ProyectoRairbnbMachineLearning

Alejandro Rios Silva

2024-09-15

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

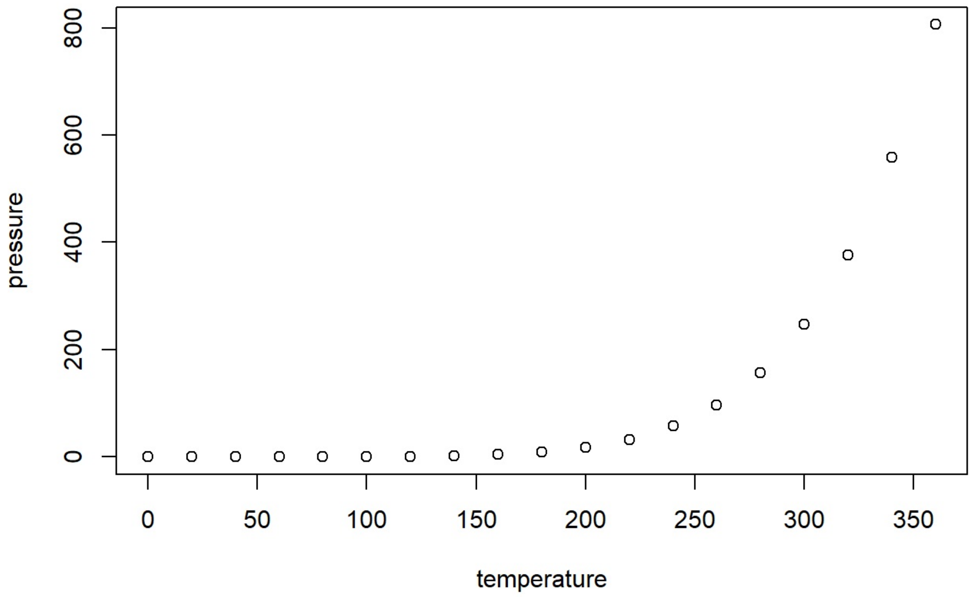
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
airbnb <- read.csv('airbnb-listings.csv', sep = ';')
options(repr.plot.height=4, repr.plot.width=6, repr.plot.res = 300)
```

Me quedo con las columnas que más interesante me parecen

```
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   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ 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
```

```
df_madrid <- airbnb[airbnb$City == "Madrid" & airbnb$Room.Type == "Entire home/apt" & airbnb$Neighbourhood != "",
]
df_madrid <- df_madrid[, c("Neighbourhood", "Accommodates", "Bathrooms", "Bedrooms", "Beds", "Price", "Square.Fee
t", "Guests.Included", "Extra.People", "Review.Scores.Rating", "Latitude", "Longitude")]

print(head(df_madrid, 10))
```

```
##          Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet
## 26             Almagro            4         1        1    2    60          NA
## 27             Almagro            4         2        1    1   141          NA
## 30             Almagro            7         3        4    4   230          NA
## 32          Rios Rosas            5         1        2    3    88          NA
## 34 Fuencarral-el Pardo            5         1        2    2    65          NA
## 40           Argüelles            6         2        4    6    78          NA
## 44              Aluche            6         1        2    4    48          NA
## 54         Carabanchel            4         1        2    2    69          NA
## 56         Carabanchel            4         1        1    1    27          NA
## 66          Gaztambide            5         3        3    4   150          NA
##    Guests.Included Extra.People Review.Scores.Rating Latitude Longitude
## 26               3           10                   99 40.43768 -3.699259
## 27               2           15                   87 40.43640 -3.692044
## 30               5           30                   93 40.42807 -3.694460
## 32               2           25                   77 40.43934 -3.698665
## 34               4           10                   NA 40.47981 -3.725272
## 40               1            0                   NA 40.43158 -3.718951
## 44               4            5                  100 40.41438 -3.727246
## 54               3           15                   93 40.39680 -3.713495
## 56               2            6                   91 40.39701 -3.711650
## 66               1            0                   80 40.43729 -3.716256
```

Paso de pies cuadrados a metros cuadrados para poder hacer los cálculos más adelante

```
df_madrid$Square.Meters <- df_madrid$Square.Feet * 0.092903

print(head(df_madrid, 10))
```

```
##          Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet
## 26             Almagro            4         1        1    2    60          NA
## 27             Almagro            4         2        1    1   141          NA
## 30             Almagro            7         3        4    4   230          NA
## 32          Rios Rosas            5         1        2    3    88          NA
## 34 Fuencarral-el Pardo            5         1        2    2    65          NA
## 40           Argüelles            6         2        4    6    78          NA
## 44              Aluche            6         1        2    4    48          NA
## 54         Carabanchel            4         1        2    2    69          NA
## 56         Carabanchel            4         1        1    1    27          NA
## 66          Gaztambide            5         3        3    4   150          NA
##    Guests.Included Extra.People Review.Scores.Rating Latitude Longitude
## 26               3           10                   99 40.43768 -3.699259
## 27               2           15                   87 40.43640 -3.692044
## 30               5           30                   93 40.42807 -3.694460
## 32               2           25                   77 40.43934 -3.698665
## 34               4           10                   NA 40.47981 -3.725272
## 40               1            0                   NA 40.43158 -3.718951
## 44               4            5                  100 40.41438 -3.727246
## 54               3           15                   93 40.39680 -3.713495
## 56               2            6                   91 40.39701 -3.711650
## 66               1            0                   80 40.43729 -3.716256
##    Square.Meters
## 26            NA
## 27            NA
## 30            NA
## 32            NA
## 34            NA
## 40            NA
## 44            NA
## 54            NA
## 56            NA
## 66            NA
```

Miro qué porcentaje de pisos no tienen los metros cuadrados puestos

```
sum(is.na(df_madrid$Square.Meters))
```

```
## [1] 5254
```

```
percentage_na <- df_madrid |> summarize(percentage_na = mean(is.na(Square.Meters)) * 100)
print(percentage_na)
```

```
##   percentage_na
## 1      93.80468
```

Miro qué porcentaje de pisos tienen 0 metros cuadrados

```
length(which(df_madrid$Square.Meters == 0))
```

```
## [1] 128
```

```
df_madrid$Square.Meters[df_madrid$Square.Meters == 0] <- NA

print(head(df_madrid, 10))
```

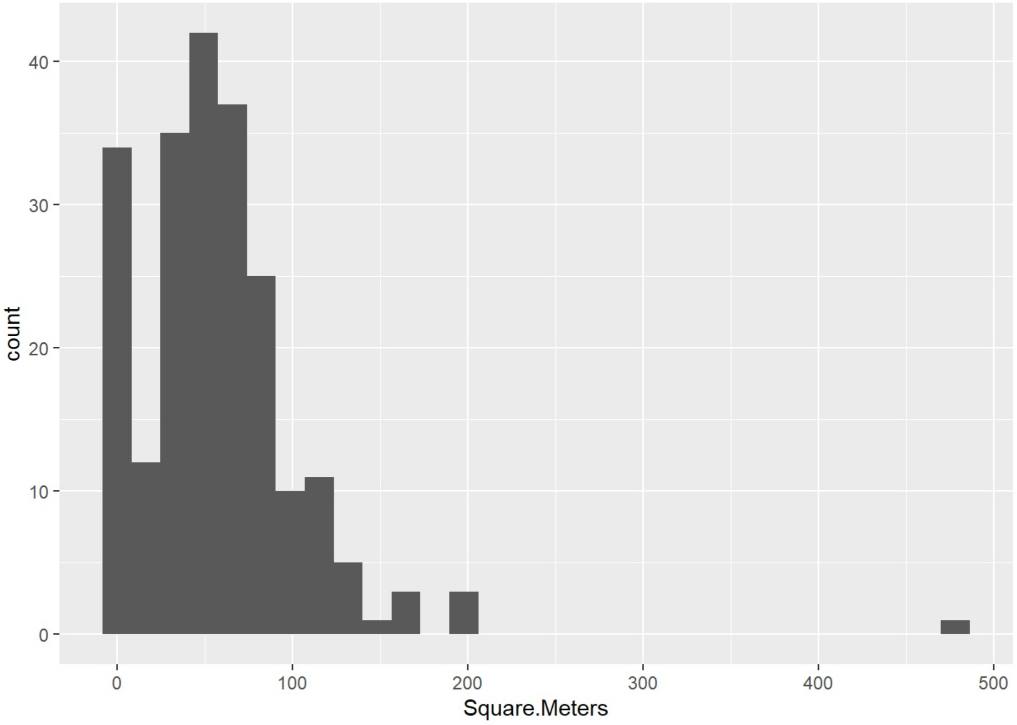
```
##          Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet
## 26            Almagro            4         1        1    2    60          NA
## 27            Almagro            4         2        1    1   141          NA
## 30            Almagro            7         3        4    4   230          NA
## 32         Rios Rosas            5         1        2    3    88          NA
## 34 Fuencarral-el Pardo            5         1        2    2    65          NA
## 40          Argüelles            6         2        4    6    78          NA
## 44             Aluche            6         1        2    4    48          NA
## 54        Carabanchel            4         1        2    2    69          NA
## 56        Carabanchel            4         1        1    1    27          NA
## 66         Gaztambide            5         3        3    4   150          NA
##    Guests.Included Extra.People Review.Scores.Rating Latitude Longitude
## 26               3           10                   99 40.43768 -3.699259
## 27               2           15                   87 40.43640 -3.692044
## 30               5           30                   93 40.42807 -3.694460
## 32               2           25                   77 40.43934 -3.698665
## 34               4           10                   NA 40.47981 -3.725272
## 40               1            0                   NA 40.43158 -3.718951
## 44               4            5                  100 40.41438 -3.727246
## 54               3           15                   93 40.39680 -3.713495
## 56               2            6                   91 40.39701 -3.711650
## 66               1            0                   80 40.43729 -3.716256
##    Square.Meters
## 26            NA
## 27            NA
## 30            NA
## 32            NA
## 34            NA
## 40            NA
## 44            NA
## 54            NA
## 56            NA
## 66            NA
```

Pinto el histograma de los metros cuadrados para ver si tengo que filtrar algún elemento más

```
library(ggplot2)

ggplot(df_madrid, aes(x = Square.Meters)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 5382 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



Asigno el valor NA a la columna Square.Meters de los apartamentos que tengan menos de 20 m^2

```
df_madrid$Square.Meters[df_madrid$Square.Meters <= 20] <- NA

print(head(df_madrid, 10))
```

```
##         Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet
## 26            Almagro            4         1        1    2    60          NA
## 27            Almagro            4         2        1    1   141          NA
## 30            Almagro            7         3        4    4   230          NA
## 32         Rios Rosas            5         1        2    3    88          NA
## 34 Fuencarral-el Pardo            5         1        2    2    65          NA
## 40          Argüelles            6         2        4    6    78          NA
## 44             Aluche            6         1        2    4    48          NA
## 54        Carabanchel            4         1        2    2    69          NA
## 56        Carabanchel            4         1        1    1    27          NA
## 66         Gaztambide            5         3        3    4   150          NA
##    Guests.Included Extra.People Review.Scores.Rating Latitude Longitude
## 26               3           10                   99 40.43768 -3.699259
## 27               2           15                   87 40.43640 -3.692044
## 30               5           30                   93 40.42807 -3.694460
## 32               2           25                   77 40.43934 -3.698665
## 34               4           10                   NA 40.47981 -3.725272
## 40               1            0                   NA 40.43158 -3.718951
## 44               4            5                  100 40.41438 -3.727246
## 54               3           15                   93 40.39680 -3.713495
## 56               2            6                   91 40.39701 -3.711650
## 66               1            0                   80 40.43729 -3.716256
##    Square.Meters
## 26            NA
## 27            NA
## 30            NA
## 32            NA
## 34            NA
## 40            NA
## 44            NA
## 54            NA
## 56            NA
## 66            NA
```

Existen varios barrios donde todas las entradas de Square.Meters son NA, vamos a eliminar del dataset todos los pisos que pertenecen a estos barrios.

```
library(dplyr)

df_num_na <- df_madrid |> group_by(Neighbourhood) |> summarise(num_NA = sum(is.na(Square.Meters)), num_total = n(
))
barrios_na_completos <- df_num_na |> filter(num_NA == num_total) |> pull(Neighbourhood)
df_madrid <- df_madrid |> filter(!Neighbourhood %in% barrios_na_completos)

print(head(df_madrid, 10))
```

```
##    Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet
## 1        Almagro            4         1        1    2    60          NA
## 2        Almagro            4         2        1    1   141          NA
## 3        Almagro            7         3        4    4   230          NA
## 4     Rios Rosas            5         1        2    3    88          NA
## 5      Argüelles            6         2        4    6    78          NA
## 6    Carabanchel            4         1        2    2    69          NA
## 7    Carabanchel            4         1        1    1    27          NA
## 8      Argüelles            4         2        2    2   100          NA
## 9      Argüelles            3         2        2    2   130          NA
## 10 Ciudad Lineal            5         1        3    4    50          NA
##    Guests.Included Extra.People Review.Scores.Rating Latitude Longitude
## 1                3           10                   99 40.43768 -3.699259
## 2                2           15                   87 40.43640 -3.692044
## 3                5           30                   93 40.42807 -3.694460
## 4                2           25                   77 40.43934 -3.698665
## 5                1            0                   NA 40.43158 -3.718951
## 6                3           15                   93 40.39680 -3.713495
## 7                2            6                   91 40.39701 -3.711650
## 8                4           20                   97 40.42264 -3.717986
## 9                3           30                   NA 40.42948 -3.722911
## 10               1            0                   89 40.42726 -3.654208
##    Square.Meters
## 1             NA
## 2             NA
## 3             NA
## 4             NA
## 5             NA
## 6             NA
## 7             NA
## 8             NA
## 9             NA
## 10            NA
```

Compruebo si todos los barrios tienen los mismos metros cuadrados de media

```
test_saphiro <- shapiro.test(df_madrid$Square.Meters)
print(test_saphiro)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  df_madrid$Square.Meters
## W = 0.66594, p-value < 2.2e-16
```

```
test_anova <- summary(aov(Square.Meters ~ Neighbourhood, data = df_madrid))
print(test_anova)
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)    
## Neighbourhood  37 167320    4522   2.986 2.21e-06 ***
## Residuals     136 205991    1515                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 4727 observations deleted due to missingness
```

Agrupo los barrios por metros cuadrados usando una matriz de similaridad de Tukey, mostrando cuán similares o diferentes son los barrios.

```
tky <- TukeyHSD(aov(Square.Meters ~ Neighbourhood, data = df_madrid))
tky.result <- data.frame(tky$Neighbourhood)
cn <- sort(unique(df_madrid$Neighbourhood))
resm <- matrix(NA, length(cn), length(cn))
rownames(resm) <- cn
colnames(resm) <- cn
resm[lower.tri(resm)] <- round(tky.result$p.adj, 4)
resm[upper.tri(resm)] <- t(resm)[upper.tri(resm)]
diag(resm) <- 1
```

En el punto anterior he creado una matriz de p-valores que indica cuán parecidos son dos barrios. Si el p-valor es alto, significa que los barrios son diferentes; si es bajo, significa que los barrios se parecen. Esta matriz la podemos usar como matriz de distancia si restamos el p-valor a 1. Es decir, si usamos como distancia 1 - p-valor. De esta forma, barrios con un p-valor alto tendrán una distancia mayor que aquellos con un p-valor bajo. Voy a crear una nueva columna en el dataframe con un nuevo identificador marcado por los clusters obtenidos.

```
resm.dist <- as.dist(1 - abs(resm))
str(resm.dist)
```

```
##  'dist' num [1:703] 0 0 0 0 0 ...
##  - attr(*, "Labels")= chr [1:38] "Acacias" "Adelfas" "Almagro" "Almenara" ...
##  - attr(*, "Size")= int 38
##  - attr(*, "call")= language as.dist.default(m = 1 - abs(resm))
##  - attr(*, "Diag")= logi FALSE
##  - attr(*, "Upper")= logi FALSE
```

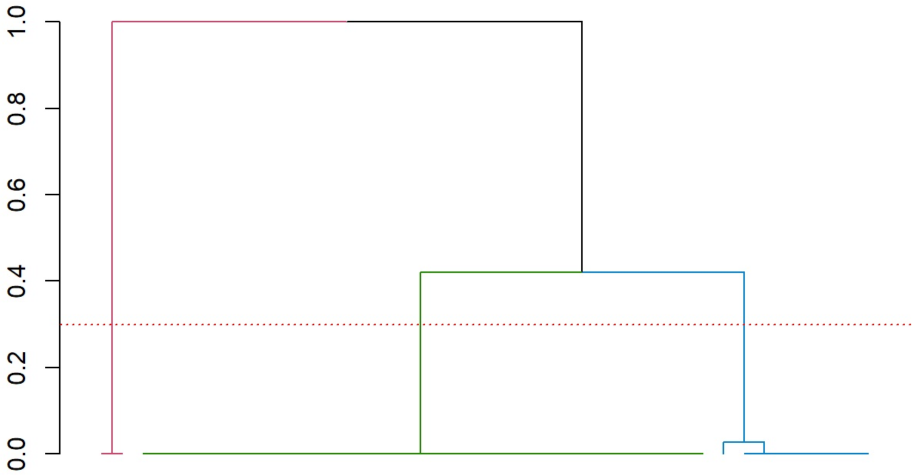
```
resm.tree <- hclust(resm.dist, method = "complete")
resm.dend <- as.dendrogram(resm.tree)

library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.17.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

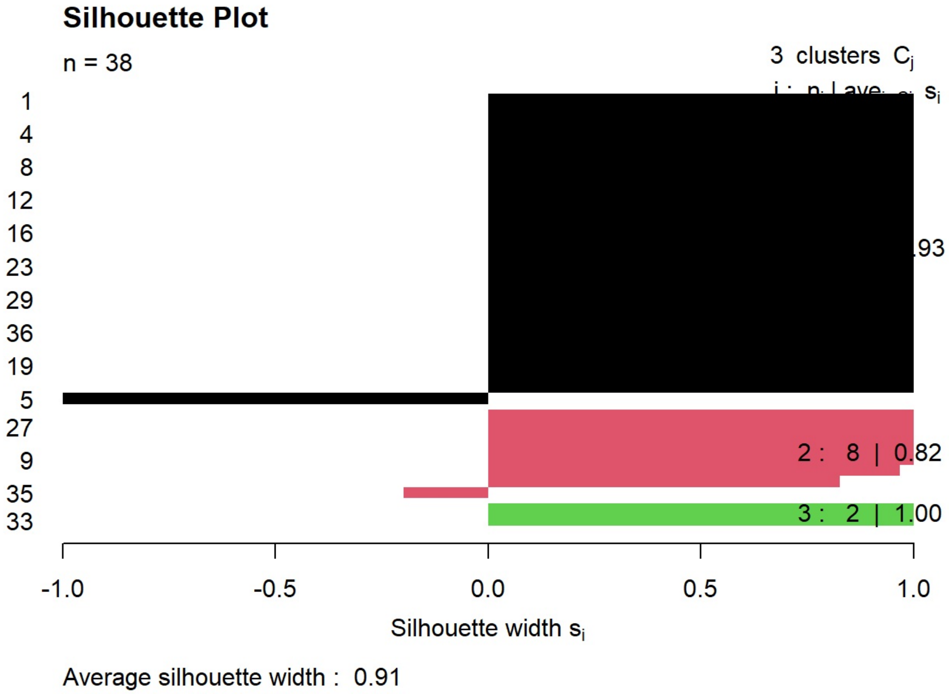
```
##
## Adjuntando el paquete: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
clusters <- cutree(resm.dend, h = 0.3)
plot(color_branches(resm.dend, h = 0.3), leaflab = "none")
abline(h = 0.3, col = "red", lty = 3)
```



```r
library(cluster)
ss <- silhouette(clusters, resm.dist)
plot(ss, col = 1:max(clusters), border = NA, main = "Silhouette Plot")
```



```r
df_clusters <- data.frame(Neighbourhood = names(clusters), Cluster = clusters)
df_madrid <- merge(df_madrid, df_clusters, by = "Neighbourhood")
names(df_madrid)[names(df_madrid) == "Cluster"] <- "neighb_id"

print(head(df_madrid, 10))
```

```
##    Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet
## 1        Acacias            2       0.5        0    2    30          NA
## 2        Acacias            2       1.0        1    1    65          NA
## 3        Acacias            6       2.0        3    4   100          NA
## 4        Acacias            5       2.0        2    2   120          NA
## 5        Acacias            3       1.0        1    1   122          NA
## 6        Acacias            6       1.0        2    3    50          NA
## 7        Acacias            2       1.0        1    1    75          NA
## 8        Acacias            3       1.0        1    2    45          NA
## 9        Acacias            2       1.0        1    1    68          NA
## 10       Acacias            2       1.0        0    1    39          NA
##    Guests.Included Extra.People Review.Scores.Rating Latitude Longitude
## 1                2            0                   81 40.40351 -3.703586
## 2                1            0                  100 40.40233 -3.705738
## 3                1            0                   NA 40.40265 -3.702798
## 4                4           20                   95 40.40519 -3.706163
## 5                1            0                   NA 40.39957 -3.702361
## 6                2           10                   68 40.40226 -3.712753
## 7                1            0                  100 40.40460 -3.708392
## 8                1            0                   NA 40.40093 -3.703781
## 9                1            0                   94 40.40452 -3.707737
## 10               1            0                  100 40.40094 -3.702806
##    Square.Meters neighb_id
## 1             NA         1
## 2             NA         1
## 3             NA         1
## 4             NA         1
## 5             NA         1
## 6             NA         1
## 7             NA         1
## 8             NA         1
## 9             NA         1
## 10            NA         1
```

Voy a crear dos grupos, uno test y otro train.

```
train_proportion <- 0.7
train_index <- sample(seq_len(nrow(df_madrid)), size = train_proportion * nrow(df_madrid))

train_df_madrid <- df_madrid[train_index, ]
test_df_madrid <- df_madrid[-train_index, ]

print(head(train_df_madrid, 10))
```

```
##        Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet
## 4134      Rios Rosas            5         1        2    4    55          NA
## 780           Cortes            4         1        0    2    82          NA
## 1413     Embajadores            4         1        1    2    60          NA
## 3952 Palos do Moguer            4         1        2    2    50          NA
## 3520        Malasaña            4         1        2    3   140          NA
## 2943        Malasaña            3         1        0    2    42          NA
## 3850         Palacio            2         1        0    1    77          NA
## 1200     Embajadores            6         1        2    2    50          NA
## 307        Argüelles            4         1        1    1    55          75
## 4783       Trafalgar            6        NA        2    3   195           0
##      Guests.Included Extra.People Review.Scores.Rating Latitude Longitude
## 4134               4           10                   NA 40.43938 -3.697175
## 780                2           15                   97 40.41406 -3.696324
## 1413               2           10                   72 40.41090 -3.701033
## 3952               2            5                   94 40.40553 -3.699355
## 3520               1            0                  100 40.42643 -3.703507
## 2943               1            0                   89 40.42072 -3.701573
## 3850               1            0                   80 40.41639 -3.710309
## 1200               1            0                   87 40.40639 -3.699787
## 307                1           12                   87 40.43103 -3.724586
## 4783               6           35                   40 40.43153 -3.700622
##      Square.Meters neighb_id
## 4134            NA         3
## 780             NA         1
## 1413            NA         1
## 3952            NA         1
## 3520            NA         1
## 2943            NA         1
## 3850            NA         1
## 1200            NA         1
## 307             NA         1
## 4783            NA         1
```

```
print(head(test_df_madrid, 10))
```

```
##    Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet
## 3        Acacias            6         2        3    4   100          NA
## 4        Acacias            5         2        2    2   120          NA
## 6        Acacias            6         1        2    3    50          NA
## 8        Acacias            3         1        1    2    45          NA
## 9        Acacias            2         1        1    1    68          NA
## 10       Acacias            2         1        0    1    39          NA
## 12       Acacias            4         1        1    2    60         538
## 19       Acacias            4         1        1    2    59          NA
## 21       Acacias            4         1        2    2    74          NA
## 25       Acacias            2         2        1    1    68          NA
##    Guests.Included Extra.People Review.Scores.Rating Latitude Longitude
## 3                1            0                   NA 40.40265 -3.702798
## 4                4           20                   95 40.40519 -3.706163
## 6                2           10                   68 40.40226 -3.712753
## 8                1            0                   NA 40.40093 -3.703781
## 9                1            0                   94 40.40452 -3.707737
## 10               1            0                  100 40.40094 -3.702806
## 12               2           15                   98 40.40513 -3.707726
## 19               2           10                   95 40.39933 -3.701477
## 21               3           15                  100 40.39801 -3.702725
## 25               1            0                   NA 40.40176 -3.700929
##    Square.Meters neighb_id
## 3             NA         1
## 4             NA         1
## 6             NA         1
## 8             NA         1
## 9             NA         1
## 10            NA         1
## 12      49.98181         1
## 19            NA         1
## 21            NA         1
## 25            NA         1
```

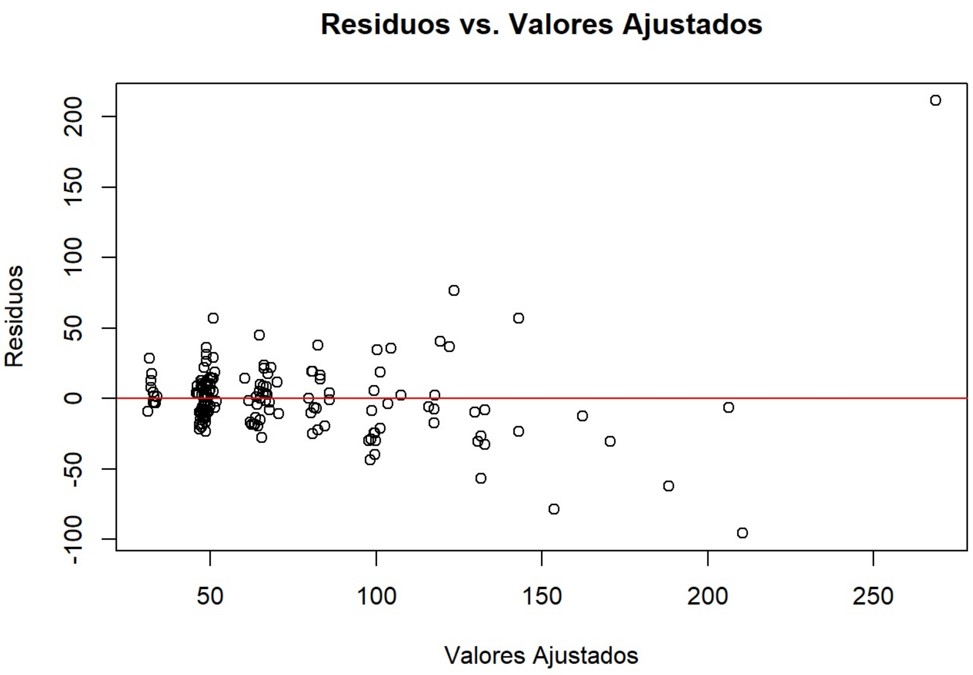
Paso a predecir los metros cuadrados en función del resto de columnas del dataframe.

```
df_madrid_filtrado <- df_madrid |> select(-Neighbourhood)
formula <- as.formula("Square.Meters ~ Bathrooms + Price + Bedrooms")
model <- lm(formula, data = df_madrid_filtrado)
summary(model)
```

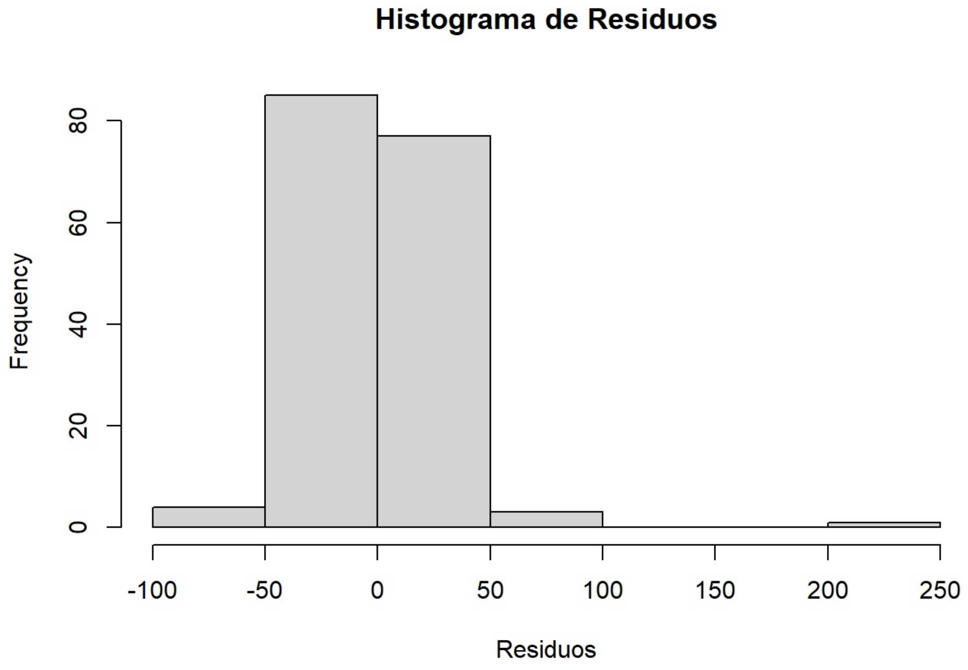
```
##
## Call:
## lm(formula = formula, data = df_madrid_filtrado)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -95.48 -10.46  -1.57  10.26 211.37
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.81918    4.88478  -1.191   0.2352
## Bathrooms   33.79246    4.70345   7.185 2.17e-11 ***
## Price        0.07779    0.03286   2.367   0.0191 *
## Bedrooms    15.42482    2.86979   5.375 2.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.26 on 166 degrees of freedom
##   (4731 observations deleted due to missingness)
## Multiple R-squared:  0.6599, Adjusted R-squared:  0.6537
## F-statistic: 107.4 on 3 and 166 DF,  p-value: < 2.2e-16
```

Evaluo la calidad del modelo

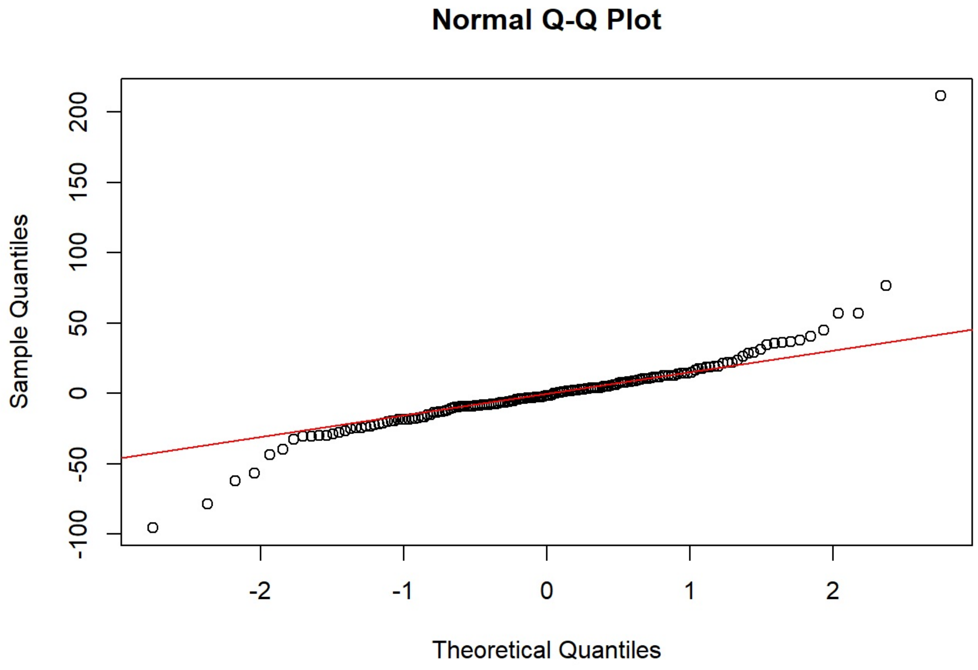
```
# Diagnóstico de los residuos
plot(model$fitted.values, model$residuals, xlab = "Valores Ajustados", ylab = "Residuos", main = "Residuos vs. Va
lores Ajustados")
abline(h = 0, col = "red")
```



```
hist(model$residuals, xlab = "Residuos", main = "Histograma de Residuos")
```



```
qqnorm(model$residuals)
qqline(model$residuals, col = "red")
```



```
# Medidas de ajuste del modelo
predicciones <- predict(model, newdata = df_madrid_filtrado)
errores <- predicciones - df_madrid_filtrado$Square.Meters
mse <- mean(errores^2)
rmse <- sqrt(mse)
mae <- mean(abs(errores))

print(paste("MSE:", mse))
```

```
## [1] "MSE: NA"
```

```
print(paste("RMSE:", rmse))
```

```
## [1] "RMSE: NA"
```

```
print(paste("MAE:", mae))
```

```
## [1] "MAE: NA"
```

```
r_squared <- summary(model)$r.squared
print(paste("R-squared:", r_squared))
```

```
## [1] "R-squared: 0.65987141382097"
```

Si tuviéramos un anuncio de un apartamento para 6 personas (Accommodates), con 1 baño, un precio de 80€/noche y 3 habitaciones en el barrio de Sol, con 3 camas y un review de 80, ¿cuántos metros cuadrados tendría? Vamos a probar cómo funciona el modelo con el ejemplo.

```
predict(model, data.frame(Bathrooms = 1, Price = 50, Bedrooms = 3))
```

```
##        1
## 78.13733
```

FIN.