


PORTADA

Nombre Alumno / DNI	David Gómez Sedas
Título del Programa	Business Analytics & Data Science
Nº Unidad y Título	Unidad 25 - Machine Learning
Año académico	2023-2024
Profesor de la unidad	Ángel Bravo
Título del Assignment	Elaboración de un informe ejecutivo
Día de emisión	20/09/2023
Día de entrega	24/01/2024
Nombre IV y fecha	
Declaración del estudiante	<p>Certifico que la presentación del assignment es completamente mi propio trabajo y entiendo completamente las consecuencias del plagio. Entiendo que hacer una declaración falsa es una forma de mala práctica.</p> <p>Fecha: 20/01/2024</p> <p>Firma del alumno:</p> 

Plagio

El plagio es una forma particular de hacer trampa. El plagio debe evitarse a toda costa y los alumnos que infrinjan las reglas, aunque sea inocentemente, pueden ser sancionados. Es su responsabilidad asegurarse de comprender las prácticas de referencia correctas. Como alumno de nivel universitario, se espera que utilice las referencias adecuadas en todo momento y mantenga notas cuidadosamente detalladas de todas sus fuentes de materiales para el material que ha utilizado en su trabajo, incluido cualquier material descargado de Internet. Consulte al profesor de la unidad correspondiente o al tutor del curso si necesita más consejos.

Airbnb - Revenue Management

David Gómez Sedas

A continuación se enumeran todas las librerías utilizadas en el AB:

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.4.4      v tibble     3.2.1
v lubridate  1.9.3      v tidyr      1.3.0
v purrr      1.0.2
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(tidymodels)
```

```
-- Attaching packages ----- tidymodels 1.1.1 --
v broom       1.0.5      v rsample     1.2.0
v dials       1.2.0      v tune        1.1.2
v infer       1.0.5      v workflows   1.1.3
v modeldata   1.3.0      v workflowsets 1.0.1
v parsnip     1.1.1      v yardstick   1.3.0
v recipes     1.0.9
-- Conflicts ----- tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
* Dig deeper into tidy modeling with R at https://www.tmw.r.org
```

```
library(quarto)
library(fastDummies)
```

Thank you for using fastDummies!

To acknowledge our work, please cite the package:

Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy (Binary) Columns and R

```
library(GGally)
```

Registered S3 method overwritten by 'GGally':

```
method from
+.gg      ggplot2
```

```
library(factoextra)
```

Welcome! Want to learn more? See two factoextra-related books at <https://goo.gl/ve3WBa>

Datos usados

```
df <- read_csv("listings_bcn.csv", show_col_types = FALSE)
```

Origen de los datos

Los datos han sido extraídos de “Inside Airbnb”, el cual es un proyecto que provee datos e información sobre el impacto que tiene Airbnb sobre las comunidades residenciales.

La finalidad de este proyecto es trabajar en torno a una visión que consiste en informar a estas comunidades sobre el uso real de airbnb.

Los datos son extraídos de la web de Airbnb oficial y son almacenados en archivos compatibles con datos tabulares para distribuirlos. Podemos encontrar que los datos están separados por ciudades y estos se suelen actualizar frecuentemente para mostrar una imagen lo mas actual posible.

Todos los datos de la web estan licenciados bajo Creative Commons Attribution 4.0 International License.

Concretamente, los datos utilizados en este trabajo consisten en todas las residencias activas en Airbnb de la ciudad de Barcelona. En ellas podemos encontrar diferentes tipos de servicios: Apartamentos, casas, habitaciones y camas en habitaciones compartidas.

Contenido

Los datos utilizados consisten en unos datos tabulares los cuales consisten en 18086 entradas y 75 columnas.

```
str(df)
```

```
spc_tbl_ [18,086 x 75] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ id                : num [1:18086] 18674 23197 117010 32711 1182
 $ listing_url       : chr [1:18086] "https://www.airbnb.com/rooms,
 $ scrape_id         : num [1:18086] 2.02e+13 2.02e+13 2.02e+13 2.
 $ last_scraped      : Date[1:18086], format: "2023-09-06" "2023-09
 $ source            : chr [1:18086] "city scrape" "city scrape" "p
 $ name              : chr [1:18086] "Rental unit in Barcelona · 4
 $ description       : chr [1:18086] "110m2 apartment to rent in Ba
 $ neighborhood_overview : chr [1:18086] "Apartment in Barcelona locate
 $ picture_url       : chr [1:18086] "https://a0.muscache.com/pictu
 $ host_id           : num [1:18086] 71615 90417 567180 135703 567
 $ host_url          : chr [1:18086] "https://www.airbnb.com/users,
 $ host_name         : chr [1:18086] "Mireia And Maria" "Etain (Ma
 $ host_since        : Date[1:18086], format: "2010-01-19" "2010-03
 $ host_location     : chr [1:18086] "Barcelona, Spain" "Catalonia
 $ host_about        : chr [1:18086] "We are Mireia (47) & Maria (4
 $ host_response_time : chr [1:18086] "within an hour" "within an ho
 $ host_response_rate : chr [1:18086] "97%" "100%" "88%" "100%" ...
 $ host_acceptance_rate : chr [1:18086] "90%" "94%" "98%" "100%" ...
 $ host_is_superhost  : logi [1:18086] FALSE TRUE FALSE FALSE FALSE
 $ host_thumbnail_url : chr [1:18086] "https://a0.muscache.com/im/us
 $ host_picture_url   : chr [1:18086] "https://a0.muscache.com/im/us
 $ host_neighbourhood : chr [1:18086] "la Sagrada Família" "El Besòs
 $ host_listings_count : num [1:18086] 47 6 19 3 19 1 34 3 34 2 ...
 $ host_total_listings_count : num [1:18086] 48 9 19 15 19 1 50 4 50 2 ...
 $ host_verifications : chr [1:18086] "["email", 'phone']" "["email
 $ host_has_profile_pic : logi [1:18086] TRUE TRUE TRUE TRUE TRUE TRUE
 $ host_identity_verified : logi [1:18086] TRUE TRUE FALSE TRUE FALSE TR
 $ neighbourhood     : chr [1:18086] "Barcelona, CT, Spain" "Sant A
 $ neighbourhood_cleansed : chr [1:18086] "la Sagrada Família" "el Besòs
 $ neighbourhood_group_cleansed : chr [1:18086] "Eixample" "Sant Martí" "Eixar
 $ latitude          : num [1:18086] 41.4 41.4 41.4 41.4 41.4 ...
 $ longitude         : num [1:18086] 2.17 2.22 2.17 2.17 2.17 ...
 $ property_type      : chr [1:18086] "Entire rental unit" "Entire r
 $ room_type         : chr [1:18086] "Entire home/apt" "Entire home
```

```

$ accommodates : num [1:18086] 8 5 8 6 8 2 7 9 7 5 ...
$ bathrooms : logi [1:18086] NA NA NA NA NA NA ...
$ bathrooms_text : chr [1:18086] "2 baths" "2 baths" "2 baths"
$ bedrooms : num [1:18086] 3 3 3 2 3 1 3 4 3 2 ...
$ beds : num [1:18086] 6 4 6 3 5 1 5 6 4 3 ...
$ amenities : chr [1:18086] "[\"Refrigerator\", \"30\\\"
$ price : chr [1:18086] "$202.00" "$255.00" "$331.00"
$ minimum_nights : num [1:18086] 1 3 2 21 2 3 3 5 3 90 ...
$ maximum_nights : num [1:18086] 1125 300 30 31 28 ...
$ minimum_minimum_nights : num [1:18086] 1 3 2 1 2 3 2 2 3 90 ...
$ maximum_minimum_nights : num [1:18086] 4 5 3 1 3 3 6 5 6 90 ...
$ minimum_maximum_nights : num [1:18086] 1125 1125 30 31 28 ...
$ maximum_maximum_nights : num [1:18086] 1125 1125 32 31 32 ...
$ minimum_nights_avg_ntm : num [1:18086] 2.2 3.2 2 1 2.1 3 3.3 4.9 3.3
$ maximum_nights_avg_ntm : num [1:18086] 1125 1125 30.3 31 28.7 ...
$ calendar_updated : logi [1:18086] NA NA NA NA NA NA ...
$ has_availability : logi [1:18086] TRUE TRUE FALSE TRUE FALSE TR
$ availability_30 : num [1:18086] 4 16 0 6 0 2 7 5 6 30 ...
$ availability_60 : num [1:18086] 11 31 0 17 0 9 18 20 12 60 ..
$ availability_90 : num [1:18086] 21 61 0 43 0 32 18 38 31 90 .
$ availability_365 : num [1:18086] 34 150 0 310 0 303 266 194 279
$ calendar_last_scraped : Date[1:18086], format: "2023-09-06" "2023-09-06"
$ number_of_reviews : num [1:18086] 38 73 48 95 50 395 32 211 85 1
$ number_of_reviews_ltm : num [1:18086] 8 11 6 21 18 62 14 31 33 0 ..
$ number_of_reviews_l30d : num [1:18086] 0 1 1 1 0 6 1 2 1 0 ...
$ first_review : Date[1:18086], format: "2013-05-27" "2011-03-01"
$ last_review : Date[1:18086], format: "2023-06-26" "2023-06-26"
$ review_scores_rating : num [1:18086] 4.3 4.77 4.55 4.46 4.56 4.86 4.4
$ review_scores_accuracy : num [1:18086] 4.41 4.93 4.59 4.44 4.5 4.9 4.4
$ review_scores_cleanliness : num [1:18086] 4.62 4.89 4.57 4.47 4.62 4.93
$ review_scores_checkin : num [1:18086] 4.76 4.94 4.82 4.86 4.56 4.93
$ review_scores_communication : num [1:18086] 4.65 4.99 4.91 4.85 4.74 4.9 4.4
$ review_scores_location : num [1:18086] 4.78 4.6 4.86 4.86 4.9 4.72 4.4
$ review_scores_value : num [1:18086] 4.27 4.64 4.59 4.52 4.52 4.82
$ license : chr [1:18086] "HUTB-002062" "HUTB005057" "HUTB005057"
$ instant_bookable : logi [1:18086] TRUE FALSE FALSE TRUE FALSE TR
$ calculated_host_listings_count : num [1:18086] 30 2 19 3 19 1 32 2 32 2 ...
$ calculated_host_listings_count_entire_homes : num [1:18086] 30 2 19 3 19 1 32 2 32 2 ...
$ calculated_host_listings_count_private_rooms : num [1:18086] 0 0 0 0 0 0 0 0 0 0 ...
$ calculated_host_listings_count_shared_rooms : num [1:18086] 0 0 0 0 0 0 0 0 0 0 ...
$ reviews_per_month : num [1:18086] 0.3 0.48 0.33 0.64 0.34 2.7 0
- attr(*, "spec")=
.. cols(

```

```

.. id = col_double(),
.. listing_url = col_character(),
.. scrape_id = col_double(),
.. last_scraped = col_date(format = ""),
.. source = col_character(),
.. name = col_character(),
.. description = col_character(),
.. neighborhood_overview = col_character(),
.. picture_url = col_character(),
.. host_id = col_double(),
.. host_url = col_character(),
.. host_name = col_character(),
.. host_since = col_date(format = ""),
.. host_location = col_character(),
.. host_about = col_character(),
.. host_response_time = col_character(),
.. host_response_rate = col_character(),
.. host_acceptance_rate = col_character(),
.. host_is_superhost = col_logical(),
.. host_thumbnail_url = col_character(),
.. host_picture_url = col_character(),
.. host_neighbourhood = col_character(),
.. host_listings_count = col_double(),
.. host_total_listings_count = col_double(),
.. host_verifications = col_character(),
.. host_has_profile_pic = col_logical(),
.. host_identity_verified = col_logical(),
.. neighbourhood = col_character(),
.. neighbourhood_cleansed = col_character(),
.. neighbourhood_group_cleansed = col_character(),
.. latitude = col_double(),
.. longitude = col_double(),
.. property_type = col_character(),
.. room_type = col_character(),
.. accommodates = col_double(),
.. bathrooms = col_logical(),
.. bathrooms_text = col_character(),
.. bedrooms = col_double(),
.. beds = col_double(),
.. amenities = col_character(),
.. price = col_character(),
.. minimum_nights = col_double(),
.. maximum_nights = col_double(),

```

```

..   minimum_minimum_nights = col_double(),
..   maximum_minimum_nights = col_double(),
..   minimum_maximum_nights = col_double(),
..   maximum_maximum_nights = col_double(),
..   minimum_nights_avg_ntm = col_double(),
..   maximum_nights_avg_ntm = col_double(),
..   calendar_updated = col_logical(),
..   has_availability = col_logical(),
..   availability_30 = col_double(),
..   availability_60 = col_double(),
..   availability_90 = col_double(),
..   availability_365 = col_double(),
..   calendar_last_scraped = col_date(format = ""),
..   number_of_reviews = col_double(),
..   number_of_reviews_ltm = col_double(),
..   number_of_reviews_l30d = col_double(),
..   first_review = col_date(format = ""),
..   last_review = col_date(format = ""),
..   review_scores_rating = col_double(),
..   review_scores_accuracy = col_double(),
..   review_scores_cleanliness = col_double(),
..   review_scores_checkin = col_double(),
..   review_scores_communication = col_double(),
..   review_scores_location = col_double(),
..   review_scores_value = col_double(),
..   license = col_character(),
..   instant_bookable = col_logical(),
..   calculated_host_listings_count = col_double(),
..   calculated_host_listings_count_entire_homes = col_double(),
..   calculated_host_listings_count_private_rooms = col_double(),
..   calculated_host_listings_count_shared_rooms = col_double(),
..   reviews_per_month = col_double()
.. )
- attr(*, "problems")=<externalptr>

```

En el archivo encontramos cuatro tipos de datos distintos:

- double
- character
- logical
- date

Los datos nos hablan de todos los apartados relacionados con el alojamiento ofrecido, la información del host, el precio medio, la disponibilidad, imagenes de los alojamientos e información de las reseñas.

Definición de variables

Vamos a definir las variables que contienen los datos

id	Identificador único del alojamiento en Airbnb
listing_url	Link al portal del alojamiento
scrape_id	Identificador de scrapping de los datos
last_scraped	Hora del scrapping
source	Fuente, de la entrada, si es reciente o antigua.
name	Nombre del alojamiento
description	Descripción del alojamiento
neighborhood_overview	Descripción del barrio del host
picture_url	Link de las fotografías del alojamiento
host_id	Identificador del host del alojamiento
host_url	Link al portal del perfil del host
host_name	Nombre del host
host_since	Fecha de registro como host
host_location	Localización del host
host_about	Descripción del host sobre si mismo.
host_response_time	Tiempo promedio de respuesta del host
host_response_rate	Porcentaje de respuestas realizadas por el host.
host_acceptance_rate	Porcentaje de solicitudes de alojamiento aceptadas por el host.
host_is_superhost	Estatus otorgado por Airbnb por ser un host con ciertas características positivas según la plataforma
host_picture_url	Link de la foto de perfil del host
host_neighbourhood	Barrio donde reside el host
host_listings_count	Número de alojamientos listados por el host (cuenta manual)
host_total_listings_count	Número de alojamientos listados por el host (cuenta segun Airbnb)
host_verifications	Métodos por los cuales el host se ha verificado
host_has_profile_pic	Boolean que indica si el host se ha verificado
host_identity_verified	Boolean que indica si la identidad se ha verificado
neighbourhood_group	Barrio según coordenadas.
latitude	Coordenada de latitud del alojamiento
longitude	Coordenada de longitud del alojamiento
property_type	Tipo de propiedad del alojamiento
room_type	Tipo de habitación, si es compartida o individual.
accommodates	Capacidad de personas
bathrooms	Número de banos

bedrooms	Número de dormitorios
beds	Número de camas
price	Precio diario para alojarse
minimum_nights	Noches mínimas que hay alojarse para reservar
maximum_nights	Noches máximas que se pueden reservar
has_availability	Indica si hay disponibilidad.
availability_30	Disponibilidad en los proximos 30 dias.
availability_60	Disponibilidad en los proximos 60 dias.
availability_90	Disponibilidad en los proximos 90 dias.
availability_365	Disponibilidad en los proximos 365 dias.
number_of_reviews	Número de reviews que ha recibido el alojamiento.
number_of_reviews_l12m	Número de reviews que ha recibido el alojamiento en los últimos 12 meses.
number_of_reviews_l30d	Número de reviews que ha recibido el alojamiento en los últimos 30 dias.
first_review	Fecha de la primera review del alojamiento.
last_review	Fecha de la última review del alojamiento
review_scores_rating	Nota media de las reviews en general
review_scores_accuracy	Nota media de las reviews en precisión
review_scores_cleanliness	Nota media de las reviews en limpieza
review_scores_checkin	Nota media de las reviews en el proceso de checkin
review_scores_communication	Nota media de las reviews en comunicación
review_scores_location	Nota media de las reviews en localización
review_scores_value	Nota media de las reviews en valor (calidad / precio)
license	Licencia del alojamiento
instant_bookable	Indica si el alojamiento se reserva automáticamente o se necesita la aprobación del host.

Muestra de los datos

A continuación, una muestra de los datos obtenidos:

```
head(df)
```

```
# A tibble: 6 x 75
  id listing_url      scrape_id last_scraped source name description
  <dbl> <chr>          <dbl> <date>    <chr> <chr> <chr>
1  18674 https://www.airbnb.com~  2.02e13 2023-09-06 city ~ Rent~ 110m2 apar~
2  23197 https://www.airbnb.com~  2.02e13 2023-09-06 city ~ Rent~ Beautiful ~
3  117010 https://www.airbnb.com~  2.02e13 2023-09-06 previ~ Rent~ Have an au~
```

```

4 32711 https://www.airbnb.com~ 2.02e13 2023-09-06 city ~ Rent~ A lovely t~
5 118228 https://www.airbnb.com~ 2.02e13 2023-09-06 previ~ Rent~ Modern 100~
6 128463 https://www.airbnb.com~ 2.02e13 2023-09-06 city ~ Rent~ My House i~
# i 68 more variables: neighborhood_overview <chr>, picture_url <chr>,
# host_id <dbl>, host_url <chr>, host_name <chr>, host_since <date>,
# host_location <chr>, host_about <chr>, host_response_time <chr>,
# host_response_rate <chr>, host_acceptance_rate <chr>,
# host_is_superhost <lgl>, host_thumbnail_url <chr>, host_picture_url <chr>,
# host_neighbourhood <chr>, host_listings_count <dbl>,
# host_total_listings_count <dbl>, host_verifications <chr>, ...

```

Vamos a ver un análisis rápido de los datos:

```
summary(df)
```

id		listing_url	scrape_id	
Min.	:1.867e+04	Length:18086	Min.	:2.023e+13
1st Qu.	:2.172e+07	Class :character	1st Qu.	:2.023e+13
Median	:4.435e+07	Mode :character	Median	:2.023e+13
Mean	:2.997e+17		Mean	:2.023e+13
3rd Qu.	:7.450e+17		3rd Qu.	:2.023e+13
Max.	:9.740e+17		Max.	:2.023e+13

last_scraped	source	name	description
Min.	:2023-09-06	Length:18086	Length:18086
1st Qu.	:2023-09-06	Class :character	Class :character
Median	:2023-09-06	Mode :character	Mode :character
Mean	:2023-09-06		
3rd Qu.	:2023-09-06		
Max.	:2023-09-06		

neighborhood_overview	picture_url	host_id
Length:18086	Length:18086	Min. : 3073
Class :character	Class :character	1st Qu.: 9919300
Mode :character	Mode :character	Median : 96299106
		Mean :166184865
		3rd Qu.:310348791
		Max. :535400790

host_url	host_name	host_since	host_location
Length:18086	Length:18086	Min. :2008-09-19	Length:18086
Class :character	Class :character	1st Qu.:2013-11-10	Class :character

Mode :character	Mode :character	Median :2016-09-22	Mode :character
		Mean :2016-12-27	
		3rd Qu.:2019-11-18	
		Max. :2023-09-04	
		NA's :2	
host_about	host_response_time	host_response_rate	host_acceptance_rate
Length:18086	Length:18086	Length:18086	Length:18086
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character

host_is_superhost	host_thumbnail_url	host_picture_url	host_neighbourhood
Mode :logical	Length:18086	Length:18086	Length:18086
FALSE:14580	Class :character	Class :character	Class :character
TRUE :3087	Mode :character	Mode :character	Mode :character
NA's :419			

host_listings_count	host_total_listings_count	host_verifications
Min. : 1.00	Min. : 1.00	Length:18086
1st Qu.: 2.00	1st Qu.: 2.00	Class :character
Median : 6.00	Median : 9.00	Mode :character
Mean : 41.13	Mean : 53.55	
3rd Qu.: 34.00	3rd Qu.: 48.00	
Max. :786.00	Max. :1853.00	
NA's :2	NA's :2	
host_has_profile_pic	host_identity_verified	neighbourhood
Mode :logical	Mode :logical	Length:18086
FALSE:330	FALSE:1471	Class :character
TRUE :17754	TRUE :16613	Mode :character
NA's :2	NA's :2	

neighbourhood_cleansed	neighbourhood_group_cleansed	latitude
Length:18086	Length:18086	Min. :41.35
Class :character	Class :character	1st Qu.:41.38
Mode :character	Mode :character	Median :41.39
		Mean :41.39
		3rd Qu.:41.40

Max. :41.46

longitude	property_type	room_type	accommodates
Min. :2.092	Length:18086	Length:18086	Min. : 1.000
1st Qu.:2.157	Class :character	Class :character	1st Qu.: 2.000
Median :2.168	Mode :character	Mode :character	Median : 3.000
Mean :2.167			Mean : 3.365
3rd Qu.:2.177			3rd Qu.: 4.000
Max. :2.228			Max. :16.000

bathrooms	bathrooms_text	bedrooms	beds
Mode:logical	Length:18086	Min. : 1.000	Min. : 1.000
NA's:18086	Class :character	1st Qu.: 1.000	1st Qu.: 1.000
	Mode :character	Median : 2.000	Median : 2.000
		Mean : 2.066	Mean : 2.358
		3rd Qu.: 3.000	3rd Qu.: 3.000
		Max. :12.000	Max. :30.000
		NA's :6360	NA's :314

amenities	price	minimum_nights	maximum_nights
Length:18086	Length:18086	Min. : 1.00	Min. : 1.0
Class :character	Class :character	1st Qu.: 1.00	1st Qu.: 180.2
Mode :character	Mode :character	Median : 3.00	Median : 365.0
		Mean : 14.76	Mean : 574.4
		3rd Qu.: 31.00	3rd Qu.:1125.0
		Max. :1125.00	Max. :3000.0

minimum_minimum_nights	maximum_minimum_nights	minimum_maximum_nights
Min. : 1.00	Min. : 1.00	Min. :1.000e+00
1st Qu.: 1.00	1st Qu.: 3.00	1st Qu.:3.000e+02
Median : 2.00	Median : 5.00	Median :3.650e+02
Mean : 14.54	Mean : 18.89	Mean :2.381e+05
3rd Qu.: 31.00	3rd Qu.: 31.00	3rd Qu.:1.125e+03
Max. :1125.00	Max. :2705.00	Max. :2.147e+09

maximum_maximum_nights	minimum_nights_avg_ntm	maximum_nights_avg_ntm
Min. :1.000e+00	Min. : 1.00	Min. :1.000e+00
1st Qu.:3.330e+02	1st Qu.: 2.00	1st Qu.:3.300e+02
Median :9.990e+02	Median : 3.70	Median :4.070e+02
Mean :2.382e+05	Mean : 17.34	Mean :2.381e+05
3rd Qu.:1.125e+03	3rd Qu.: 31.00	3rd Qu.:1.125e+03
Max. :2.147e+09	Max. :1125.00	Max. :2.147e+09

calendar_updated has_availability availability_30 availability_60

Mode:logical	Mode :logical	Min. : 0.000	Min. : 0.00
NA's:18086	FALSE:1188	1st Qu.: 0.000	1st Qu.: 0.00
	TRUE :16898	Median : 4.000	Median :13.00
		Mean : 7.718	Mean :19.37
		3rd Qu.:12.000	3rd Qu.:34.00
		Max. :30.000	Max. :60.00

availability_90	availability_365	calendar_last_scraped	number_of_reviews
Min. : 0.00	Min. : 0.0	Min. :2023-09-06	Min. : 0.00
1st Qu.: 1.00	1st Qu.: 47.0	1st Qu.:2023-09-06	1st Qu.: 1.00
Median :35.00	Median :175.0	Median :2023-09-06	Median : 6.00
Mean :36.43	Mean :171.8	Mean :2023-09-06	Mean : 42.22
3rd Qu.:62.00	3rd Qu.:302.0	3rd Qu.:2023-09-06	3rd Qu.: 42.00
Max. :90.00	Max. :365.0	Max. :2023-09-06	Max. :1817.00

number_of_reviews_ltm	number_of_reviews_l30d	first_review
Min. : 0.00	Min. : 0.0000	Min. :2010-10-03
1st Qu.: 0.00	1st Qu.: 0.0000	1st Qu.:2017-08-16
Median : 2.00	Median : 0.0000	Median :2020-06-28
Mean : 11.44	Mean : 0.9239	Mean :2019-12-27
3rd Qu.: 15.00	3rd Qu.: 1.0000	3rd Qu.:2022-09-25
Max. :836.00	Max. :91.0000	Max. :2023-09-05
		NA's :4466

last_review	review_scores_rating	review_scores_accuracy
Min. :2011-06-23	Min. :0.00	Min. :0.000
1st Qu.:2023-01-31	1st Qu.:4.41	1st Qu.:4.500
Median :2023-08-02	Median :4.68	Median :4.750
Mean :2022-10-26	Mean :4.54	Mean :4.621
3rd Qu.:2023-08-22	3rd Qu.:4.90	3rd Qu.:4.930
Max. :2023-09-06	Max. :5.00	Max. :5.000
NA's :4466	NA's :4466	NA's :4549

review_scores_cleanliness	review_scores_checkin	review_scores_communication
Min. :0.000	Min. :0.000	Min. :0.000
1st Qu.:4.450	1st Qu.:4.640	1st Qu.:4.640
Median :4.720	Median :4.850	Median :4.850
Mean :4.586	Mean :4.711	Mean :4.709
3rd Qu.:4.920	3rd Qu.:5.000	3rd Qu.:5.000
Max. :5.000	Max. :5.000	Max. :5.000
NA's :4548	NA's :4553	NA's :4547

review_scores_location	review_scores_value	license	instant_bookable
Min. :0.000	Min. :0.000	Length:18086	Mode :logical
1st Qu.:4.670	1st Qu.:4.280	Class :character	FALSE:11248
Median :4.830	Median :4.560	Mode :character	TRUE :6838

Mean	:4.737	Mean	:4.444
3rd Qu.:	5.000	3rd Qu.:	4.780
Max.	:5.000	Max.	:5.000
NA's	:4552	NA's	:4553
calculated_host_listings_count			
Min.	: 1.00	Min.	: 0.00
1st Qu.:	1.00	1st Qu.:	0.00
Median	: 5.00	Median	: 2.00
Mean	: 31.63	Mean	: 23.87
3rd Qu.:	28.00	3rd Qu.:	19.00
Max.	:294.00	Max.	:294.00
calculated_host_listings_count_entire_homes			
Min.	: 0.000	Min.	: 0.00
1st Qu.:	0.000	1st Qu.:	0.00
Median	: 0.000	Median	: 2.00
Mean	: 7.602	Mean	: 23.87
3rd Qu.:	2.000	3rd Qu.:	19.00
Max.	:233.000	Max.	:294.00
calculated_host_listings_count_private_rooms			
Min.	: 0.000	Min.	: 0.00
1st Qu.:	0.000	1st Qu.:	0.00
Median	: 0.000	Median	: 2.00
Mean	: 7.602	Mean	: 23.87
3rd Qu.:	2.000	3rd Qu.:	19.00
Max.	:233.000	Max.	:294.00
calculated_host_listings_count_shared_rooms			
Min.	: 0.00000	Min.	: 0.010
1st Qu.:	0.00000	1st Qu.:	0.240
Median	: 0.00000	Median	: 0.850
Mean	: 0.08332	Mean	: 1.436
3rd Qu.:	0.00000	3rd Qu.:	2.100
Max.	:12.00000	Max.	:55.020
		NA's	:4466

Objetivo del proyecto

El objetivo con el análisis de datos es crear un modelo el cual enseñe a actuales o futuros propietarios de alojamientos de airbnb a obtener los máximos ingresos posibles teniendo en cuenta todas las características en del alojamiento y su perfil en Airbnb.

La variable a predecir es “price”, la cual es el precio de reservar cada alojamiento en su fecha de disponibilidad mas reciente.

El modelo ideal será en el que haya menos residuo entre los valores reales y los valores predichos.

Modelo escogido

El modelo escogido es una regresión lineal, que es de aprendizaje supervisado y continuo. Hemos tenido en cuenta otros posibles modelos y hemos llegado a las siguientes conclusiones:

Modelo	Pros	Contras
Regresión lineal	<ul style="list-style-type: none">• Es simple y fácil de entender, los resultados pueden ser visualizados rápidamente.	<ul style="list-style-type: none">• Se basa en supuestos estrictos que muchas veces no se cumplen en datos reales.
Regresión Ridge	<ul style="list-style-type: none">• Los coeficientes de regresión pueden interpretarse directamente.• Controla el sobreajuste, penalizando a los coeficientes grandes.• Menos sensible a valores atípicos.	<ul style="list-style-type: none">• Es sensible a valores atípicos.• No realiza selección de variables, todos los predictores se mantienen en el modelo.
Regresión Lasso	<ul style="list-style-type: none">• Es capaz de realizar selección de variables reduciendo algunos coeficientes a cero.• Útil cuando se tienen muchas variables y se desea simplificar el modelo.	<ul style="list-style-type: none">• Requiere la selección del parámetros de regularización.• Puede ser inestable en la presencia de predictores altamente correlacionados.• También requiere la selección cuidadosa del parámetro de regularización.

Programa y paquetes seleccionados: Decisión y razones

R es un lenguaje de programación muy utilizado en estadística y aprendizaje automático debido a su gran cantidad de paquetes y funciones para análisis de datos y modelado estadístico. Su popularidad entre los estadísticos y científicos de datos se debe a la facilidad con la que se pueden manipular datos, realizar cálculos estadísticos y generar gráficos avanzados. Para esta funcionalidad concreta, en los últimos años, han destacado dos grupos de librerías que también vamos a utilizar en nuestro modelo, como son Tidyverse y Tidymodels.

El tidyverse es una colección de paquetes de R diseñados para la ciencia de datos que comparten una filosofía subyacente de diseño y gramática. Algunos de los paquetes más destacados del tidyverse incluyen:

- dplyr y tidyr para la manipulación de datos.
- Ggplot para la creación de visualizaciones avanzadas.

Tidymodels tiene el objetivo de proporcionar una gramática coherente y fácil de usar para la modelización predictiva en R. Algunos paquetes son:

- rsample que contiene funciones para la separación de sets de un dataframe
- yardstick: Incluye métricas para medir el rendimiento de los modelos.

Ambos grupos de librerías tienen finalidades similares como el uso de una gramática consistente, el formato con pipes y en general que sea legible y simple.

Limpieza de datos

En primer lugar vamos a eliminar aquellas variables que contienen datos que no vamos a utilizar por alguno de los siguientes motivos:

- Corresponden a links de cada una de las entradas
- Id's de elementos concretos que no vamos a utilizar
- Descripciones y otros elementos strings
- Variables cubiertas por otras las cuales estandarizan el formato

```
df <- df |>
  select(c(id,
           host_id,
           host_since,
           host_response_time,
           host_response_rate,
           host_acceptance_rate,
           host_is_superhost,
           host_verifications,
           host_has_profile_pic,
           host_identity_verified,
           calculated_host_listings_count,
           calculated_host_listings_count_entire_homes,
           calculated_host_listings_count_private_rooms,
           calculated_host_listings_count_shared_rooms,
           neighbourhood_group_cleansed,
           latitude,
           longitude,
           room_type,
           accommodates,
           bathrooms_text,
```



```

bedrooms,
beds,
price,
minimum_nights,
maximum_nights,
has_availability,
availability_30,
availability_60,
availability_90,
availability_365,
instant_bookable,
number_of_reviews,
number_of_reviews_l30d,
review_scores_rating,
review_scores_accuracy,
review_scores_cleanliness,
review_scores_checkin,
review_scores_communication,
review_scores_location,
review_scores_value,
))

```

Ajustes de formato

host_response_time

Nos encontramos con variables categoricas en diferentes medidas de tiempo. Sería conveniente ajustar todas estas para que estén en la misma medida de tiempo.

```
unique(df$host_response_time)
```

```

[1] "within an hour"      "within a few hours" "within a day"
[4] "N/A"                 "a few days or more" NA

```

Encontramos seis valores. A continuación listamos las diferentes categorías y su nuevo valor:

- “within an hour” -> <1h
- “within a few hours” -> ~12h
- “within a day” -> <24h
- “a few days or more” -> >48h
- “N/A” -> Unificar con NA

```
df$host_response_time <- ifelse(df$host_response_time == "within an hour", "<1h",
                                ifelse(df$host_response_time == "within a few hours", "~12h",
                                        ifelse(df$host_response_time == "within a day", "~24h",
                                              ifelse(df$host_response_time == "a few days or more", ">48h",
                                                    ifelse(df$host_response_time == "N/A", NA, df$host_response_time))))))
```

bathrooms_text

Esta variable incluye el valor numérico de baños y además nos indica si estos son compartidos o no. Para la claridad en los datos sería conveniente crear dos variables de esta:

1. Una variable número con el número de baños.
2. Un Boolean indicando si estos son compartidos o no.

```
unique(df$bathrooms_text)
```

[1] "2 baths"	"1.5 baths"	"2.5 baths"
[4] "3 baths"	"1 shared bath"	"1 bath"
[7] "1 private bath"	"3.5 baths"	"4 baths"
[10] "1.5 shared baths"	NA	"2 shared baths"
[13] "2.5 shared baths"	"5.5 baths"	"7.5 baths"
[16] "4.5 baths"	"6 baths"	"0 shared baths"
[19] "Half-bath"	"Private half-bath"	"0 baths"
[22] "5 baths"	"8 baths"	"3 shared baths"
[25] "8 shared baths"	"4 shared baths"	"Shared half-bath"
[28] "5 shared baths"	"3.5 shared baths"	"6 shared baths"
[31] "5.5 shared baths"	"10 baths"	"10 shared baths"
[34] "6.5 baths"	"4.5 shared baths"	

Vemos que casi todas las categorías tienen números. A excepción de tres de ellas. Vamos a añadirsele para compartir la estructura:

```
df$bathrooms_text <- ifelse(df$bathrooms_text == "Shared half-bath", "0.5 Shared bath",
                            ifelse(df$bathrooms_text == "Private half-bath", "0.5 bath",
                                    ifelse(df$bathrooms_text == "Half-bath", "0.5 bath", df$bathrooms_text)))
```

A continuación creamos la columna boolean la cual nos indica si un baño es compartido o no (Asumimos que si no es indicado, es privado).

```
df$shared_bathrooms <- grepl("shared|Shared", df$bathrooms_text)
```

Seguidamente creamos una nueva columna que extraiga el número de baños del valor bathrooms_text

```
df$n_bathrooms <- sapply(strsplit(df$bathrooms_text, " "), function(x) as.numeric(x[1]))
```

Recolocamos las columnas en la posición de bathrooms_text y eliminamos esta.

```
bathrooms_text_position <- which(colnames(df) == "bathrooms_text")  
  
df <- df %>%  
  relocate(all_of(c("n_bathrooms", "shared_bathrooms")), .before = bathrooms_text_position)
```

Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.

i Please use `all_of()` or `any_of()` instead.

Was:

```
data %>% select(bathrooms_text_position)
```

Now:

```
data %>% select(all_of(bathrooms_text_position))
```

See <<https://tidyselect.r-lib.org/reference/faq-external-vector.html>>.

```
df <- df %>%  
  select(-bathrooms_text)  
  
rm(bathrooms_text_position)
```

host_verifications

En esta variable podemos ver en formato lista, los medios por los cuales se ha verificado un host. Para que los datos sean mas accesibles podemos crear una variable de tipo boolean por cada tipo de medio posible de verificación.

```
unique(df$host_verifications)
```

[1] "["email", 'phone']"	"['email', 'phone', 'work_email']"
[3] "["phone", 'work_email']"	"['phone']"
[5] "["email"]"	"None"
[7] "[]"	"['work_email']"

Encontramos tres tipos de verificación:

- “email”
- “phone”
- “work_email”

Creamos las tres variables y añadimos true si ese método está incluido en la columna `host_verifications`.

```
df <- df %>%
  mutate(verification_email = str_detect(df$host_verifications, "email"),
         verification_phone = str_detect(df$host_verifications, "phone"),
         verification_work_email = str_detect(df$host_verifications, "work_email"))

host_verification_text_position <- which(colnames(df) == "host_verifications")

df <- df %>%
  relocate(all_of(c("verification_email", "verification_phone", "verification_work_email")),
```

Warning: Using an external vector in selections was deprecated in tidysselect 1.1.0.

i Please use `all_of()` or `any_of()` instead.

Was:

```
data %>% select(host_verification_text_position)
```

Now:

```
data %>% select(all_of(host_verification_text_position))
```

See <<https://tidysselect.r-lib.org/reference/faq-external-vector.html>>.

```
df <- df %>%
  select(-host_verifications)

rm(host_verification_text_position)
```

price

Podemos convertir `price` en una variable numérica para poderla usar en visualizaciones y otros tipos de funciones de tipo continuo:

```
df$price <- as.numeric(gsub("\\$", "", df$price))
```

Warning: NAs introduced by coercion

review__score__rating

El valor mínimo que puede tener un alojamiento es 1. En algunos campos pone 0 a causa de que el resto de valores son NA. Vamos a modificar estos valores a NA.

```
df$review_scores_rating <- ifelse(df$review_scores_rating == 0, NA, df$review_scores_rating)
```

host__response__rate y host__acceptance__rate

Cambiamos ambas variables de tipo character con porcentaje a tipo numérico double:

```
df$host_response_rate <- as.numeric(sub("%", "", df$host_response_rate))/100
```

Warning: NAs introduced by coercion

```
df$host_acceptance_rate <- as.numeric(sub("%", "", df$host_acceptance_rate))/100
```

Warning: NAs introduced by coercion

Conversión de campos con tags en factores

Cambiamos todas las columnas de tipos strings que tengan etiquetas a tipo factor para que su uso sea mas sencillo.

```
df <- df %>%  
  mutate(  
    host_response_time = factor(host_response_time),  
    neighbourhood_group_cleansed = factor(neighbourhood_group_cleansed),  
    room_type = factor(room_type)  
  )
```

Tratamiento de NA's

Comprobamos cuantos NA's faltan en cada columna para ver que tratamiento hacer con ellos.

```
na_analysis <- df %>% summarize(across(everything(), ~sum(is.na(.))))  
na_analysis
```

```
# A tibble: 1 x 43
  id host_id host_since host_response_time host_response_rate
  <int>   <int>   <int>           <int>           <int>
1     0     0         2             2963             2963
# i 38 more variables: host_acceptance_rate <int>, host_is_superhost <int>,
# verification_email <int>, verification_phone <int>,
# verification_work_email <int>, host_has_profile_pic <int>,
# host_identity_verified <int>, calculated_host_listings_count <int>,
# calculated_host_listings_count_entire_homes <int>,
# calculated_host_listings_count_private_rooms <int>,
# calculated_host_listings_count_shared_rooms <int>, ...
```

Columnas numéricas

En primer lugar vamos a promediar todas las columnas numéricas:

```
df <- df |>
  mutate(across(where(is.numeric), ~if (any(is.na(.))) floor(replace_na(., mean(., na.rm = T
```

host_response_time

Vamos a hacer una tabla para ver la distribución de cada categoría:

```
frecuencias_host_response_time <- table(df$host_response_time)
porcentajes_host_response_time <- prop.table(frecuencias_host_response_time) * 100
print(porcentajes_host_response_time)
```

<1h	>48h	~12h	~24h
71.936785	1.937446	15.823580	10.302189

Vemos como la gran mayoría responde en menos de una hora, al no ser un componente mayor esta variable vamos a justar los NA's como menores de una hora.

```
df$host_response_time[is.na(df$host_response_time)] <- "<1h"
```

host_is_superhost

Como no tenemos información asumimos que los valores NA son false

```
df$host_is_superhost[is.na(df$host_is_superhost)] <- FALSE
```

host_has_profile_pic

Hacemos la misma gestión en este caso

```
df$host_has_profile_pic[is.na(df$host_has_profile_pic)] <- FALSE
```

host_identity_verified

Volvemos a convertir los valores NA que tenemos en FALSE

```
df$host_identity_verified[is.na(df$host_identity_verified)] <- FALSE
```

host_since

Vamos a usar el valor promedio para sustituir los NA de la fecha de ingreso de los host

```
mean_host_since <- mean(df$host_since, na.rm=TRUE)
df$host_since[is.na(df$host_since)] <- mean_host_since
```

Análisis descriptivo de los datos

Vamos a hacer un análisis descriptivo de nuestros datos para poder tendencias y patrones para poder tomar decisiones sobre el modelo mas adelante.

Summary

Vamos a ver como quedaría el summary después de haber limpiado los datos.

```
summary(df)
```

id	host_id	host_since
Min. :1.867e+04	Min. : 3073	Min. :2008-09-19
1st Qu.:2.172e+07	1st Qu.: 9919300	1st Qu.:2013-11-10
Median :4.435e+07	Median : 96299106	Median :2016-09-22
Mean :2.997e+17	Mean :166184865	Mean :2016-12-27
3rd Qu.:7.450e+17	3rd Qu.:310348791	3rd Qu.:2019-11-17
Max. :9.740e+17	Max. :535400790	Max. :2023-09-04

host_response_time	host_response_rate	host_acceptance_rate	host_is_superhost
<1h :13842	Min. :0.0000	Min. :0.0000	Mode :logical
>48h: 293	1st Qu.:0.0000	1st Qu.:0.0000	FALSE:14999
~12h: 2393	Median :0.0000	Median :0.0000	TRUE :3087
~24h: 1558	Mean :0.4511	Mean :0.2769	
	3rd Qu.:1.0000	3rd Qu.:1.0000	
	Max. :1.0000	Max. :1.0000	

verification_email	verification_phone	verification_work_email
Mode :logical	Mode :logical	Mode :logical
FALSE:1264	FALSE:43	FALSE:15153
TRUE :16822	TRUE :18043	TRUE :2933

host_has_profile_pic	host_identity_verified	calculated_host_listings_count
Mode :logical	Mode :logical	Min. : 1.00
FALSE:332	FALSE:1473	1st Qu.: 1.00
TRUE :17754	TRUE :16613	Median : 5.00
		Mean : 31.63
		3rd Qu.: 28.00
		Max. :294.00

calculated_host_listings_count_entire_homes
Min. : 0.00
1st Qu.: 0.00
Median : 2.00
Mean : 23.87
3rd Qu.: 19.00
Max. :294.00

calculated_host_listings_count_private_rooms
Min. : 0.000
1st Qu.: 0.000
Median : 0.000
Mean : 7.602
3rd Qu.: 2.000
Max. :233.000

calculated_host_listings_count_shared_rooms	neighbourhood_group_cleansed
Min. : 0.00000	Eixample :6469

1st Qu.: 0.00000		Ciutat Vella	:4218
Median : 0.00000		Sants-Montjuïc	:1883
Mean : 0.08332		Sant Martí	:1662
3rd Qu.: 0.00000		Gràcia	:1591
Max. :12.00000		Sarrià-Sant Gervasi:	890
		(Other)	:1373

latitude	longitude	room_type	accommodates
Min. :41.35	Min. :2.092	Entire home/apt:10622	Min. : 1.000
1st Qu.:41.38	1st Qu.:2.157	Hotel room : 134	1st Qu.: 2.000
Median :41.39	Median :2.168	Private room : 7173	Median : 3.000
Mean :41.39	Mean :2.167	Shared room : 157	Mean : 3.365
3rd Qu.:41.40	3rd Qu.:2.177		3rd Qu.: 4.000
Max. :41.46	Max. :2.228		Max. :16.000

n_bathrooms	shared_bathrooms	bedrooms	beds
Min. : 0.000	Mode :logical	Min. : 1.000	Min. : 1.000
1st Qu.: 1.000	FALSE:13774	1st Qu.: 2.000	1st Qu.: 1.000
Median : 1.000	TRUE :4312	Median : 2.000	Median : 2.000
Mean : 1.309		Mean : 2.043	Mean : 2.351
3rd Qu.: 2.000		3rd Qu.: 2.000	3rd Qu.: 3.000
Max. :10.000		Max. :12.000	Max. :30.000

price	minimum_nights	maximum_nights	has_availability
Min. : 8.0	Min. : 1.00	Min. : 1.0	Mode :logical
1st Qu.: 52.0	1st Qu.: 1.00	1st Qu.: 180.2	FALSE:1188
Median :100.0	Median : 3.00	Median : 365.0	TRUE :16898
Mean :135.1	Mean : 14.76	Mean : 574.4	
3rd Qu.:182.0	3rd Qu.: 31.00	3rd Qu.:1125.0	
Max. :999.0	Max. :1125.00	Max. :3000.0	

availability_30	availability_60	availability_90	availability_365
Min. : 0.000	Min. : 0.00	Min. : 0.00	Min. : 0.0
1st Qu.: 0.000	1st Qu.: 0.00	1st Qu.: 1.00	1st Qu.: 47.0
Median : 4.000	Median :13.00	Median :35.00	Median :175.0
Mean : 7.718	Mean :19.37	Mean :36.43	Mean :171.8
3rd Qu.:12.000	3rd Qu.:34.00	3rd Qu.:62.00	3rd Qu.:302.0
Max. :30.000	Max. :60.00	Max. :90.00	Max. :365.0

instant_bookable	number_of_reviews	number_of_reviews_l30d	review_scores_rating
Mode :logical	Min. : 0.00	Min. : 0.0000	Min. :1.00
FALSE:11248	1st Qu.: 1.00	1st Qu.: 0.0000	1st Qu.:4.00
TRUE :6838	Median : 6.00	Median : 0.0000	Median :4.00
	Mean : 42.22	Mean : 0.9239	Mean :4.08

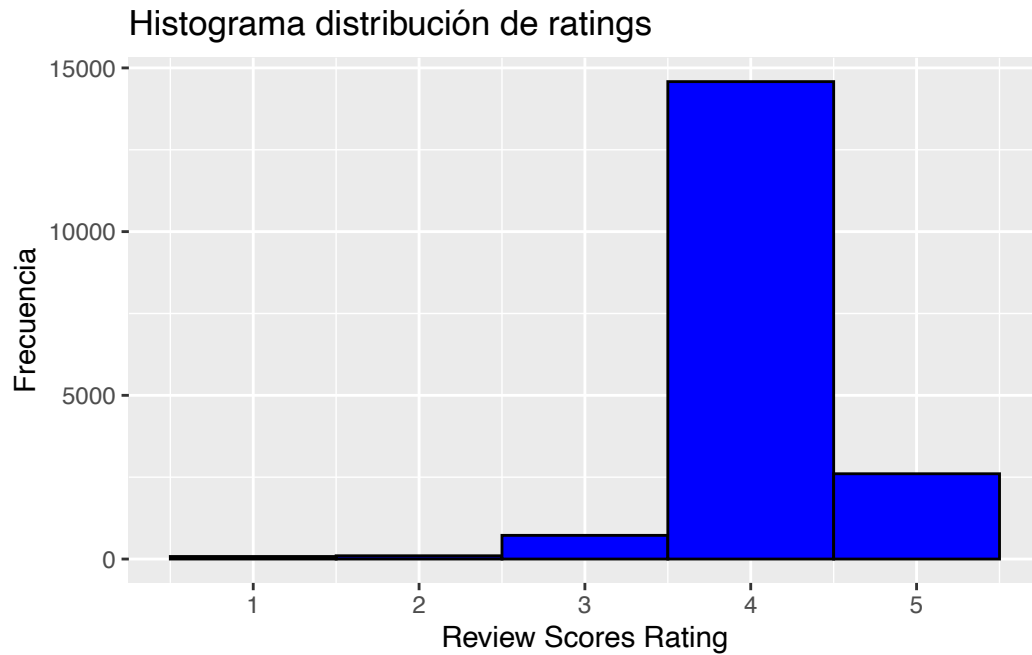
3rd Qu.:	42.00	3rd Qu.:	1.0000	3rd Qu.:	4.00
Max.	:1817.00	Max.	:91.0000	Max.	:5.00

review_scores_accuracy	review_scores_cleanliness	review_scores_checkin
Min. :0.0	Min. :0.000	Min. :0.000
1st Qu.:4.0	1st Qu.:4.000	1st Qu.:4.000
Median :4.0	Median :4.000	Median :4.000
Mean :4.1	Mean :4.085	Mean :4.165
3rd Qu.:4.0	3rd Qu.:4.000	3rd Qu.:4.000
Max. :5.0	Max. :5.000	Max. :5.000

review_scores_communication	review_scores_location	review_scores_value
Min. :0.000	Min. :0.000	Min. :0.000
1st Qu.:4.000	1st Qu.:4.000	1st Qu.:4.000
Median :4.000	Median :4.000	Median :4.000
Mean :4.165	Mean :4.167	Mean :4.009
3rd Qu.:4.000	3rd Qu.:4.000	3rd Qu.:4.000
Max. :5.000	Max. :5.000	Max. :5.000

Distribución de las notas de review

```
ggplot(df, aes(x = review_scores_rating)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black") +
  ggtitle("Histograma distribución de ratings") +
  xlab("Review Scores Rating") +
  ylab("Frecuencia")
```

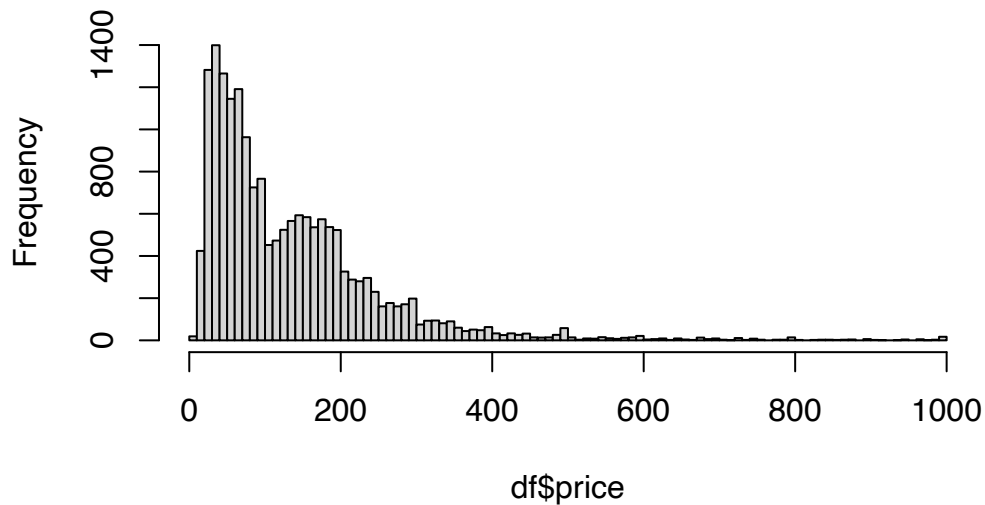


En el gráfico podemos ver como la gran mayoría de reseñas tienen un nivel de entre 4 y 5, siendo 4 con mucha diferencia la moda. Vemos como las notas entre 1 y 3 son mínimas.

Distribución del precio por noche

```
hist(df$price, main = "Histograma distribución precio", breaks=100)
```

Histograma distribución precio

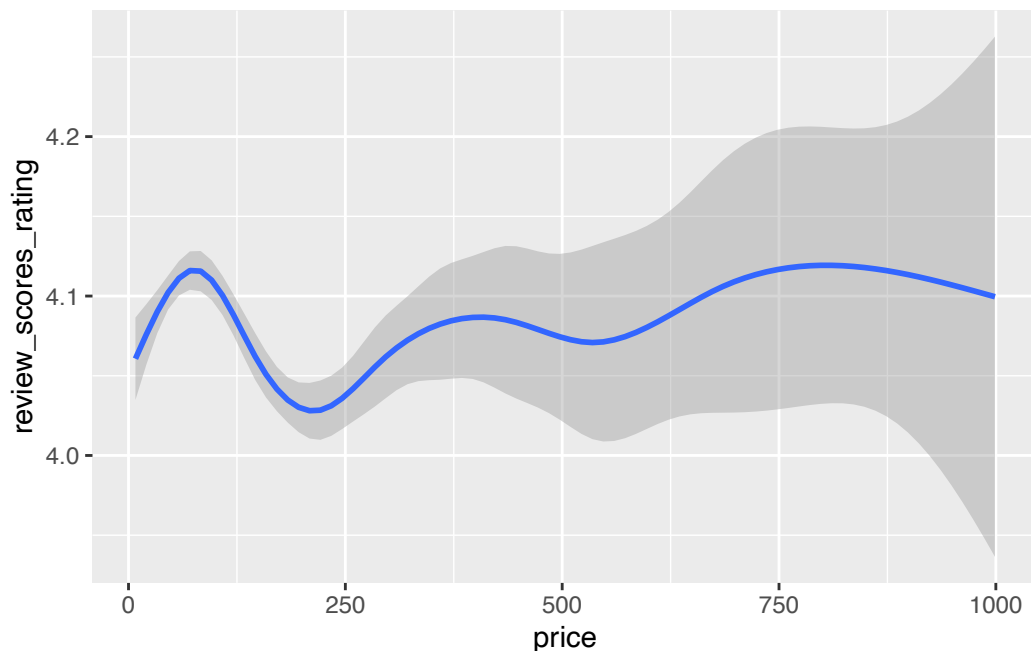


El rango de precios con mayor frecuencia esta entre 0 y 100, por otro lado existen pocas observaciones con precios superiores a 400. La distribución de los precios es asimétrica positiva (sesgada a la derecha), con una cola larga que se extiende hacia los precios más altos lo que implica que las frecuencias de precios disminuyen rápidamente a medida que aumenta el precio.

Distribución entre precio y reviews

```
ggplot(df, aes(price, review_scores_rating)) +  
  geom_smooth()
```

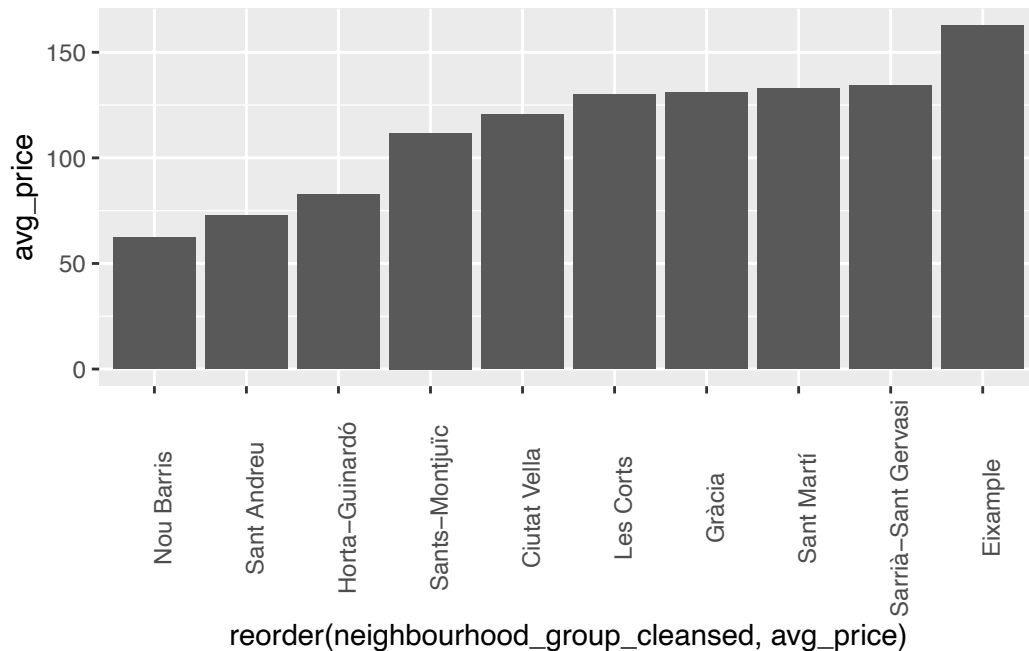
``geom_smooth()`` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'



Existe una tendencia que muestra que a medida que el precio aumenta, las calificaciones de revisión tienden a mejorar, especialmente después del punto de precio de 750. El intervalo de confianza se amplía a medida que el precio aumenta, lo que sugiere que hay más variabilidad en las calificaciones de revisión para productos o servicios más caros. La mayoría de los datos parece estar concentrada en el rango de precios más bajo.

Media de precio por barrio

```
df_precio_barrio <- df %>% select(price, neighbourhood_group_cleansed) %>% group_by(neighbourhood_group_cleansed) %>%
  summarise(avg_price = mean(price))
ggplot(df_precio_barrio, aes(reorder(neighbourhood_group_cleansed, avg_price), avg_price)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90))
```



Hay una variabilidad notable en los precios promedio entre los diferentes barrios. Eixample parece ser el barrio con el precio promedio más alto, seguido de cerca por Sarrià-Sant Gervasi. Por otro lado, Nou Barris parece ser el más económico en términos de precio promedio.

Separación de modelo en sets

Queremos separar nuestro modelo en 3 para poder ver la efectividad de este. Usaremos un set de entrenamiento, otro de test y otro de validación. A medida que avancemos iré comprobando la efectividad del test y una vez veamos que es efectivo haremos la validación global.

Como hemos podido ver en el análisis descriptivo encontrábamos mucha diferencia de valores en *price* dependiendo del barrio en el que estuvieran situadas. Por ello, vamos a hacer una distribución proporcional de valores en cada uno de los sets.

Primer retiramos las variables id para los sets

```
df <- df |>
  select(-id,
         -host_id)
```

```
set.seed(123)
```

```
training_test_split <- initial_split(df, prop = 0.8, strata = neighbourhood_group_cleaned)

# Crear los data frames de entrenamiento y el resto
training_data <- training(training_test_split)
test_data <- testing(training_test_split)
rm(training_test_split)
```

Primer modelo

Vamos a hacer una prueba con el primer modelo para ver que resultados obtenemos y como podemos mejorarlo aplicando técnicas estadísticas.

En este caso solo vamos a aplicar las variables numéricas.

```
training_data1 <- training_data %>%
  select_if(is.numeric)
```

Las aplicamos al modelo

```
modelo1 <- lm(price ~ ., data = training_data1)
```

Duplicamos test_data para esta prueba

```
test_data1 <- test_data
```

Predecimos con el test_data:

```
predicciones_modelo1 <- predict(modelo1, newdata = test_data1)
```

Añadimos las predicciones una columna para comparar

```
test_data1 <- test_data1 %>%
  mutate(predicciones = predicciones_modelo1)
```

Calcular las métricas resultantes la variable price

```
metrics1 <- test_data1 %>%
  metrics(truth = price, estimate = predicciones)
```

```
# Asegurarse de que 'predicciones' es un vector numérico sin nombres
test_data1$predicciones <- unname(test_data1$predicciones)
```

Calculamos el promedio de residuos y lo agregamos al data.frame de metrics

```
residuo1 <- sum(residuals(modelo1))

residuo1 <- c("residuo", "standard", residuo1)

metrics1 <- rbind(metrics1, residuo1)
```

Transformación de variables en valores numéricos y estandarización

A continuación, vamos a realizar cambios en las variables que no son de tipo numérico para poder ser incluidas en la regresión lineal:

- Para interactuar con la fecha nos vamos a quedar simplemente con el año de ingreso en la plataforma en `host_since`
- Escalamos las variables numéricas
- Convertiremos los booleanos a numéricos
- Usamos datos dummy para las columnas categóricas

```
training_data <- training_data |>
  mutate(host_since = as.numeric(format(host_since, "%Y"))) |>
  mutate(across(where(is.numeric), scale)) |>
  mutate(across(where(is.logical), as.numeric))

test_data <- test_data |>
  mutate(host_since = as.numeric(format(host_since, "%Y"))) |>
  mutate(across(where(is.numeric), scale)) |>
  mutate(across(where(is.logical), as.numeric))

training_data <- dummy_cols(training_data,
  select_columns = names(which(sapply(training_data, is.factor))),
  remove_selected_columns = TRUE)

test_data <- dummy_cols(test_data,
  select_columns = names(which(sapply(test_data, is.factor))),
  remove_selected_columns = TRUE)
```


Segundo modelo

```
# En este caso solo vamos a aplicar las variables numéricas.
training_data2 <- training_data

# Las aplicamos al modelo
modelo2 <- lm(price ~ ., data = training_data2)

# Duplicamos test_data para esta prueba
test_data2 <- test_data

# Predecimos con el test_data
predicciones_modelo2 <- predict(modelo2, newdata = test_data2)

# Añadimos las predicciones una columna para comparar
test_data2 <- test_data2 %>%
  mutate(predicciones = predicciones_modelo2)

# Calcular las métricas resultantes la variable price
metrics2 <- test_data2 %>%
  metrics(truth = price, estimate = predicciones)

# Asegurarse de que 'predicciones' es un vector numérico sin nombres
test_data2$predicciones <- unname(test_data2$predicciones)

# Calculamos el promedio de residuos y lo agregamos al data.frame de metrics
residuo2 <- sum(residuals(modelo2))

residuo2 <- c("residuo", "standard", residuo2)

metrics2 <- rbind(metrics2, residuo2)
```

Análisis de componentes principales

Vamos a realizar el análisis de componentes principales, ya que tenemos un total de 43 variables, entonces vamos a ver cuales son las mas relevantes para nuestro modelo:

Vamos a utilizar la función `pccomp` para aplicar el PCA

```
pca_result <- prcomp(training_data, center = TRUE, scale. = FALSE)

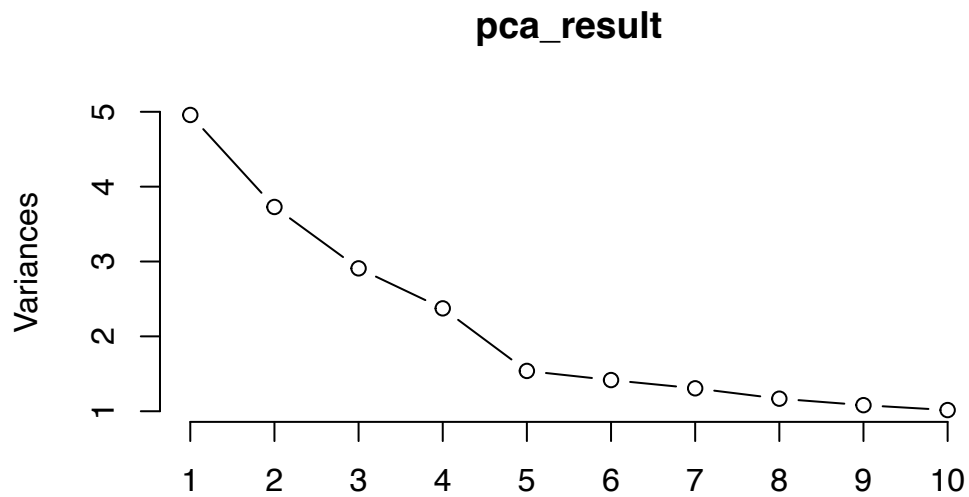
summary(pca_result)
```

Importance of components:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	2.2266	1.931	1.70541	1.54084	1.2403	1.19012	1.14289
Proportion of Variance	0.1569	0.118	0.09207	0.07516	0.0487	0.04484	0.04135
Cumulative Proportion	0.1569	0.275	0.36704	0.44220	0.4909	0.53574	0.57709
	PC8	PC9	PC10	PC11	PC12	PC13	PC14
Standard deviation	1.08057	1.03985	1.00771	0.95473	0.91907	0.87915	0.82879
Proportion of Variance	0.03696	0.03423	0.03215	0.02886	0.02674	0.02447	0.02174
Cumulative Proportion	0.61405	0.64828	0.68042	0.70928	0.73602	0.76049	0.78223
	PC15	PC16	PC17	PC18	PC19	PC20	PC21
Standard deviation	0.7969	0.77042	0.75043	0.71498	0.70163	0.64249	0.62454
Proportion of Variance	0.0201	0.01879	0.01783	0.01618	0.01558	0.01307	0.01235
Cumulative Proportion	0.8023	0.82113	0.83895	0.85513	0.87072	0.88379	0.89613
	PC22	PC23	PC24	PC25	PC26	PC27	PC28
Standard deviation	0.61392	0.53499	0.51869	0.50456	0.49769	0.46253	0.44662
Proportion of Variance	0.01193	0.00906	0.00852	0.00806	0.00784	0.00677	0.00631
Cumulative Proportion	0.90807	0.91713	0.92564	0.93370	0.94154	0.94832	0.95463
	PC29	PC30	PC31	PC32	PC33	PC34	PC35
Standard deviation	0.40400	0.38497	0.36813	0.35794	0.33608	0.33070	0.30716
Proportion of Variance	0.00517	0.00469	0.00429	0.00406	0.00358	0.00346	0.00299
Cumulative Proportion	0.95980	0.96449	0.96878	0.97283	0.97641	0.97987	0.98286
	PC36	PC37	PC38	PC39	PC40	PC41	PC42
Standard deviation	0.2919	0.27896	0.27066	0.24328	0.23427	0.18280	0.16969
Proportion of Variance	0.0027	0.00246	0.00232	0.00187	0.00174	0.00106	0.00091
Cumulative Proportion	0.9856	0.98802	0.99034	0.99221	0.99395	0.99501	0.99592
	PC43	PC44	PC45	PC46	PC47	PC48	PC49
Standard deviation	0.15306	0.14298	0.13266	0.12822	0.11115	0.10987	0.10642
Proportion of Variance	0.00074	0.00065	0.00056	0.00052	0.00039	0.00038	0.00036
Cumulative Proportion	0.99666	0.99731	0.99786	0.99839	0.99878	0.99916	0.99952
	PC50	PC51	PC52	PC53	PC54	PC55	
Standard deviation	0.08365	0.07668	0.04854	0.004806	5.147e-15	2.325e-15	
Proportion of Variance	0.00022	0.00019	0.00007	0.000000	0.000e+00	0.000e+00	
Cumulative Proportion	0.99974	0.99992	1.00000	1.000000	1.000e+00	1.000e+00	
	PC56						
Standard deviation	1.759e-15						
Proportion of Variance	0.000e+00						
Cumulative Proportion	1.000e+00						

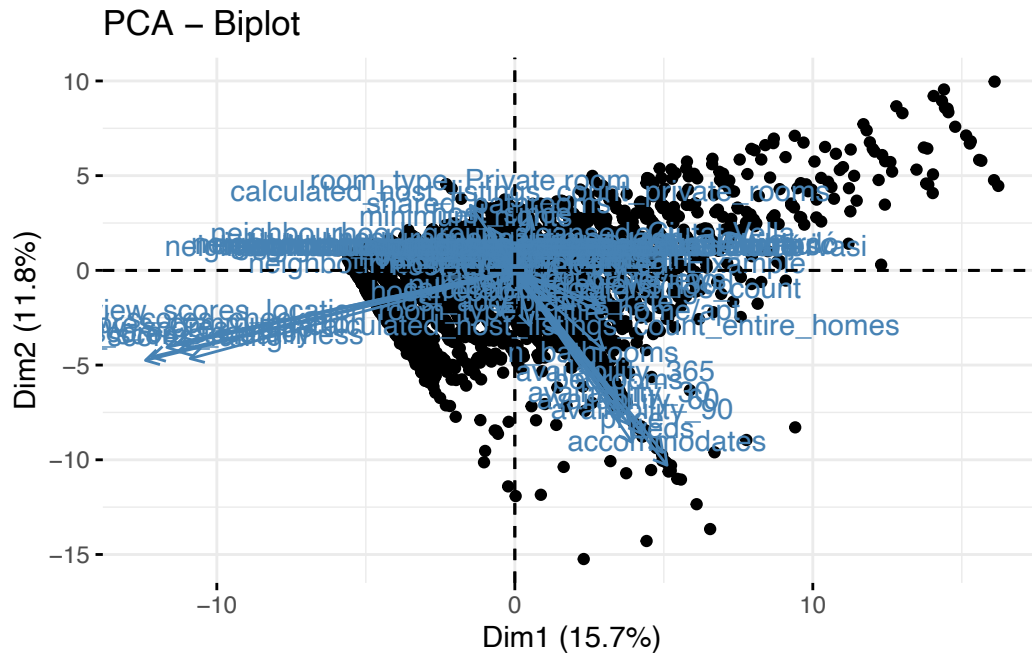
Vamos a hacer un Scree plot para ver cual es el número de componentes principales ideal

```
scree_plot <- plot(pca_result, type = "l")
```



Hacemos también un biplot para ver las variables correspondientes.

```
fviz_pca_biplot(pca_result,  
                 label="var")
```



Queremos visualizar las cargas de cada componente y ver que variables tienen mayor influencia.

```
loadings <- pca_result$rotation
```

```
# Ver las cargas para los primeros cinco componentes principales
```

```
loadings[, 1:5]
```

	PC1	PC2
host_since	-1.329794e-02	0.0600468772
host_response_rate	-1.506824e-02	-0.0827350674
host_acceptance_rate	1.510227e-02	-0.1012475554
host_is_superhost	-1.367036e-02	-0.0199229726
verification_email	2.763991e-03	-0.0082802230
verification_phone	5.304676e-04	-0.0011367906
verification_work_email	1.310210e-02	-0.0246635517
host_has_profile_pic	1.389175e-04	-0.0015329134
host_identity_verified	1.406812e-02	-0.0359118977
calculated_host_listings_count	8.885619e-02	-0.0777939849
calculated_host_listings_count_entire_homes	9.014873e-02	-0.1464410138
calculated_host_listings_count_private_rooms	1.582687e-02	0.1000974203
calculated_host_listings_count_shared_rooms	4.646351e-03	0.0003550779
latitude	2.280411e-03	0.0011033939

longitude	-1.012975e-02	0.0012180506
accommodates	1.542212e-01	-0.3582400287
n_bathrooms	7.952464e-02	-0.1948566849
shared_bathrooms	-2.628781e-02	0.0783844509
bedrooms	1.044553e-01	-0.2450313646
beds	1.498007e-01	-0.3346220856
price	1.188991e-01	-0.3161530926
minimum_nights	-4.972958e-02	0.0535750381
maximum_nights	2.083702e-02	-0.0232843628
has_availability	1.354768e-02	-0.0356863188
availability_30	1.058507e-01	-0.2689988576
availability_60	1.150830e-01	-0.2823692393
availability_90	1.290433e-01	-0.3000048211
availability_365	1.021080e-01	-0.2304152371
instant_bookable	3.610544e-02	-0.0533977259
number_of_reviews	8.571696e-02	-0.0620207138
number_of_reviews_l30d	5.699088e-02	-0.0704526095
review_scores_rating	-3.742444e-01	-0.1653872979
review_scores_accuracy	-3.681967e-01	-0.1590602539
review_scores_cleanliness	-3.288750e-01	-0.1621079173
review_scores_checkin	-3.415546e-01	-0.1331689109
review_scores_communication	-3.525812e-01	-0.1415901566
review_scores_location	-3.035424e-01	-0.1188669675
review_scores_value	-3.409034e-01	-0.1489737754
host_response_time_<1h	8.936130e-03	-0.0181814663
host_response_time_>48h	-8.314110e-04	0.0011442877
host_response_time_~12h	-3.948391e-03	0.0033565281
host_response_time_~24h	-4.156328e-03	0.0136806505
neighbourhood_group_cleansed_Ciutat Vella	-1.205371e-02	0.0236200658
neighbourhood_group_cleansed_Eixample	1.253710e-02	-0.0328179915
neighbourhood_group_cleansed_Gràcia	1.981800e-03	-0.0009039891
neighbourhood_group_cleansed_Horta-Guinardó	2.410852e-04	0.0051592680
neighbourhood_group_cleansed_Les Corts	-3.169725e-05	-0.0005420218
neighbourhood_group_cleansed_Nou Barris	-5.695354e-04	0.0023659183
neighbourhood_group_cleansed_Sant Andreu	-1.835902e-03	0.0022127315
neighbourhood_group_cleansed_Sant Martí	-1.560804e-03	-0.0014779770
neighbourhood_group_cleansed_Sants-Montjuïc	1.185339e-04	0.0038832191
neighbourhood_group_cleansed_Sarrià-Sant Gervasi	1.173125e-03	-0.0014992234
room_type_Entire home/apt	4.501571e-02	-0.1177308016
room_type_Hotel room	-6.299076e-04	-0.0014842007
room_type_Private room	-4.505813e-02	0.1197540981
room_type_Shared room	6.723325e-04	-0.0005390958
	PC3	PC4

host_since	-0.2088346520	0.0510071397
host_response_rate	-0.0539675008	-0.3521861395
host_acceptance_rate	-0.0936141586	-0.2398610348
host_is_superhost	0.0136690136	-0.0844453789
verification_email	0.0114193918	0.0025701272
verification_phone	-0.0006214859	-0.0001126510
verification_work_email	0.0321680337	0.0467680109
host_has_profile_pic	0.0092739799	0.0017763847
host_identity_verified	-0.0233973805	-0.0006013192
calculated_host_listings_count	0.0257247502	0.5280374449
calculated_host_listings_count_entire_homes	0.0811329621	0.4161606737
calculated_host_listings_count_private_rooms	-0.0867688093	0.2949745710
calculated_host_listings_count_shared_rooms	-0.0282702279	0.0011387480
latitude	0.0198275337	-0.0376246988
longitude	0.0090864595	-0.0616367743
accommodates	0.2895985923	-0.0254703211
n_bathrooms	0.1884428122	0.1139661282
shared_bathrooms	-0.0454644862	-0.0044994108
bedrooms	0.2760289457	0.0479025241
beds	0.2816140166	-0.0049099798
price	0.1599195379	-0.0683033624
minimum_nights	-0.0480298108	0.1499057048
maximum_nights	0.0617814506	-0.0759677650
has_availability	-0.0231607370	-0.0041115488
availability_30	-0.3980835741	0.0024607266
availability_60	-0.4338196680	0.0108998670
availability_90	-0.4105238328	-0.0063265263
availability_365	-0.2657484376	0.0635007750
instant_bookable	-0.0309782687	-0.0339443865
number_of_reviews	0.1040138507	-0.3353858360
number_of_reviews_l30d	0.0036729611	-0.2935354540
review_scores_rating	0.0059930924	0.0154329047
review_scores_accuracy	0.0121662861	0.0242194045
review_scores_cleanliness	-0.0052088973	0.0245073461
review_scores_checkin	-0.0084051610	0.0039899424
review_scores_communication	-0.0040230510	0.0092521951
review_scores_location	-0.0033231582	0.0589862970
review_scores_value	0.0281256056	-0.0003308220
host_response_time_<1h	0.0202009453	-0.0472507624
host_response_time_>48h	-0.0033893354	0.0023887155
host_response_time_~12h	-0.0023128998	0.0104544173
host_response_time_~24h	-0.0144987101	0.0344076295
neighbourhood_group_cleansed_Ciutat Vella	-0.0288505034	-0.0008779528

neighbourhood_group_cleansed_Eixample	0.0392741900	0.0198844444
neighbourhood_group_cleansed_Gràcia	-0.0020103956	-0.0013702546
neighbourhood_group_cleansed_Horta-Guinardó	-0.0019060571	-0.0032291175
neighbourhood_group_cleansed_Les Corts	0.0013907935	0.0005662008
neighbourhood_group_cleansed_Nou Barris	-0.0028508955	-0.0019151077
neighbourhood_group_cleansed_Sant Andreu	-0.0011854522	-0.0009796646
neighbourhood_group_cleansed_Sant Martí	0.0025652553	-0.0165772490
neighbourhood_group_cleansed_Sants-Montjuïc	-0.0038138222	-0.0043432135
neighbourhood_group_cleansed_Sarrià-Sant Gervasi	-0.0026131128	0.0088419145
room_type_Entire home/apt	0.0912746021	0.0054439093
room_type_Hotel room	-0.0025575390	-0.0028117113
room_type_Private room	-0.0859278026	-0.0006926291
room_type_Shared room	-0.0027892605	-0.0019395689
	PC5	
host_since	-0.2204167432	
host_response_rate	-0.0462318593	
host_acceptance_rate	0.1153871668	
host_is_superhost	0.0339606899	
verification_email	0.0114422775	
verification_phone	0.0010527661	
verification_work_email	0.0684347398	
host_has_profile_pic	0.0095338017	
host_identity_verified	0.0279872719	
calculated_host_listings_count	0.3282649777	
calculated_host_listings_count_entire_homes	0.4453477339	
calculated_host_listings_count_private_rooms	-0.1293635123	
calculated_host_listings_count_shared_rooms	-0.0622836078	
latitude	-0.0788645990	
longitude	-0.0362393383	
accommodates	-0.0861829186	
n_bathrooms	-0.3351919991	
shared_bathrooms	-0.0721513994	
bedrooms	-0.3182836001	
beds	-0.1534420560	
price	0.0297888200	
minimum_nights	-0.1667698788	
maximum_nights	0.0027743017	
has_availability	0.0270989318	
availability_30	-0.1067672933	
availability_60	-0.0565450039	
availability_90	0.0094036250	
availability_365	0.0586712414	
instant_bookable	0.0693660438	

number_of_reviews	0.3606523420
number_of_reviews_l30d	0.3560720235
review_scores_rating	0.0374610612
review_scores_accuracy	0.0336798345
review_scores_cleanliness	0.0495934344
review_scores_checkin	-0.0221275293
review_scores_communication	0.0088569349
review_scores_location	-0.0187894211
review_scores_value	0.0374091340
host_response_time_<1h	0.0545077367
host_response_time_>48h	-0.0070998954
host_response_time_~12h	-0.0202826472
host_response_time_~24h	-0.0271251942
neighbourhood_group_cleansed_Ciutat Vella	-0.0179801694
neighbourhood_group_cleansed_Eixample	0.0345009163
neighbourhood_group_cleansed_Gràcia	0.0081219490
neighbourhood_group_cleansed_Horta-Guinardó	-0.0063807623
neighbourhood_group_cleansed_Les Corts	-0.0009901357
neighbourhood_group_cleansed_Nou Barris	-0.0054226138
neighbourhood_group_cleansed_Sant Andreu	-0.0048427055
neighbourhood_group_cleansed_Sant Martí	-0.0128668951
neighbourhood_group_cleansed_Sants-Montjuïc	0.0137072066
neighbourhood_group_cleansed_Sarrià-Sant Gervasi	-0.0078467900
room_type_Entire home/apt	0.1062638271
room_type_Hotel room	0.0016725317
room_type_Private room	-0.1045304097
room_type_Shared room	-0.0034059491

```
# Convertir la matriz de cargas en un data frame
loadings_df <- as.data.frame(loadings) |>
  select(PC1, PC2, PC3, PC4, PC5)

# Sumamos los PC de cada variable
loadings_df$Suma <- rowSums(loadings_df[, c("PC1", "PC2", "PC3", "PC4", "PC5")])

loadings_df_sorted <- loadings_df |>
  arrange(desc(Suma))
```

El resultado es bastante sorprendente, pero tiene sentido vemos que las variables mas relevantes en primer lugar son el número de propiedades que tiene el host en si. Esto se puede deber a que a mayor número de propiedades hablamos de una dedicación profesional y conlleva un aumento de precios en sus alojamientos en general. Otras variables como el número de reviews influencia también.

Vamos a crear una lista con las variables mas relevantes, para el siguiente modelo probaremos con las variables cuya suma de componentes sea mayor que 0:

```
loadings_df <- loadings_df_sorted |>
  filter(Suma > 0)

seleccion_variables <- rownames(loadings_df)
seleccion_variables <- c(seleccion_variables, "price")
```

Tercer modelo

Vamos a hacer un modelo solo con las variables seleccionadas:

```
# En este caso solo vamos a aplicar las variables numéricas.
training_data3 <- training_data |>
  select(seleccion_variables)
```

Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
i Please use `all_of()` or `any_of()` instead.

Was:

```
data %>% select(seleccion_variables)
```

Now:

```
data %>% select(all_of(seleccion_variables))
```

See <<https://tidyselect.r-lib.org/reference/faq-external-vector.html>>.

```
# Las aplicamos al modelo
modelo3 <- lm(price ~ ., data = training_data3)

# Duplicamos test_data para esta prueba
test_data3 <- test_data |>
  select(seleccion_variables)

# Predecimos con el test_data
predicciones_modelo3 <- predict(modelo3, newdata = test_data3)

# Añadimos las predicciones una columna para comparar
test_data3 <- test_data3 %>%
  mutate(predicciones = predicciones_modelo2)
```

```
# Calcular las métricas resultantes la variable price
metrics3 <- test_data3 %>%
  metrics(truth = price, estimate = predicciones)

# Asegurarse de que 'predicciones' es un vector numérico sin nombres
test_data3$predicciones <- unname(test_data3$predicciones)

# Calculamos el promedio de residuos y lo agregamos al data.frame de metrics
residuo3 <- sum(residuals(modelo3))

residuo3 <- c("residuo", "standard", residuo3)

metrics3 <- rbind(metrics3, residuo3)
```

Cuarto y quinto modelo

Vamos a hacer dos modelos mas: Uno con una selección mas amplia de variables y otra con una selección mas corta, al final de todo comparemos todos los modelos y sacaremos conclusiones.

Cuarto modelo

Vamos a hacer una selección mas corta de variables filtrando por 0.10

```
loadings_df <- loadings_df_sorted |>
  filter(Suma > 0.10)

seleccion_variables <- rownames(loadings_df)
seleccion_variables <- c(seleccion_variables, "price")
```

```
# En este caso solo vamos a aplicar las variables numéricas.
training_data4 <- training_data |>
  select(seleccion_variables)

# Las aplicamos al modelo
modelo4 <- lm(price ~ ., data = training_data4)

# Duplicamos test_data para esta prueba
test_data4 <- test_data |>
  select(seleccion_variables)
```

```

# Predecimos con el test_data
predicciones_modelo4 <- predict(modelo4, newdata = test_data4)

# Añadimos las predicciones una columna para comparar
test_data4 <- test_data4 %>%
  mutate(predicciones = predicciones_modelo4)

# Calcular las métricas resultantes la variable price
metrics4 <- test_data4 %>%
  metrics(truth = price, estimate = predicciones)

# Asegurarse de que 'predicciones' es un vector numérico sin nombres
test_data4$predicciones <- unname(test_data4$predicciones)

# Calculamos el promedio de residuos y lo agregamos al data.frame de metrics
residuo4 <- sum(residuals(modelo4))

residuo4 <- c("residuo", "standard", residuo4)

metrics4 <- rbind(metrics4, residuo4)

```

Quinto modelo

Ahora haremos una selección mas larga filtrando por -0.10

```

loadings_df <- loadings_df_sorted |>
  filter(Suma > -0.20)

seleccion_variables <- rownames(loadings_df)
seleccion_variables <- c(seleccion_variables, "price")

```

```

training_data5 <- training_data |>
  select(seleccion_variables)

# Las aplicamos al modelo
modelo5 <- lm(price ~ ., data = training_data5)

# Duplicamos test_data para esta prueba
test_data5 <- test_data |>

```

```

select(seleccion_variables)

# Predecimos con el test_data
predicciones_modelo5 <- predict(modelo5, newdata = test_data5)

# Añadimos las predicciones una columna para comparar
test_data5 <- test_data5 %>%
  mutate(predicciones = predicciones_modelo5)

# Calcular las métricas resultantes la variable price
metrics5 <- test_data5 %>%
  metrics(truth = price, estimate = predicciones)

# Asegurarse de que 'predicciones' es un vector numérico sin nombres
test_data5$predicciones <- unname(test_data5$predicciones)

# Calculamos el promedio de residuos y lo agregamos al data.frame de metrics
residuo5 <- sum(residuals(modelo5))

residuo5 <- c("residuo", "standard", residuo5)

metrics5 <- rbind(metrics5, residuo5)

```

Comparativa de modelos

Unificamos las métricas de los 5 modelos:

```

metrics <- bind_rows(metrics1, metrics2, metrics3, metrics4, metrics5) |>
  select(-.estimator)
n_modelo <- rep(seq(5), each = 4)
metrics$n_modelo <- n_modelo

metrics <- metrics %>%
  pivot_wider(names_from = .metric, values_from = .estimate)

print(metrics)

```

```

# A tibble: 5 x 5
  n_modelo rmse          rsq          mae          residuo
  <int> <chr>          <chr>          <chr>          <chr>

```

1	1	92.2370396724672	0.395825071402194	55.1237606977542	-8.12008238426~
2	2	0.750713817706505	0.43642217949015	0.444070417275347	-1.05693231944~
3	3	0.750713817706505	0.43642217949015	0.444070417275347	-1.49963375051~
4	4	0.898035879986293	0.193568100662718	0.56628162831109	5.153100168797~
5	5	0.763072846484759	0.417753707452463	0.455242020155687	-4.17055279200~

Vamos a comentar cada modelo basándonos en las métricas proporcionadas y luego seleccionaremos el mejor modelo.

Modelo 1

- El RMSE es muy alto, lo que indica que los errores de predicción son, en promedio, grandes.
- R^2 es razonablemente alto, lo que sugiere que el modelo explica una buena cantidad de la varianza de los datos.
- MAE es bastante alto, lo que indica errores significativos en la predicción.
- El residuo es negativo y muy cercano a cero, lo que sugiere que en promedio, el modelo podría estar subestimando ligeramente las predicciones.

Modelo 2

- El RMSE es considerablemente más bajo que el del Modelo 1, lo que indica una reducción significativa en los errores de predicción en comparación con el modelo que no utiliza normalización ni variables dummy.
- El R^2 , es menor que el del Modelo 1. Un valor más bajo aquí sugiere que, aunque las predicciones son más precisas en términos de error cuadrático, la proporción de la varianza total que el modelo puede explicar ha disminuido.
- El MAE reducido en el Modelo 2 refleja la mejora en la precisión de las predicciones después de la normalización y la inclusión de variables dummy. Como el MAE no da tanto peso a los errores más grandes como el

Modelo 3

- El RMSE y el MAE son idénticos a los del Modelo 2, lo que indica que la selección de componentes no perjudicó la precisión de la predicción.
- El R^2 permanece sin cambios respecto al Modelo 2, lo que sugiere que la cantidad de varianza explicada por el modelo es similar.
- El residuo es negativo y muy pequeño, lo cual es consistente con el Modelo 2.

Modelo 4

- Este modelo muestra un RMSE ligeramente más alto y un R^2 ligeramente más bajo que los Modelos 2 y 3, lo que indica que la eliminación de algunas variables mediante un umbral de carga más alto ha tenido un pequeño impacto negativo en el rendimiento del modelo.
- El MAE es ligeramente más alto, lo que sugiere que las predicciones son menos precisas en promedio.
- El residuo es positivo, lo que indica que este modelo podría estar sobreestimando ligeramente las predicciones en promedio.

Modelo 5

- Este modelo tiene la menor RMSE y MAE de todos los modelos, lo que sugiere que tiene el mejor rendimiento en términos de precisión de la predicción.
- El R^2 es también el más alto, indicando que este modelo explica la mayor parte de la varianza en los datos.
- El residuo es negativo, pero muy cercano a cero, similar a los Modelos 2 y 3.

Selección del modelo

El Modelo 5 parece ser el mejor en términos de todas las métricas proporcionadas. Tiene el RMSE más bajo, lo que indica que tiene el menor error de predicción promedio. También tiene el MAE más bajo, lo que sugiere que es consistente en su precisión a través de diferentes muestras. Además, el R^2 más alto indica que explica la mayor cantidad de varianza en la variable dependiente comparado con los otros modelos. Aunque el residuo es negativo, lo cual puede implicar una leve subestimación, está muy cerca de cero, lo que sugiere que en promedio, el sesgo del modelo es mínimo.

La inclusión de más variables (con cargas mayores de -0.10) en el Modelo 5 parece haber capturado mejor la complejidad subyacente de los datos, lo que ha resultado en un modelo más preciso.

Sin embargo, en la práctica, se recomendaría realizar una validación cruzada o pruebas en un conjunto de datos de prueba separado para evaluar mejor la capacidad del modelo para generalizar a nuevos datos antes de finalizar la selección del modelo.