correlacionadas com a variável alvo são 'baths_int', 'nbedrooms' e 'nbeds' positivamente, estas seriam possivelmente três variáveis a incluir no modelo de previsão. No entanto, esta mesma matriz permite verificar que a correlação entre 'nbedrooms' e 'nbeds' é superior a 0.7, assim como entre 'number_of_reviews' e 'number_of_reviews_ltm' e 'reviews_per_month' com 'number_of_reviews_ltm'. Logo, entre cada um destes 3 pares de variáveis, nenhum poderia coexistir nos modelos criados pelo que as variáveis 'nbeds' e 'number_of_reviews_ltm' foram excluídas.

Foi depois verificada uma representação gráfica dos pares de variáveis através do comando pairs()

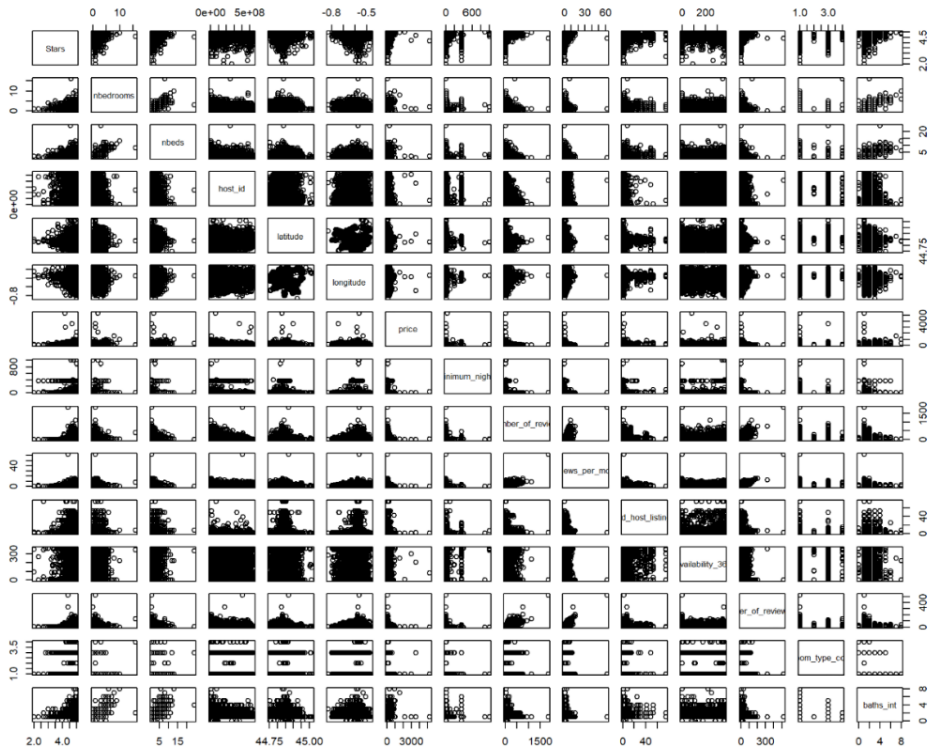


Figura 46 - Representação gráfica de pares de variáveis

Ou seja, pelas representações gráficas existentes e relativas a 'price' verifica-se que existem ambas relações lineares como não lineares para os conjuntos de variáveis.

Relação entre as variáveis explicativas e a target

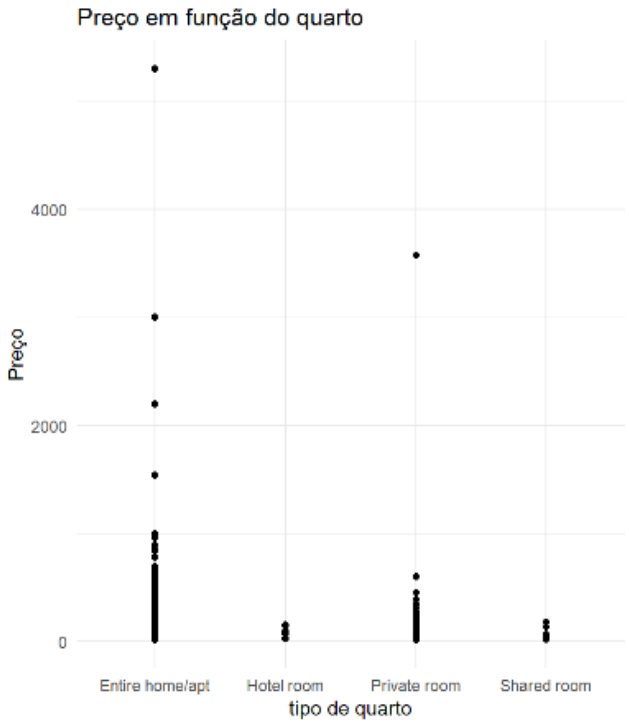
Com o objetivo de estudar melhor as variáveis explicativas, e perceber quais as mais indicadas para trabalhar, isto é que maior impacto e explicabilidade exercem sobre a variável alvo, efetuaram-se diversas análises.

Para isso reproduziram-se tabelas de valores médios de 'price' e gráficos de barras.

```
room_type     Price
1 Entire home/apt 122.63262
2      Hotel room  85.92857
3    Private room  57.03250
4     Shared room  46.31818
```

Obteve-se então a seguinte tabela que demonstra a grande variabilidade no valor médio do preço em relação ao tipo de quarto correspondente, demonstrada também pelo seguinte gráfico.

Output:



Criação do modelo de regressão

Com o objetivo de otimizar os modelos criados, foi utilizada a biblioteca olsrr, que devido às suas funções, ols_step_both_p e ols_step_both_aic, permitiam, com base no valor de p-value e no critério de informação Akaike, respetivamente, verificar quais as melhores opções para o modelo de regressão.

Stepwise Summary

Variable	AIC	SBC	SBIC	R2	Adj. R2
Base Model	100772.710	100786.739	77442.035	0.00000	0.00000
nbedrooms (+)	99333.545	99354.588	76003.152	0.16080	0.16070
baths_int (+)	99104.840	99132.897	75774.513	0.18402	0.18382
room_type_cod (+)	99005.830	99040.902	75675.555	0.19398	0.19369
availability_365 (+)	98931.347	98973.434	75601.143	0.20145	0.20106
reviews_per_month (+)	98919.390	98968.491	75589.203	0.20280	0.20232
latitude (+)	98917.517	98973.632	75587.337	0.20318	0.20260
host_id (+)	98916.608	98979.738	75586.434	0.20346	0.20278

Figura 17 - Sumário da função ols_step_both_p

Stepwise Summary

Variable	AIC	SBC	SBIC	R2	Adj. R2
Base Model	100772.710	100786.739	77442.035	0.00000	0.00000
nbedrooms (+)	99333.545	99354.588	76003.152	0.16080	0.16070
baths_int (+)	99104.840	99132.897	75774.513	0.18402	0.18382
room_type_cod (+)	99005.830	99040.902	75675.555	0.19398	0.19369
availability_365 (+)	98931.347	98973.434	75601.143	0.20145	0.20106
reviews_per_month (+)	98919.390	98968.491	75589.203	0.20280	0.20232
latitude (+)	98917.517	98973.632	75587.337	0.20318	0.20260
host_id (+)	98916.608	98979.738	75586.434	0.20346	0.20278

Figura 58 - Sumário da função ols_step_both_aic

Verifica-se então total concordância entre as duas funções, o que implica que as variáveis que são estatisticamente significativas (ou seja, têm p-values baixos) também levam a modelos com AIC mais baixos. Isso pode acontecer quando as variáveis adicionadas ao modelo contribuem significativamente para a explicação da variância na variável de resposta e não aumentam muito a complexidade do modelo.

Para a criação do modelo, propriamente dita, foi primeiramente alterada a target para o logaritmo(price), o que resultou no modelo seguinte:

```
modelo1 <- lm(log(price) ~ nbedrooms + baths_int + room_type_cod + host_id
              + reviews_per_month + availability_365 + latitude, data =
                amostra_final1)
```

Figura 69 - Criação do modelo1

De modo a melhorar o primeiro modelo, para a obtenção do modelo seguinte, modelo2 foram retirados os outliers da variável dependente (price), passando a amostra utilizada a ser a seguinte:

```
(outliers_price <- boxplot(amostra_final1$price, plot = FALSE)$out)
amostra_final2 <- amostra_final1[!amostra_final1$price %in% outliers_price, ]
```

Figura 20 - Exclusão de outliers da amostra_final2

Assim verificamos as diferenças entre um diagrama de bigodes antes e depois da exclusão dos outliers:

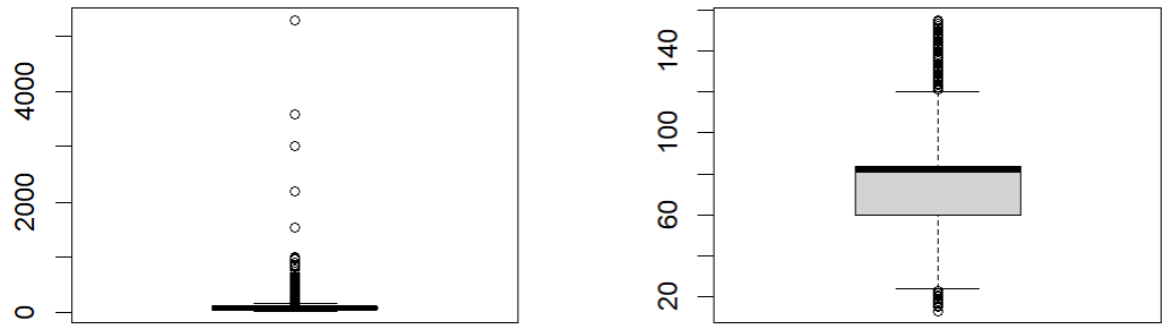


Figura 21 - Boxplot para a variável price antes e depois da exclusão dos outliers

Comparação Modelo1 e Modelo2

Após a criação dos dois primeiros modelos, foi necessário realizar uma avaliação dos mesmos segundo 3 critérios

```
     n_modelo  MAPE1_2     AICs R_quadrado
[1,]        1 31.91183 8574.561  0.4456668
[2,]        2 24.57146 3420.607  0.3149679
```

Figura 22 - Avaliação dos modelos

Perante estes resultados verifica-se que depois da remoção de outliers o valor do MAPE (mean absolute percentage error) é menor (a diferença entre os valores é superior a 7 pontos percentuais), o valor do critério do critério de informação Akaike também melhora significativamente, no entanto, apenas é explicada aproximadamente 31.5% da variabilidade da variável target no modelo2 em comparação com os 44.6% de variabilidade explicada de price no modelo1.

Conclui-se então que apesar da perda de variabilidade, na decisão entre estes dois modelos, a escolha seria o modelo2.

Não linearidade

Para verificar se existia não linearidade nas variáveis e resíduos, foi utilizada a função crPlots(), que resultou nos seguintes gráficos:

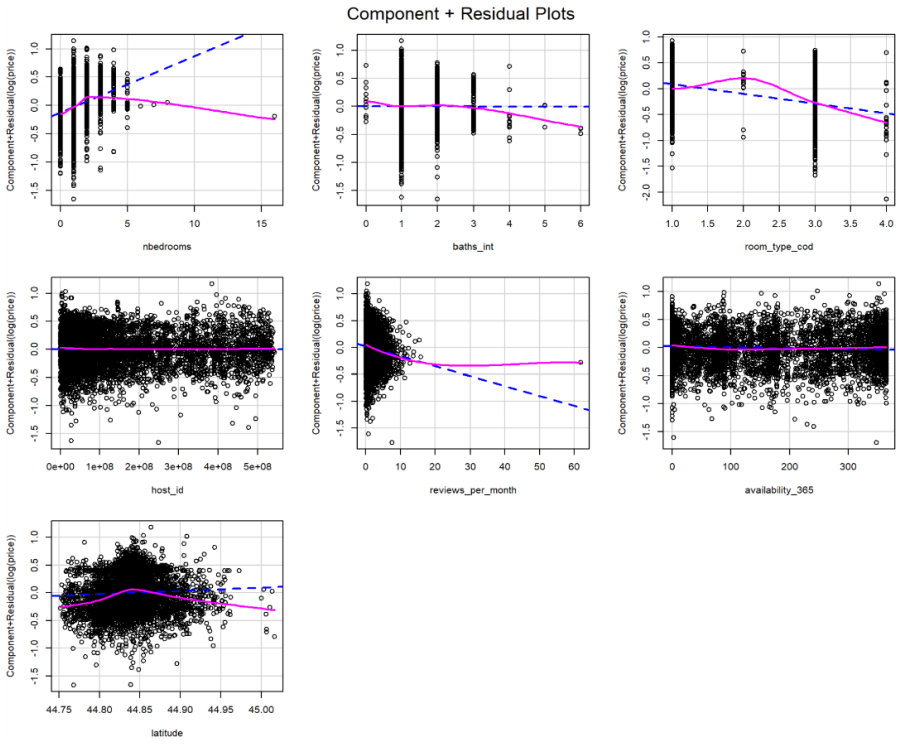


Figura 73 – Gráficos de Resíduos

Finalmente e devido às relações não lineares verificadas, procurou-se acrescentar não linearidade ao modelo e nas respetivas variáveis. Para room_type_cod e latitude segundo um polinómio de grau 2 e para nbedrooms, bath_int, reviews_per_month segundo um polinómio de grau 3, resultando assim num modelo do tipo:

```
modelo3 <- lm(log(price) ~ poly(nbedrooms,degree=3) + poly(baths_int,degree = 3) + poly(room_type_cod,degree=2)
              + host_id
              + poly(reviews_per_month,degree=3) + availability_365 + poly(latitude,degree=2), data =
                amostra_final2)
```

Figura 84 - Criação do modelo3

Comparação e escolha final de modelos

Perante a criação do modelo3, foi necessária nova avaliação de modelos e posterior escolha do modelo preferencial.

```
     n_modelo1   erros1    AICs1 R_quadrado1
[1,]         1 31.91183 8574.561   0.4456668
[2,]         2 24.57146 3420.607   0.3149679
[3,]         3 23.83074 3000.795   0.3551701
```

Figura 95 - Avaliação de modelos

Depois da polinomização das variáveis ocorre nova redução no valor de MAPE, acompanhado de nova melhoria do critério AIC, assim como um aumento da variabilidade do target em relação ao modelo2, o que determina que o modelo a escolher será o modelo3.

Interpretação e análise final

```
Call:
lm(formula = log(price) ~ poly(nbedrooms, degree = 3) + poly(baths_int,
    degree = 3) + poly(room_type_cod, degree = 2) + host_id +
    poly(reviews_per_month, degree = 3) + availability_365 +
    poly(latitude, degree = 2), data = amostra_final2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.46187 -0.17501 -0.00975  0.19952  1.18914

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                        4.294e+00  5.597e-03 767.158  < 2e-16 ***
poly(nbedrooms, degree = 3)1       7.565e+00  3.559e-01  21.256  < 2e-16 ***
poly(nbedrooms, degree = 3)2      -3.273e+00  3.045e-01 -10.749  < 2e-16 ***
poly(nbedrooms, degree = 3)3       1.882e+00  3.203e-01   5.876 4.39e-09 ***
poly(baths_int, degree = 3)1       7.656e-01  3.555e-01   2.154 0.031303 *
poly(baths_int, degree = 3)2      -5.875e-01  3.035e-01  -1.936 0.052919 .
poly(baths_int, degree = 3)3      -4.328e-01  2.991e-01  -1.447 0.147869
poly(room_type_cod, degree = 2)1  -1.389e+01  3.068e-01 -45.266  < 2e-16 ***
poly(room_type_cod, degree = 2)2  -3.708e-01  3.034e-01  -1.222 0.221689
host_id                            2.343e-11  2.460e-11   0.953 0.340872
poly(reviews_per_month, degree = 3)1 -3.009e+00  3.062e-01  -9.825  < 2e-16 ***
poly(reviews_per_month, degree = 3)2  1.112e+00  2.999e-01   3.707 0.000211 ***
poly(reviews_per_month, degree = 3)3 -9.523e-01  3.029e-01  -3.145 0.001670 **
availability_365                   -1.032e-04  2.707e-05  -3.814 0.000138 ***
poly(latitude, degree = 2)1        1.387e+00  2.986e-01   4.644 3.47e-06 ***
poly(latitude, degree = 2)2       -4.754e+00  3.019e-01 -15.746  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2976 on 7190 degrees of freedom
Multiple R-squared:  0.3552,    Adjusted R-squared:  0.3538
F-statistic:   264 on 15 and 7190 DF,  p-value: < 2.2e-16
```

Figura 106 - Sumário do modelo3

Intercept (4.294): Este é o valor esperado do target quando todas as variáveis são 0.

Os coeficientes para um aumento unitário de nbedrooms, baths_int, reviews_per_month e latitude não são claros e variam consoante o grau do polinómio, não apresentando linearidade

Room_type_cod: Os coeficientes para room_type_cod indicam a mudança esperada na variável dependente para cada aumento unitário em room_type_cod, mantendo todas as outras variáveis constantes. Host_id (2.343e-11): Para cada aumento unitário em host_id, espera-se que a variável dependente mude por 2.343e-11, mantendo todas as outras variáveis constantes. Availability_365 (-1.032e-04): Para cada aumento unitário em availability_365, espera-se que a variável dependente diminua por 1.032e-04, mantendo todas as outras variáveis constantes.

As variáveis baths_int (grau3), room_type_cod(grau2) e host_id apresentam valores que permitem concluir que não são estatísticamente significativas.

Um RSE de 0.2976 indica que a variação típica entre a resposta observada e a resposta prevista pelo modelo é de aproximadamente 0.2976. A estatística F de 264 e o p-valor menor que 2.2e-16 indicam que o modelo como um todo é significativo. O R2 múltiplo de 0.3552 indica que aproximadamente 35.52% da variância na variável dependente é explicada pelo modelo.

Verificação de pressupostos

O primeiro pressuposto, média nula verifica-se, já que esta é muito reduzida.

```
> mean(modelo3$residuals)
[1] -1.458098e-17
```

O segundo pressuposto, variância constante é dado pelo teste Breusch-Pagan. Neste a hipótese nula (erros homocedásticos) é rejeitada pois o valor p-value associado é inferior a 0.05. O que permite concluir que os erros são heterocedásticos.

```
> bptest(modelo3)

        studentized Breusch-Pagan test

data:  modelo3
BP = 596.52, df = 15, p-value < 2.2e-16
```

O terceiro pressuposto, ausência de correlação é dado pelo teste Breusch-Godfrey. H0 corresponde a uma independência de resíduos, que também não é verificada. Tal como o pressuposto anterior, um p-value inferior a 0.05 implica uma rejeição de H0 e consequente dependência de resíduos.

```
> bgtest(modelo3)

        Breusch-Godfrey test for serial correlation of order up to 1

data:  modelo3
LM test = 8.0408, df = 1, p-value = 0.004574
```

Por fim, o quarto pressuposto (dos resíduos normalmente distribuídos) é dado pelo teste Jarque Bera. Como hipótese nula tem-se que os resíduos têm distribuição normal, o que para o modelo3 também não é verificado.

```
> jarque.bera.test(modelo3$residuals)

        Jarque Bera Test

data:  modelo3$residuals
X-squared = 171.96, df = 2, p-value < 2.2e-16
```

Finalmente foi obtida a representação gráfica dos resíduos do modelo:

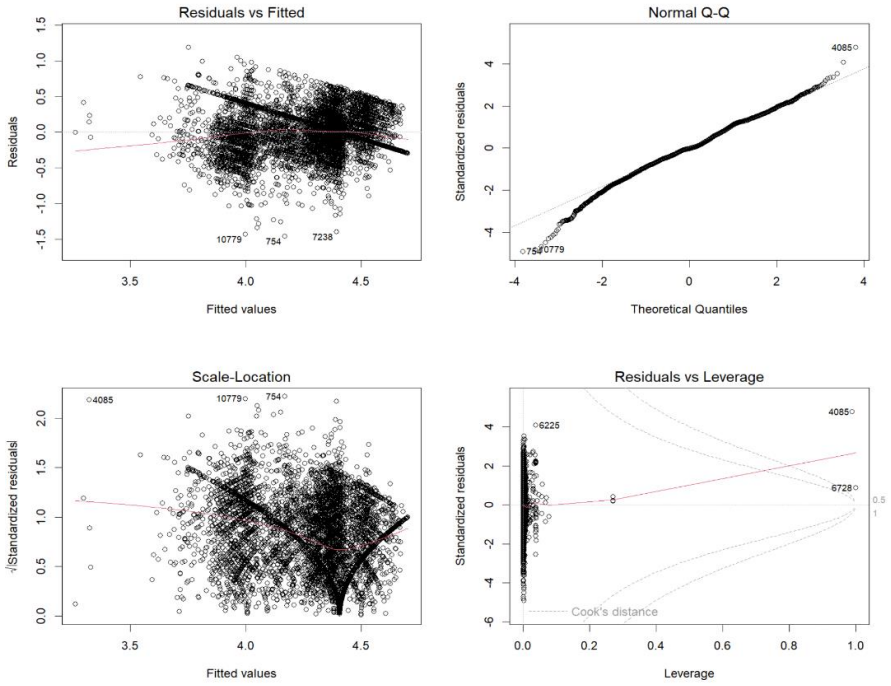


Figura 118 – Representação gráfica dos resíduos em relação ao modelo3

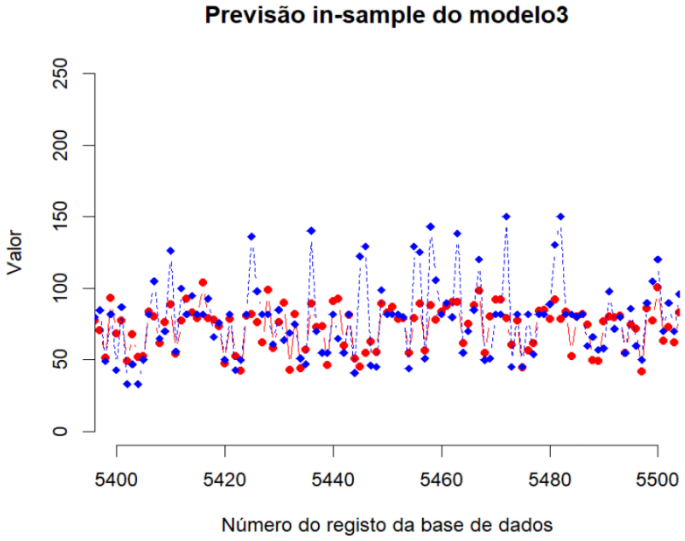
- Resíduos vs Ajustados: Verifica-se uma aleatoriedade nos pontos sem curvatura distinta ou funil, o que indicaria não linearidade ou heterocedasticidade.
- Normal Q-Q: Os pontos seguem aproximadamente a linha diagonal.
- Escala-Localização: Parece haver funil e aleatoriedade nos pontos
- Resíduos vs Alavancagem: Não parece haver outliers

Previsão

Previsão in-sample

O modelo3 foi então o modelo mais ajustado para fazer a previsão pelo seu menor valor de erro (MAPE) de apenas 23,83%.

Output:



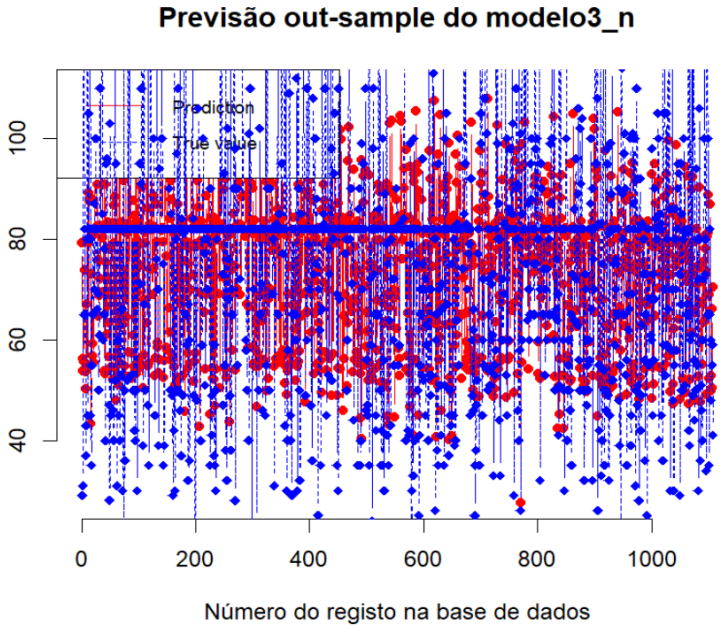
Previsão out-sample

Foi decidido fazer também uma previsão out-sample para perceber o comportamento do modelo fora da amostra inicial. Foi então dividida em conjuntos de treino e teste (ponderação 90/10 em percentagem).

Foi depois treinado o modelo sobre o conjunto de treino e obtida a seguinte previsão com os valores de treino a vermelho e os valores reais a azul. Finalmente foi obtido o respetivo valor de erro para a previsão out-sample que era de aproximadamente 25,17%.

Como o valor de MAPE é superior ao valor inicial pode indiciar um sobre ajustamento do modelo ou overfitting.

Output:



Subamostra

Devido à impossibilidade de verificação dos pressupostos dos resíduos nos modelos criados foi decidido criar uma subamostra de menor dimensão, usando o modelo3, que provava ser o de menor valor de MAPE.

Esta subamostra contém apenas as 300 primeiras linhas da amostra correspondente aos modelos "modelo2" e "modelo3".

```
amostra_final3<-amostra_final2[1:300,]
```

Verificação dos pressupostos (subamostra)

O primeiro pressuposto (média nula) é verificado.

```
> mean(modelo4$residuals) #Media nula
[1] -5.912516e-19
```

O segundo pressuposto (variância constante) não é verificado (p-value menor que 0.05) pelo que os erros são heterocedásticos.

```
> bptest(modelo4) #variância constante

        studentized Breusch-Pagan test

data:  modelo4
BP = 30.121, df = 15, p-value = 0.01149
```

O terceiro pressuposto (independência dos resíduos) é verificado pelo que se denota ausência de correlação e resíduos independentes (p-value maior que 0.05).

```
> bgtest(modelo4) #ausência de correlação

        Breusch-Godfrey test for serial correlation of order up to 1

data:  modelo4
LM test = 0.13945, df = 1, p-value = 0.7088
```

O último pressuposto (distribuição normal dos resíduos) é também verificado (p-value maior que 0.05).

```
> jarque.bera.test(modelo4$residuals) #resíduos normalmente distribuídos

        Jarque Bera Test

data:  modelo4$residuals
X-squared = 4.9346, df = 2, p-value = 0.08481
```

Conclusão

Após a análise das variáveis chega-se à conclusão de que as variáveis que mais influenciam a variável target são nbedrooms, nbeds e baths_int, de forma positiva, ou seja, quando uma das variáveis aumenta, o target price também aumenta. No entanto, as variáveis number_of_reviews, reviews_per_month e number_of_reviews_ltm estão também relacionadas com a variável price, enquanto que de forma negativa e em menor escala.

Tivemos de abdicar de algumas variáveis da base de dados, para o modelo devido a sua elevada correlação em relação a outras, não sendo possível ter em conta todas as características de cada alojamento/Hotel, mas de forma geral obtivemos um modelo bastante preciso.

No que diz respeito aos modelos, foi possível concluir que o modelo mais correto seria o modelo3 uma vez que existe uma redução no valor de MAPE, uma melhoria no critério AIC e um valor intermédio da variabilidade da variável target em relação aos outros modelos.

Consideramos fazer uma previsão in-sample e out-sample, em que observamos um valor ligeiramente superior do MAPE (dois pontos percentuais) na previsão fora da amostra, que sugere uma possível sobreposição do modelo.

Referências Bibliográficas

Explore France. Bordeaux. (2018). https://www.france.fr/pt/bordeaux ;

Inside Airbnb Adding data to the debate. *Data Downloads*. (2023). http://insideairbnb.com/get-the-data/ .