ProyectoRairbnbMachineLearning

Francisco Clemente Fernández

2024-09-01

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.

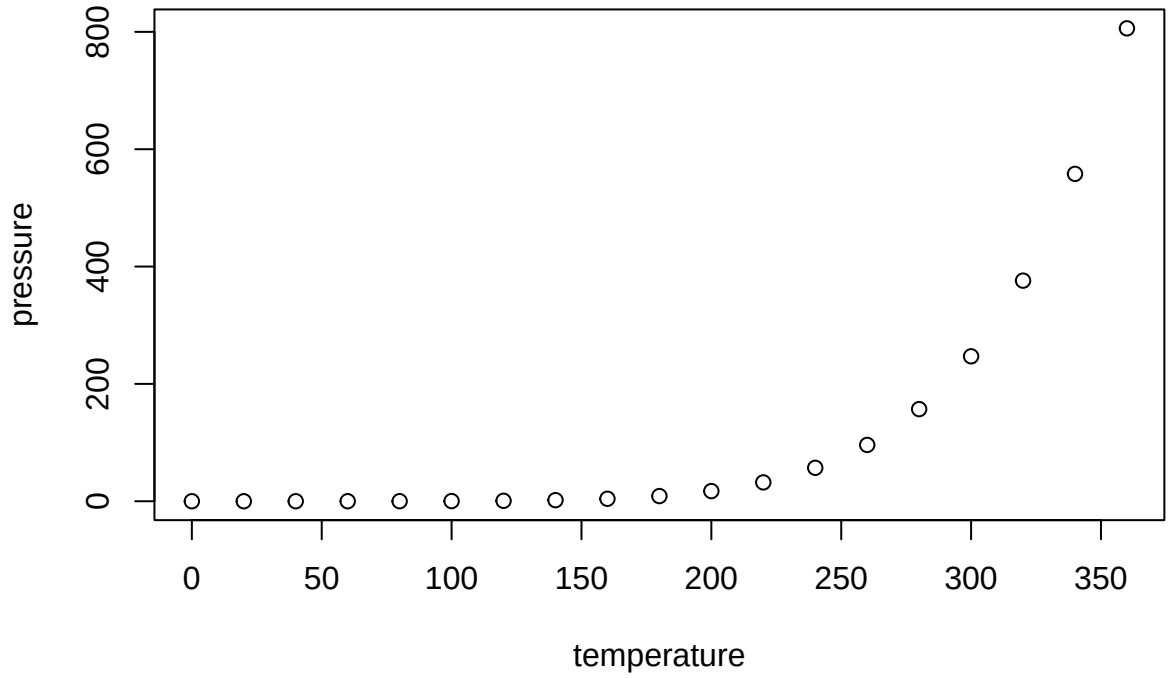
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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)
```

Selecciono las columnas con la información más importante

```
library(tidyverse)

df_madrid <- airbnb[airbnb$City == "Madrid" & airbnb$Room.Type == "Entire home/apt" & airbnb$Neighbourh
df_madrid <- df_madrid[, c("Neighbourhood", "Accommodates", "Bathrooms", "Bedrooms", "Beds", "Price", "

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

```
##          Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet Guests.Included Extra.
## 26             Almagro            4         1        1    2    60          NA               3
## 27             Almagro            4         2        1    1   141          NA               2
## 30             Almagro            7         3        4    4   230          NA               5
## 32          Rios Rosas            5         1        2    3    88          NA               2
## 34 Fuencarral-el Pardo            5         1        2    2    65          NA               4
## 40           Argüelles            6         2        4    6    78          NA               1
## 44              Aluche            6         1        2    4    48          NA               4
## 54         Carabanchel            4         1        2    2    69          NA               3
## 56         Carabanchel            4         1        1    1    27          NA               2
## 66          Gaztambide            5         3        3    4   150          NA               1
```

Realizo una conversión 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 Guests.Included Extra.
## 26             Almagro            4         1        1    2    60          NA               3
## 27             Almagro            4         2        1    1   141          NA               2
## 30             Almagro            7         3        4    4   230          NA               5
## 32          Rios Rosas            5         1        2    3    88          NA               2
## 34 Fuencarral-el Pardo            5         1        2    2    65          NA               4
## 40           Argüelles            6         2        4    6    78          NA               1
## 44              Aluche            6         1        2    4    48          NA               4
## 54         Carabanchel            4         1        2    2    69          NA               3
## 56         Carabanchel            4         1        1    1    27          NA               2
## 66          Gaztambide            5         3        3    4   150          NA               1
```

Miro qué porcentaje de pisos no tienen la información de metros cuadrados

```
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 Guests.Included Extra.
## 26             Almagro            4         1        1    2    60          NA               3
## 27             Almagro            4         2        1    1   141          NA               2
## 30             Almagro            7         3        4    4   230          NA               5
## 32          Rios Rosas            5         1        2    3    88          NA               2
## 34 Fuencarral-el Pardo            5         1        2    2    65          NA               4
## 40           Argüelles            6         2        4    6    78          NA               1
## 44              Aluche            6         1        2    4    48          NA               4
## 54         Carabanchel            4         1        2    2    69          NA               3
## 56         Carabanchel            4         1        1    1    27          NA               2
## 66          Gaztambide            5         3        3    4   150          NA               1
```

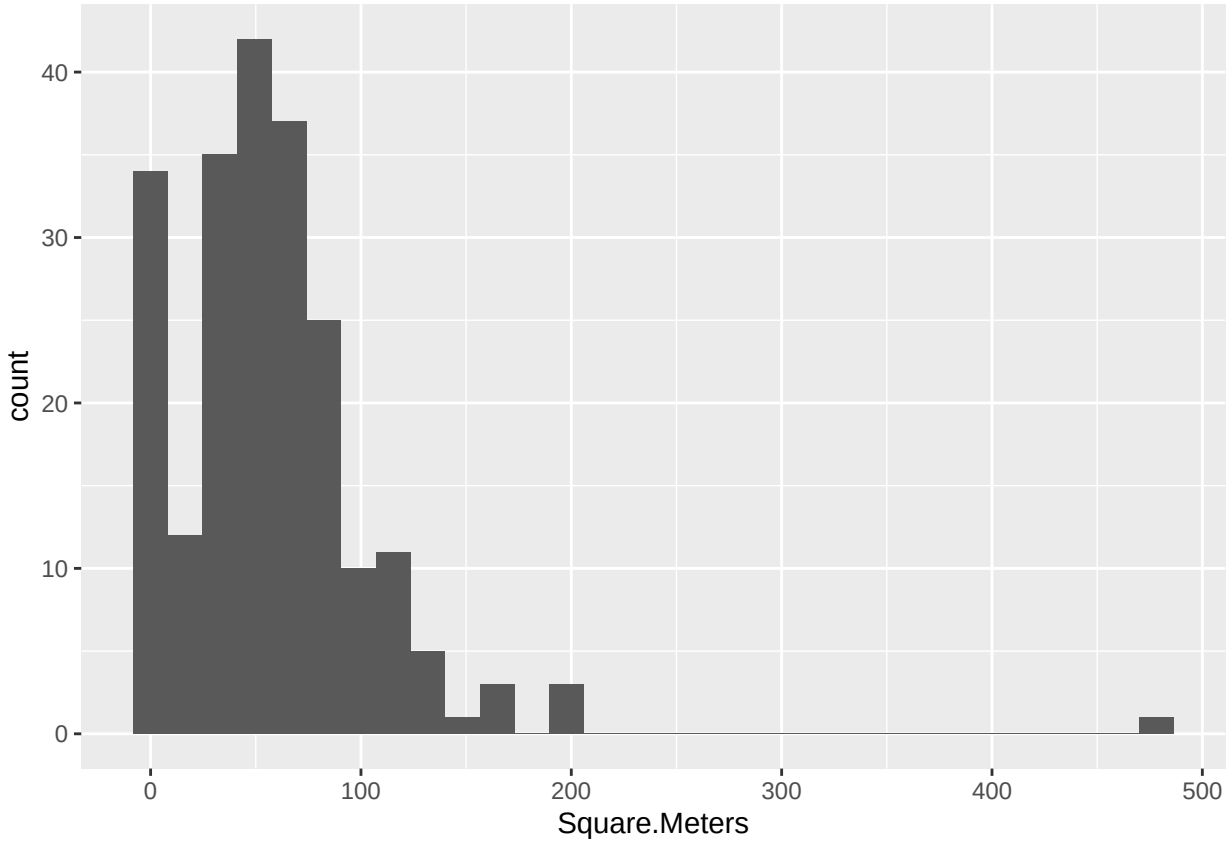
Pinto el histograma de los metros cuadrados para ver si tengo que filtrar algún elemento más

```
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 Guests.Included Extra.
## 26             Almagro            4         1        1    2    60          NA               3
## 27             Almagro            4         2        1    1   141          NA               2
## 30             Almagro            7         3        4    4   230          NA               5
## 32          Rios Rosas            5         1        2    3    88          NA               2
## 34 Fuencarral-el Pardo            5         1        2    2    65          NA               4
## 40           Argüelles            6         2        4    6    78          NA               1
```

```
## 44              Aluche            6         1         2    4    48          NA               4
## 54         Carabanchel            4         1         2    2    69          NA               3
## 56         Carabanchel            4         1         1    1    27          NA               2
## 66          Gaztambide            5         3         3    4   150          NA               1
```

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_
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 Guests.Included Extra.People
## 1         Almagro            4         1        1    2    60          NA               3           10
## 2         Almagro            4         2        1    1   141          NA               2           15
## 3         Almagro            7         3        4    4   230          NA               5           30
## 4      Rios Rosas            5         1        2    3    88          NA               2           25
## 5       Argüelles            6         2        4    6    78          NA               1            0
## 6     Carabanchel            4         1        2    2    69          NA               3           15
## 7     Carabanchel            4         1        1    1    27          NA               2            6
## 8       Argüelles            4         2        2    2   100          NA               4           20
## 9       Argüelles            3         2        2    2   130          NA               3           30
## 10 Ciudad Lineal             5         1        3    4    50          NA               1            0
```

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 las similitudes y diferencias de los barrios.

```r
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 que 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.

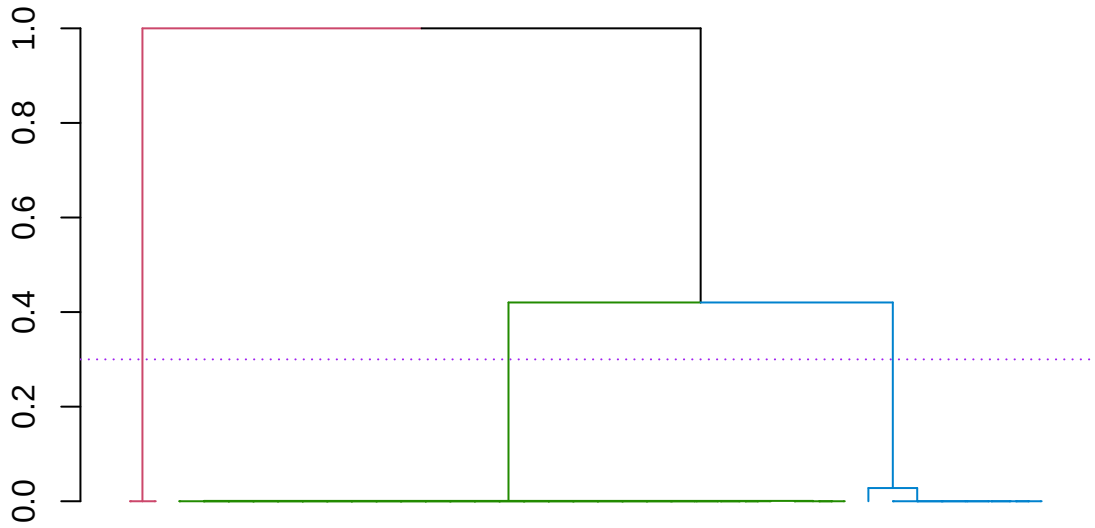
```r
resm.dist <- as.dist(1 - abs(resm))
str(resm.dist)
```

```
##  '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
```

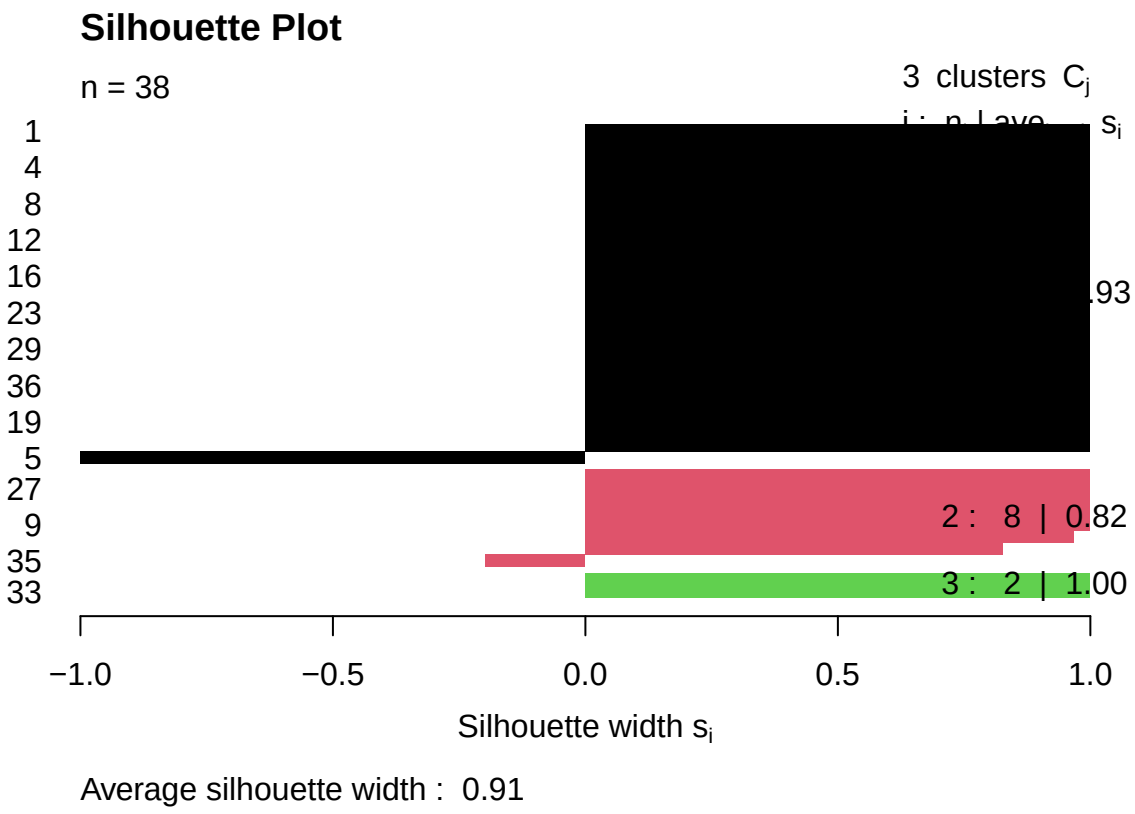
```r
resm.tree <- hclust(resm.dist, method = "complete")
resm.dend <- as.dendrogram(resm.tree)

library(dendextend)

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




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




```
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 Guests.Included Extra.People
## 1         Acacias            2       0.5        0    2    30          NA               2            0
## 2         Acacias            2       1.0        1    1    65          NA               1            0
## 3         Acacias            6       2.0        3    4   100          NA               1            0
## 4         Acacias            5       2.0        2    2   120          NA               4           20
## 5         Acacias            3       1.0        1    1   122          NA               1            0
## 6         Acacias            6       1.0        2    3    50          NA               2           10
## 7         Acacias            2       1.0        1    1    75          NA               1            0
## 8         Acacias            3       1.0        1    2    45          NA               1            0
## 9         Acacias            2       1.0        1    1    68          NA               1            0
## 10        Acacias            2       1.0        0    1    39          NA               1            0
```

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, ]
```

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

```
##      Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet Guests.Included Extra.Peopl
## 970         Cortes            2         1        1    1   100          NA               1
## 3085      Malasaña            3         1        2    2    55          NA               1
## 4158      San Blas            3         1        1    2    27          NA               1
## 493       Castilla            2         1        0    1    45          NA               1
## 2130      Justicia            6         2        3    3   290          NA               1
## 4261           Sol            6         2        2    2    81          NA               1
## 4402           Sol            2         1        1    1    65          NA               2
## 2705     La Latina            6         2        2    3    59          NA               1
## 1805          Goya            2         1        1    1   120          NA               1
## 2297     La Latina            3         1        1    2    70         592               2
```

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

```
##    Neighbourhood Accommodates Bathrooms Bedrooms Beds Price Square.Feet Guests.Included Extra.People
## 1        Acacias            2       0.5        0    2    30          NA               2            0
## 7        Acacias            2       1.0        1    1    75          NA               1            0
## 9        Acacias            2       1.0        1    1    68          NA               1            0
## 13       Acacias            8       3.0        4    7   140          NA               6           20
## 18       Acacias            4       1.0        2    4    70           0               4           15
## 19       Acacias            4       1.0        1    2    59          NA               2           10
## 23       Acacias            2       1.0        0    1    22          NA               1            5
## 25       Acacias            2       2.0        1    1    68          NA               1            0
## 26       Acacias            4       1.0        3    3    60          NA               2           10
## 32       Acacias            4       1.0        2    2    45          NA               2           25
```

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

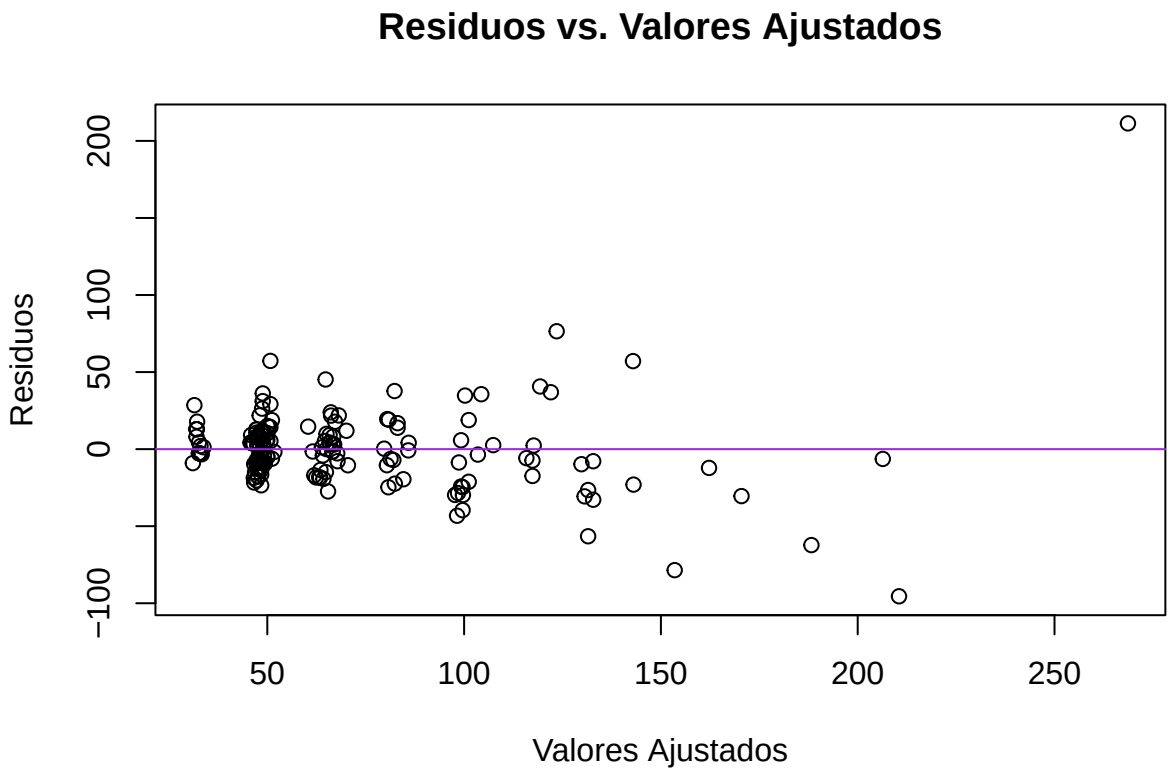
```r
df_madrid_filtrado <- df_madrid |> select(-Neighbourhood)
formula <- as.formula("Square.Meters ~ Bathrooms + Price + Bedrooms")
model <- lm(formula, data = df_madrid_filtrado)
summary(model)
```

```
##
## 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
```

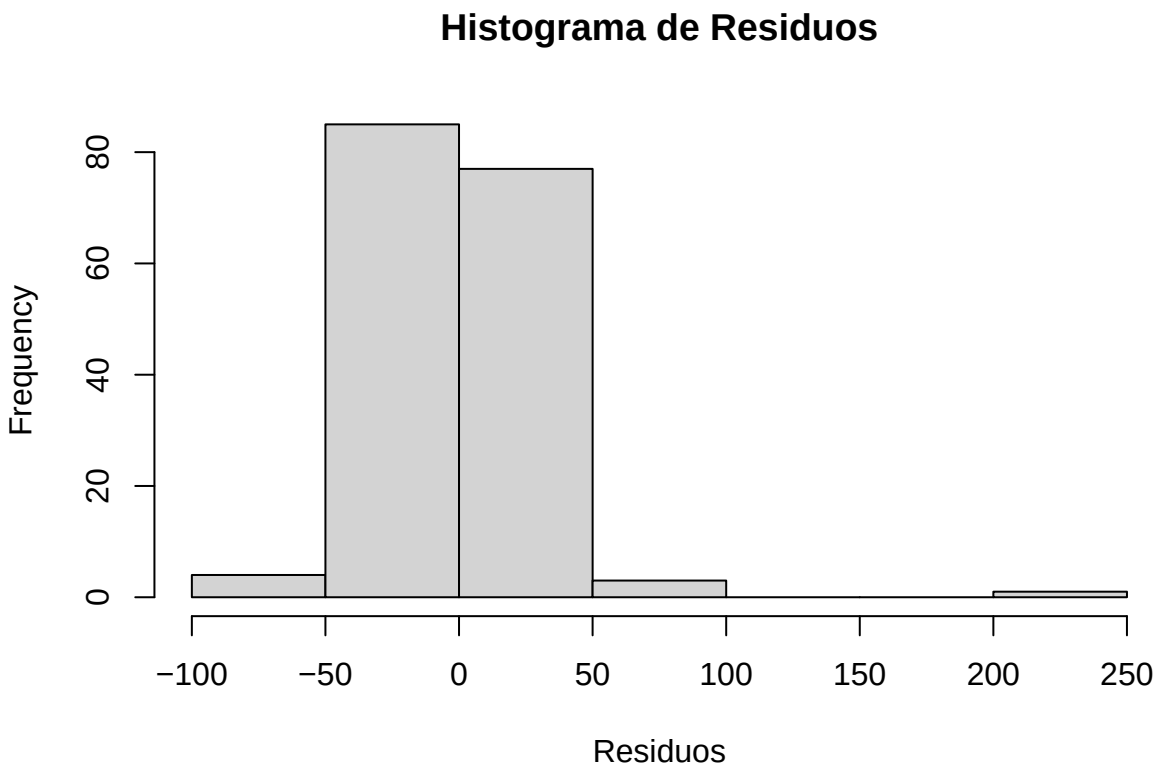
```
##
## Residual standard error: 27.26 on 166 degrees of freedom
##   (4731 observations deleted due to missingness)
## Multiple R-squared:  0.6599, Adjusted R-squared:  0.6537
## F-statistic: 107.4 on 3 and 166 DF,  p-value: < 2.2e-16
```

Evaluo la calidad del modelo

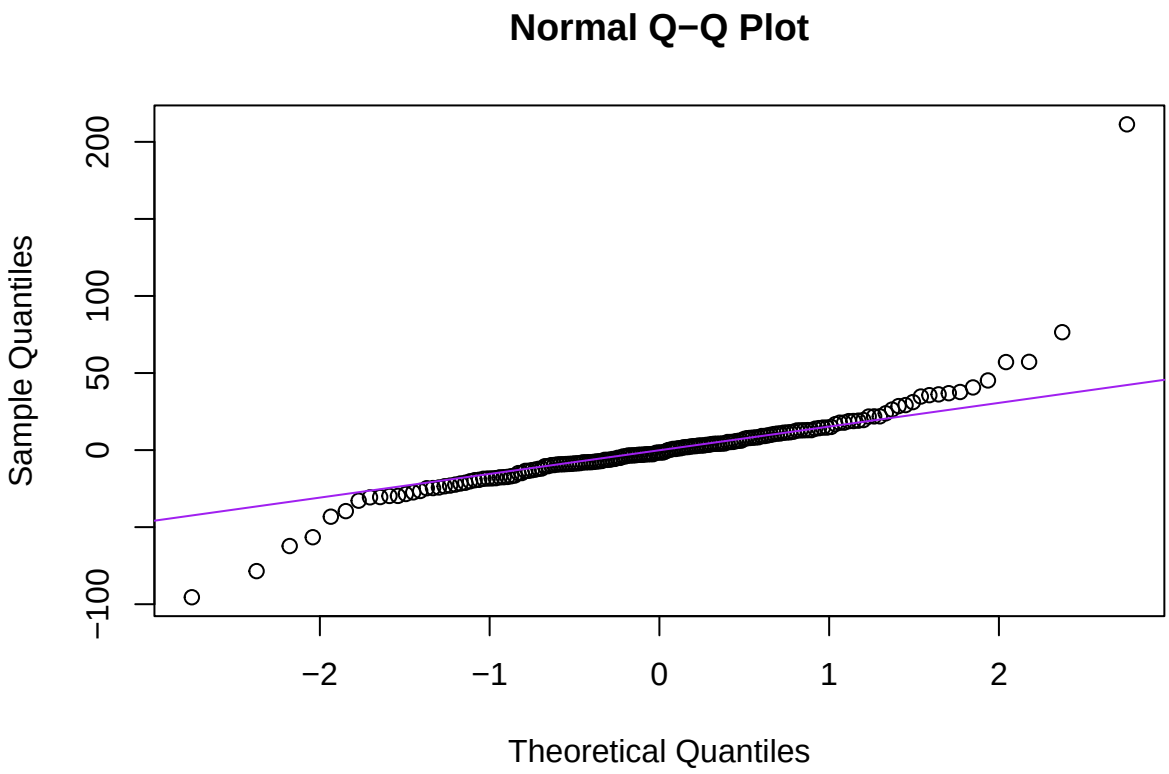
```r
# Diagnóstico de los residuos
plot(model$fitted.values, model$residuals, xlab = "Valores Ajustados", ylab = "Residuos", main = "Residu
abline(h = 0, col = "purple")
```




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




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




```
# 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 50€/noche y 3 habitaciones en el barrio de Sol, con 4 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 = 4))
```

```
##        1
## 93.56214
```

FIN.