Proyecto Final Big Data

Bootcamp Glovo Mujeres en Tech- KeepCoding.

Desarrollo de herramienta de búsqueda en Airbnb ‘’Encuentra tu alojamiento en Madrid.’’

Grupo 9

Brissette Berrios

Diana Antón

Giuliana Scoppettone

Alejandra Macedo

Irene de Freitas

Idea general

Realizar una herramienta que responde a las necesidades de búsqueda de un usuario de Airbnb que desea realizar una reserva en Madrid.

Suposiciones iniciales

Tenemos una base de datos en Excel, que contiene una serie de datos relacionados a inmuebles de alquiler en la plataforma de Airbnb, de la cual debemos extraer las columnas que nos permitan llevar a cabo el objetivo de desarrollar.

Para lograr esto, debemos tomar en consideración el tipo de alojamiento, su ubicación y su precio.

Debemos tomar en cuenta los detalles acerca del perfil del Host para que el usuario pueda filtrar por estos datos. Esta suposición resultó ser incorrecta en la práctica, ya que estos datos no aportan información vinculante en cuanto al tipo de alojamiento y precio.

Los precios y valoraciones no podían ser utilizados con los valores establecidos en el dataset, ya que no arrojaban datos fiables a la hora de realizar valoraciones de totales. Es por ello que ha tenido que ser calculada la media de cada uno de estos.

Métricas

Se han tomado en consideración: Barrio, precio, tipo de alojamiento, Fianza y otros gastos, reseñas, A pesar de que estas métricas han sido correctas, luego hemos tenido que añadir otros adicionales como camas baño, cantidad de huéspedes y tipo de habitación como datos complementarios para poder tener mayor cantidad de datos en el desarrollo del modelo de regresión lineal.

Arquitectura y validación de los datos

a. Muestreo y exploración inicial de los datos:

Se escogieron las siguientes columnas:

['ID', 'Name', 'Listing Url', 'Host ID', 'Neighbourhood Group Cleansed', 'City', 'State', 'Zipcode', 'Country Code', 'Country', 'Latitude', 'Longitude', 'Property Type', 'Room Type', 'Square Feet', 'Price', 'Weekly Price', 'Monthly Price', 'Security Deposit', 'Cleaning Fee', 'Number of Reviews', 'Review Scores Rating', 'Review Scores Accuracy', 'Review Scores Cleanliness', 'Review Scores Checkin', 'Review Scores Communication', 'Review Scores Location', 'Review Scores Value', 'Cancellation Policy', 'Accommodates', 'Bathrooms', 'Bedrooms', 'Beds', 'Host ID', 'Host URL', 'Host

Name', 'Host Since', 'Host Location', 'Host About', 'Host Response Time', 'Host Response Rate', 'Host Acceptance Rate', 'Host Thumbnail Url', 'Host Picture Url', 'Host Neighbourhood', 'Host Listings Count', 'Host Total Listings Count', 'Host Verifications']

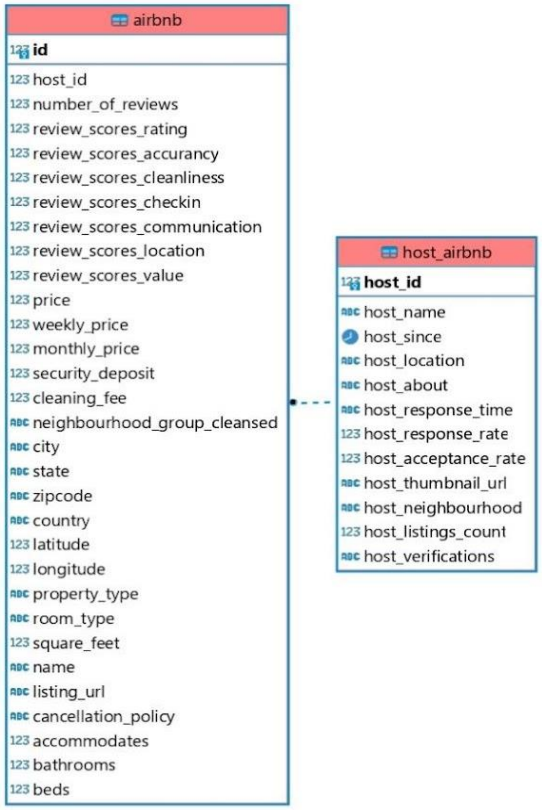
- Se identificó diferentes idiomas en las columnas de City, State, Country.
- Hay celdas en blanco por ejemplo en las columnas de State, City y Price.
- Las fechas tienen diferentes formatos.
- Hay datos de diferentes países.

b. Definir e implementar el Datawarehouse:

- Se creó un diagrama de Entidad-Relación con Drawio.
- Luego se creó el script en DBeaver, donde luego también aparece el diagrama de ER.

Diagrama con Drawio

Diagrama con DBeaver



Script:

--creamos un esquema

create schema Encuentra_tu_Airbnb_Madrid authorization postgres;

--Añadimos la estructura

--Tabla de Host

```
create table Encuentra_tu_Airbnb_Madrid.Host_Airbnb(
host_id int not null, --PK
host_name varchar (300) null,
host_since date null,
host_location varchar (300) null,
host_about varchar (300) null,
host_response_time varchar (300) null,
host_response_rate int not null,
host_acceptance_rate int not null,
host_thumbnail_url varchar (300) not null,
host_neighbourhood varchar (300) not null,
host_listings_count int not null,
host_verifications varchar (300) not null
);

--ahora añadimos los PK
alter table encuentra_tu_airbnb_madrid.host_airbnb
add constraint host_airbnb_PK primary key (host_id);

--tabla de airbnb

create table encuentra_tu_airbnb_madrid.airbnb(
id int not null, --PK
host_id int not null, --FK
number_of_reviews int null,
review_scores_rating int null,
review_scores_accurancy int null,
review_scores_cleanliness int null,
review_scores_checkin int null,
review_scores_communication int null,
review_scores_location int null,
review_scores_value int null,
price float null,
weekly_price float null,
monthly_price float null,
```

```
security_deposit float null,
cleaning_fee float null,
neighbourhood_group_cleansed varchar (300) null,
city varchar (300) null,
state varchar (300) null,
zipcode varchar (300) null,
country varchar (300) null,
latitude float null,
longitude float null,
property_type varchar (300) null,
room_type varchar (300) null,
square_feet float null,
name varchar (300) null,
listing_url varchar (300) null,
cancellation_policy varchar (300) null,
accommodates int null,
bathrooms float null,
beds int null

);

--ahora añadimos los PK y FK

alter table encuentra_tu_airbnb_madrid.airbnb
add constraint airbnb_PK primary key (id);


alter table encuentra_tu_airbnb_madrid.airbnb
add constraint airbnb_FK foreign key (host_id)
references encuentra_tu_airbnb_madrid.host_airbnb(host_id);
```

Limpieza y calidad de datos

Tenemos un dataset con:
Hacemos un df. shape para ver su forma y vemos que tiene 14780 rows y 89 columns.

Hacemos un df.columns.values para ver los nombres de las columnas y decidir cuáles nos interesan:

[89 columns]
['ID' 'Listing Url' 'Scrape ID' 'Last Scraped' 'Name' 'Summary' 'Space'
 'Description' 'Experiences Offered' 'Neighborhood Overview' 'Notes'
 'Transit' 'Access' 'Interaction' 'House Rules' 'Thumbnail Url'
 'Medium Url' 'Picture Url' 'XL Picture Url' 'Host ID' 'Host URL'
 'Host Name' 'Host Since' 'Host Location' 'Host About'
 'Host Response Time' 'Host Response Rate' 'Host Acceptance Rate'
 'Host Thumbnail Url' 'Host Picture Url' 'Host Neighbourhood'
 'Host Listings Count' 'Host Total Listings Count' 'Host Verifications'
 'Street' 'Neighbourhood' 'Neighbourhood Cleansed'
 'Neighbourhood Group Cleansed' 'City' 'State' 'Zipcode' 'Market'
 'Smart Location' 'Country Code' 'Country' 'Latitude' 'Longitude'
 'Property Type' 'Room Type' 'Accommodates' 'Bathrooms' 'Bedrooms' 'Beds'
 'Bed Type' 'Amenities' 'Square Feet' 'Price' 'Weekly Price'
 'Monthly Price' 'Security Deposit' 'Cleaning Fee' 'Guests Included'
 'Extra People' 'Minimum Nights' 'Maximum Nights' 'Calendar Updated'
 'Has Availability' 'Availability 30' 'Availability 60' 'Availability 90'
 'Availability 365' 'Calendar last Scraped' 'Number of Reviews'
 'First Review' 'Last Review' 'Review Scores Rating'
 'Review Scores Accuracy' 'Review Scores Cleanliness'
 'Review Scores Checkin' 'Review Scores Communication'
 'Review Scores Location' 'Review Scores Value' 'License'
 'Jurisdiction Names' 'Cancellation Policy'
 'Calculated host listings count' 'Reviews per Month' 'Geolocation'
 'Features']

Ante cualquier duda sobre el dato que contienen hacemos un df ['Nombre de columna']. head() para ver una muestra de lo que contiene.

Por nuestros objetivos decidimos seleccionar las siguientes columnas:

['ID', 'Name', 'Listing Url', 'Host ID', 'Neighbourhood Group Cleansed', 'City', 'State', 'Zipcode', 'Country Code', 'Country', 'Latitude', 'Longitude', 'Property Type', 'Room Type', 'Square Feet', 'Price', 'Weekly Price', 'Monthly Price', 'Security Deposit', 'Cleaning Fee', 'Number of Reviews', 'Review Scores Rating', 'Review Scores Accuracy', 'Review Scores Cleanliness', 'Review Scores Checkin', 'Review Scores Communication', 'Review Scores Location', 'Review Scores Value', 'Cancellation Policy', 'Accommodates', 'Bathrooms', 'Bedrooms', 'Beds', 'Host ID', 'Host URL', 'Host Name', 'Host Since', 'Host Location', 'Host About', 'Host Response Time', 'Host Response Rate', 'Host Acceptance Rate', 'Host Thumbnail Url', 'Host Picture Url', 'Host Neighbourhood', 'Host Listings Count', 'Host Total Listings Count', 'Host Verifications']

Inspección general de los datos

Hacemos un df.dtypes para ver de qué tipo son nuestros datos y obtenemos:

ID	int64
Listing Url	object
Host ID	int64
Neighbourhood Group Cleansed	object
City	object
State	object
Zipcode	object

Country Code	object
Country	object
Latitude	float64
Longitude	float64
Property Type	object
Room Type	object
Square Feet	float64
Price	float64
Weekly Price	float64
Monthly Price	float64
Security Deposit	float64
Cleaning Fee	float64
Number of Reviews	float64
Review Scores Rating	float64
Review Scores Cleanliness	float64
Review Scores Checkin	float64
Review Scores Communication	float64
Review Scores Location	float64
Review Scores Value	float64
Cancellation Policy	object

dtype:
Hacemos un df.describe general para ver nuestros datos:

	ID	Host ID	Latitude	Longitude	...	Review Scores Checkin	Review Scores Communication	Review Scores Location	Review Scores Va
lue									
count	1.478000e+04	1.478000e+04	14780.000000	14780.000000	...	11443.000000	11460.000000	11440.000000	11439.000
000									
mean	1.028089e+07	3.608080e+07	40.497626	-3.858041	...	9.621778	9.647033	9.532168	9.218
201									
std	5.564829e+06	3.425360e+07	4.641387	14.123146	...	0.802736	0.767116	0.774527	0.950
578									
min	1.862800e+04	1.745300e+04	-37.851182	-123.131344	...	2.000000	2.000000	2.000000	2.000
000									
25%	5.554732e+06	6.787360e+06	40.409726	-3.707604	...	9.000000	9.000000	9.000000	9.000
000									
50%	1.133492e+07	2.464875e+07	40.419466	-3.700785	...	10.000000	10.000000	10.000000	9.000
000									
75%	1.532631e+07	5.432919e+07	40.430916	-3.684057	...	10.000000	10.000000	10.000000	10.000
000									
max	1.910969e+07	1.247534e+08	55.966912	153.371427	...	10.000000	10.000000	10.000000	10.000
000									

De momento, no nos da mucha información útil, ya que tenemos variables numéricas cuyos cálculos básicos no son relevantes. Es por ello que utilizaremos este método de nuevo más adelante, con columnas específicas.

Limpieza de datos NaN

Vemos el número de NaN de las columnas que hemos elegido para comprobar su calidad, con df.isna().sum(). Además, debemos decidir qué haremos con los valores NA de cualquier columna.

ID	0
Name	1
Listing Url	0
Host ID	0
Neighbourhood Group Cleansed	1020
City	6
State	144
Zipcode	506
Country Code	0
Country	1
Latitude	0
Longitude	0
Property Type	0
Room Type	0
Square Feet	14182
Price	17
Weekly Price	11190
Monthly Price	11219
Security Deposit	8524
Cleaning Fee	6093
Number of Reviews	0
Review Scores Rating	3304
Review Scores Accuracy	3326
Review Scores Cleanliness	3320
Review Scores Checkin	3337
Review Scores Communication	3320
Review Scores Location	3340
Review Scores Value	3341
Cancellation Policy	0
Accommodates	0
Bathrooms	55
Bedrooms	25
Beds	49
Host ID	0
Host URL	0
Host Name	3
Host Since	3
Host Location	43
Host About	5241
Host Response Time	1899
Host Response Rate	1899
Host Acceptance Rate	14741
Host Thumbnail Url	3
Host Picture Url	3
Host Neighbourhood	3876
Host Listings Count	3
Host Total Listings Count	3

Host Verifications	6
dtype: int64	

En base a esta información, decidimos deshacernos de las columnas que hemos coloreado en rojo, ya que la cantidad de datos que tienen no es suficiente para considerarse relevante, o la información no será realmente de utilidad para nuestros objetivos.

Con las columnas en amarillo quitamos los valores NaN, ya que observándolos detenidamente, son datos que no son de calidad.

A continuación, sustituimos por 0 los valores NA de las columnas en naranja, Security Deposit y Cleaning Fee. Ya que consideramos que si no cobran estos extras es porque ya están incluidos en el precio general.

Para las columnas en morado sobre Reviews, le añadimos 0 a los que tienen NA, para poder hacer cálculos sobre ellas, por otra parte creamos una nueva columna con la media de esas reviews, para así tener un dato fiable y que entendemos, ya que los datos que parecen el conjunto de estas valoraciones no parecen cuadrar.

Con las columnas en azul, sobre el weekly price y el monthly price, lo que hacemos es crear una columna con el precio calculado por semana o mes, basándonos en el precio por día. Tras esto creamos dos nuevas columnas, Weekly Price New y Monthly Price New que contienen los precios establecidos por los propietarios y los precios calculados por nosotras.

Por último, para la columna en verde, Host Verifications, cambiamos los valores que tienen por Yes o No. Yes será para las filas que tienen datos de verificación, y No para las que tienen valores NA.

El resto de las columnas NA decicimos dejarlas como están, ya que no afectará a nuestras predicciones.

Tras la limpieza de las columnas, nos quedamos con un Dataset de filas13306 y 45 columnas, con los siguientes datos NA.

ID	0
Name	0
Listing Url	0
Host ID	0
Neighbourhood Group Cleansed	0
City	2
State	42
Zipcode	441
Country Code	0
Country	0
Latitude	0
Longitude	0
Property Type	0
Room Type	0
Price	0
Weekly Price	9945
Monthly Price	9988

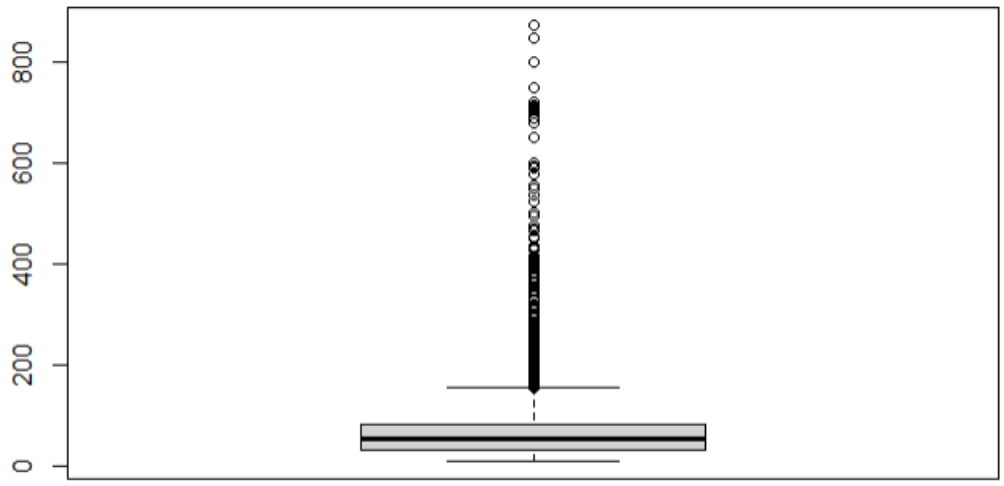
Security Deposit	0
Cleaning Fee	0
Number of Reviews	0
Review Scores Rating	0
Review Scores Accuracy	0
Review Scores Cleanliness	0
Review Scores Checkin	0
Review Scores Communication	0
Review Scores Location	0
Review Scores Value	0
Cancellation Policy	0
Accommodates	0
Bathrooms	48
Bedrooms	23
Beds	48
Host ID	0
Host URL	0
Host Name	0
Host Since	0
Host Verifications	0
Extras	0
Weekly Price Calculated	0
Weekly Price New	0
Monthly Price Calculated	0
Monthly Price New	0
Monthly Price Discounted	0
Weekly Price Discounted	0
Reviews Mean	0
dtype: int64	

Limpieza de outliers

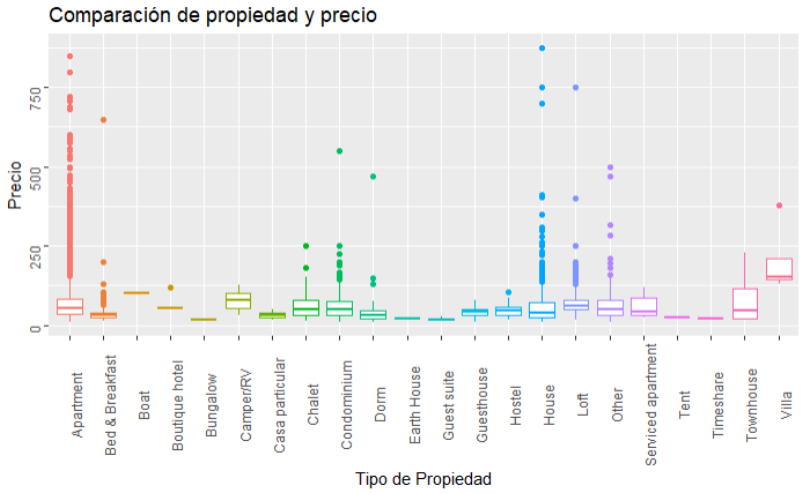
Para esta parte usaremos R, ya que las opciones de gráficas son mayores, y sus limitaciones menores que python.

Empezamos haciendo un boxplot del precio, ya que consideramos que es la forma más visual de ver los valores que se alejan mucho de la media, o son muy diferentes al resto.
Obtenemos lo siguiente:

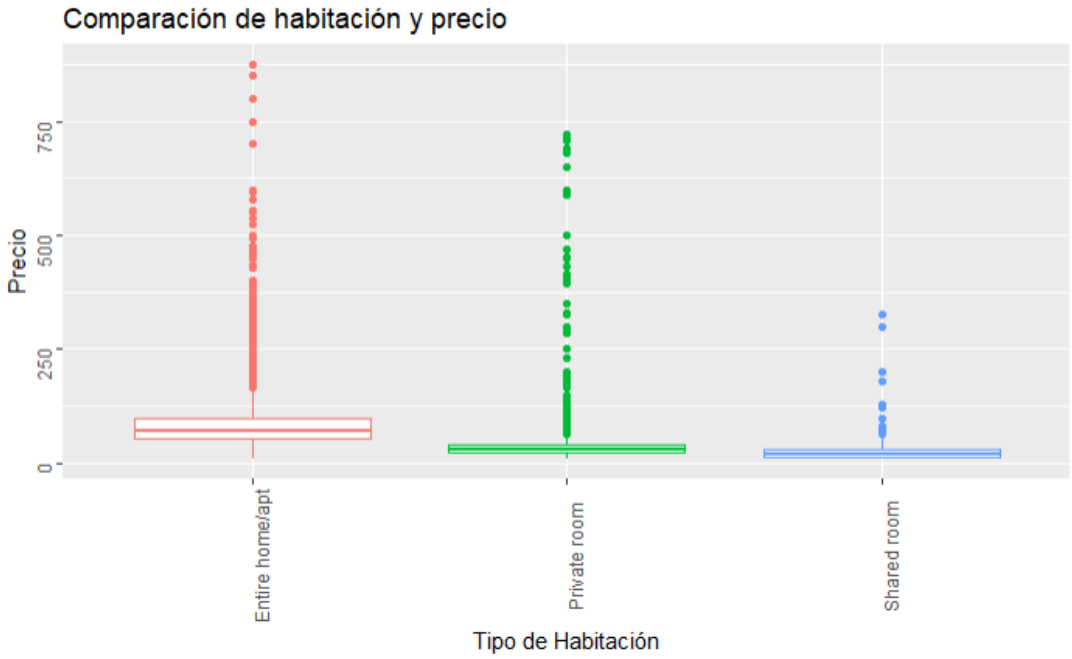
```{r}
boxplot(airbnbmadrid$Price)

```



Este boxplot no nos da suficiente información para tomar decisiones acerca de los datos.
Por ello hacemos otro boxplot en base al tipo de propiedad:



Este boxplot es más interesante, porque podemos ver claros outliers según una característica compartida.
Además, también comparamos el tipo de habitación con el precio:



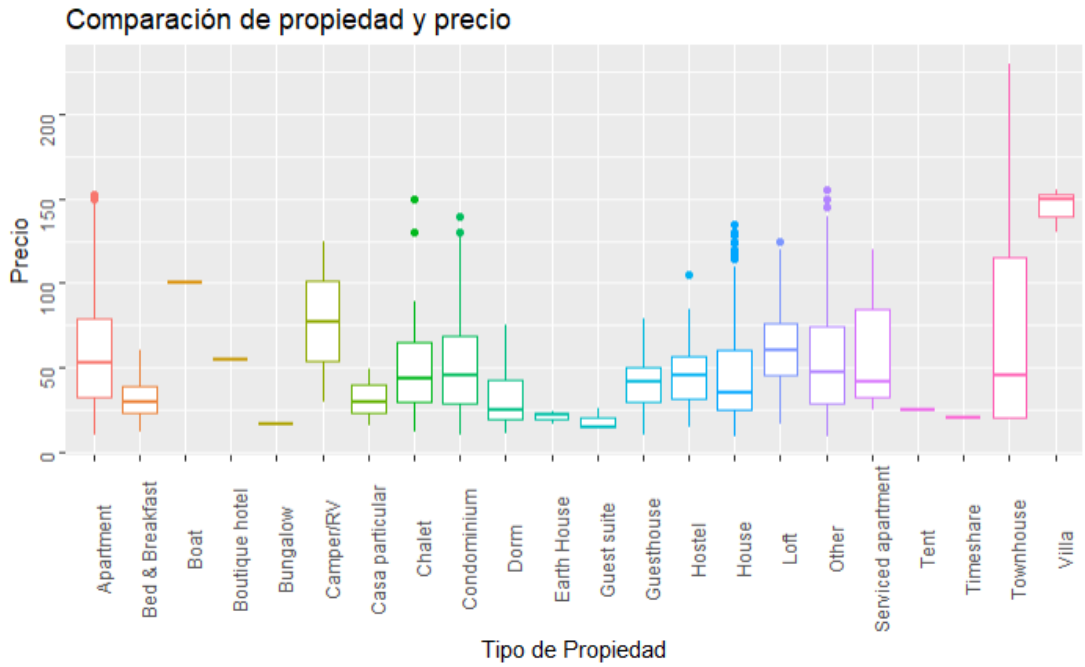
Según los datos que hemos observado, en la clasificación según el tipo de propiedad es más identificable qué valores que podrían ser claramente outliers.

Así que, miramos cual es el precio mínimo posible para ver si es algo plausible, y si nos convendría hacer una limpieza de los datos máximos y mínimos.

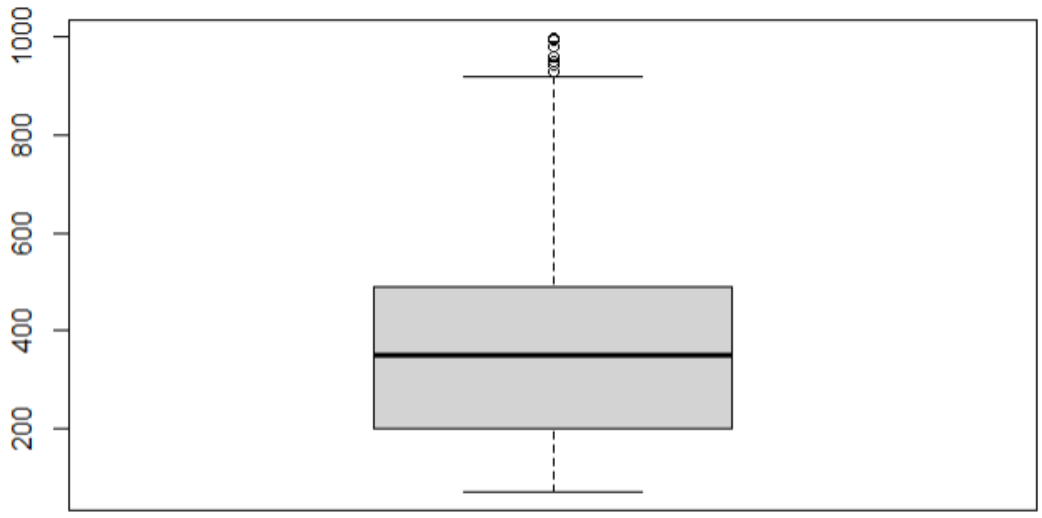
count 13303.000000
mean 67.319251
std 61.303384
min 9.000000
25% 31.000000
50% 52.000000
75% 80.000000
max 875.000000
Name: Price, dtype: float64

Con lo obtenido, vemos que el mínimo es más similar a la mayoría de los datos que el precio máximo, el cual se aleja mucho del cuartil 75. Por lo que decidimos limpiar las columnas siguientes mediante el método de eliminar 1.5 veces la distancia iqr, de los valores mas altos de los siguientes tipos de propiedad: Villa, Other, Loft, House, Dorm, Condominium, Chalet, Bed and Breakfast, Apartment, Boutique Hotel.

Tras esto obtenemos los siguientes boxplots:

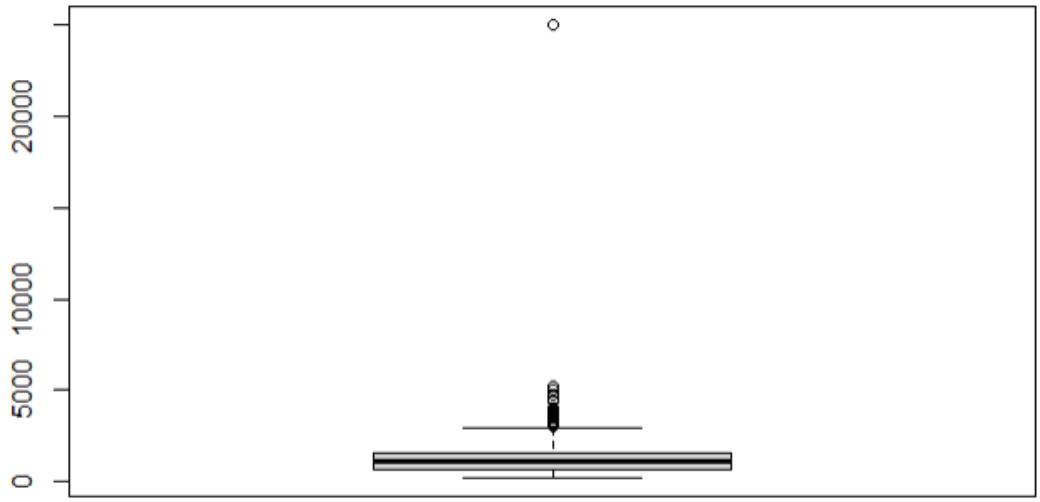


Ahora pasamos a ver el boxplot del Weekly Price. Decidimos basarnos en este porque contiene los datos reales del precio semanal establecidos por los dueños.:



Parece que los valores extremos no se alejan demasiado de la media, por lo que no realizaremos más acciones en este punto.

En cuanto al boxplot del Monthly Price:



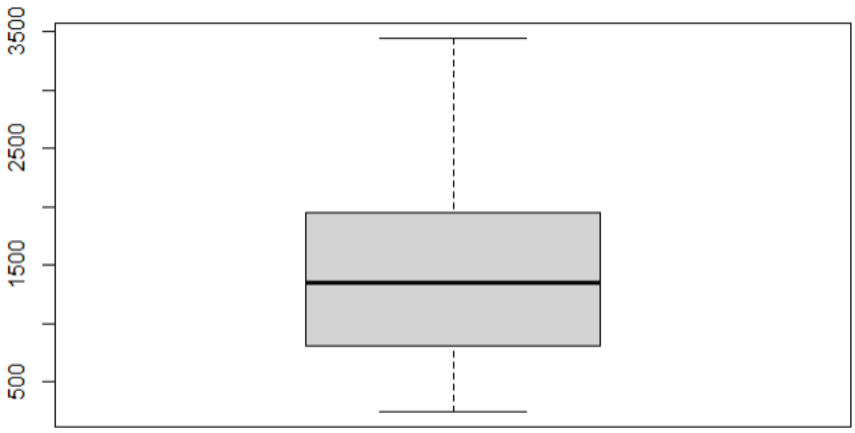
Si que tenemos al menos un valor que se aleja demasiado, por lo que procedemos a hacer una limpieza de ese outlier. Para poder realizar previsiones más acertadas.

Primero comprobamos los datos de esta columna:

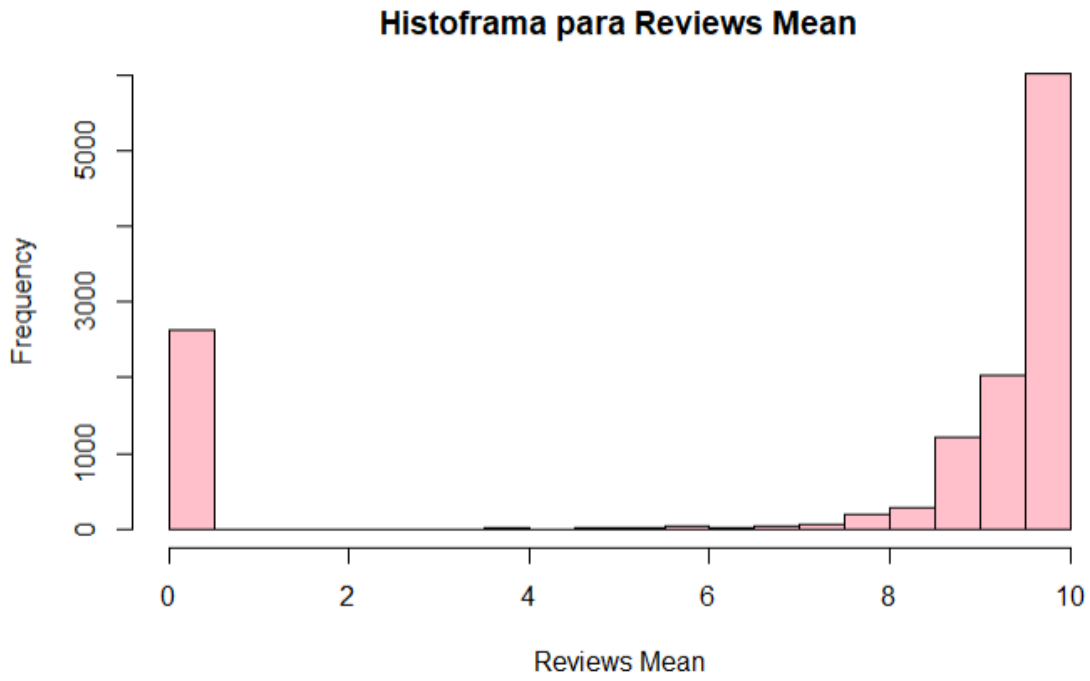
count	12539.000000
mean	1556.715448
std	932.834542
min	250.000000
25%	850.000000
50%	1350.000000
75%	2100.000000
max	25000.000000

Name: Monthly Price New, dtype: float64

Al ver que su distribución es similar a la observada en Price, procedemos a la limpieza. Decidimos hacer la limpieza sobre la columna nueva que hemos creado con los precios calculados para el mes, y los precios establecidos por los dueños, ya que no tiene valores NA y es la que usaremos finalmente para la visualización y predicción. Finalmente obtenemos este boxplot del Monthly Price New:



Por último, revisamos la distribución de la media de las reviews con un histograma.



Los datos se distribuyen normalmente, ya que el 0 representa las viviendas que no han recibido ninguna valoración.

Tras la limpieza de los datos, nuestro dataset tiene 12542 filas y 45 columnas.

Visualización de los datos en Power BI

1. **Origen de datos:**

El dashboard se alimenta del archivo excel:

airbnbmadrid_selected_excelfinal_v1.xlsx

2. **Filtros**

El objetivo de los filtros es que el usuario obtenga resultados precisos según sus necesidades.

a. Barrio:

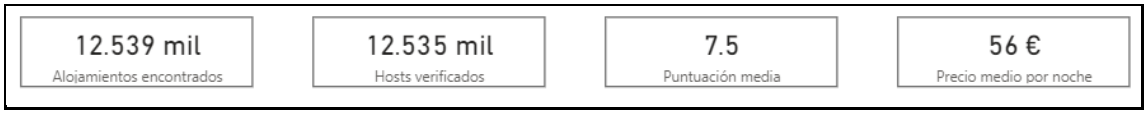
Columna: Neighbourhood Group Cleansed
Incluye todos los barrios de Madrid del Dataset.

b. Tipo de alojamiento:
Columna: Property Type
Tipo de propiedad/vivienda

c. Camas
Columna: Beds
Indica el número de camas disponibles en el alojamiento.

d. Precio noche
Columna: Price.
Incluye todos los precios por noche del dataset y permite filtrar por rango.

3. KPI's globales



a. Total alojamientos:
Medida DAX total_airbnb:

```
total_airbnb = CALCULATE(COUNT('Sheet1'[ID]))
```

b. Hosts verificados:
Campo: Host verification, filter by "Yes".

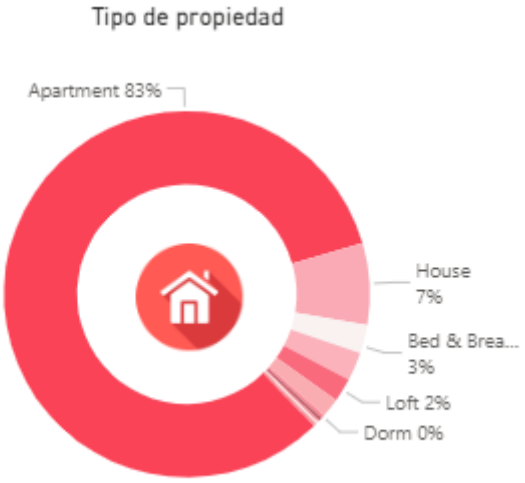
c. Puntuación media:
Campo: Reviews Mean (Promedio)

d. Precio medio por noche:
Campo: Price (Promedio)

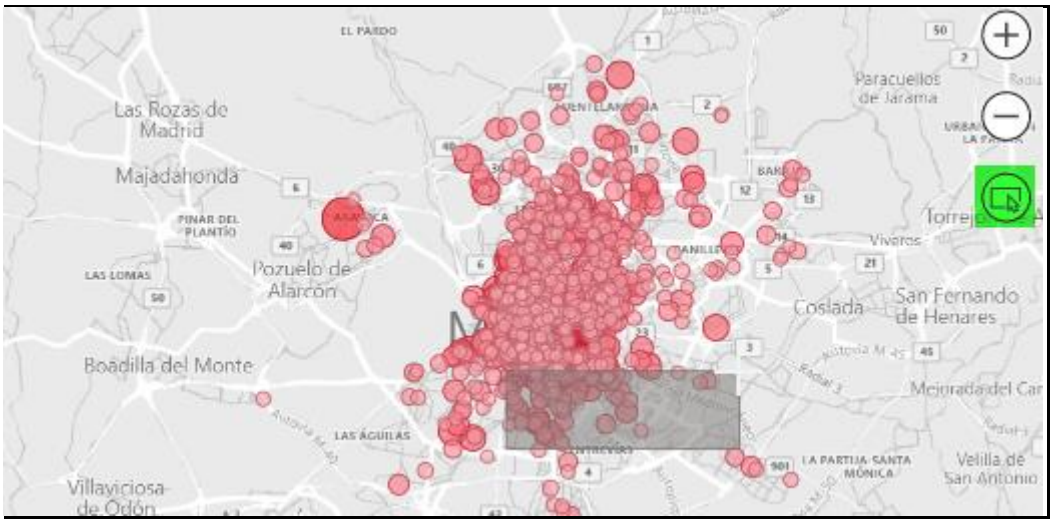
4. Visualizaciones

a. Top 10 barrios según promedio de puntuación

b. Tipos de propiedades por recuento ID



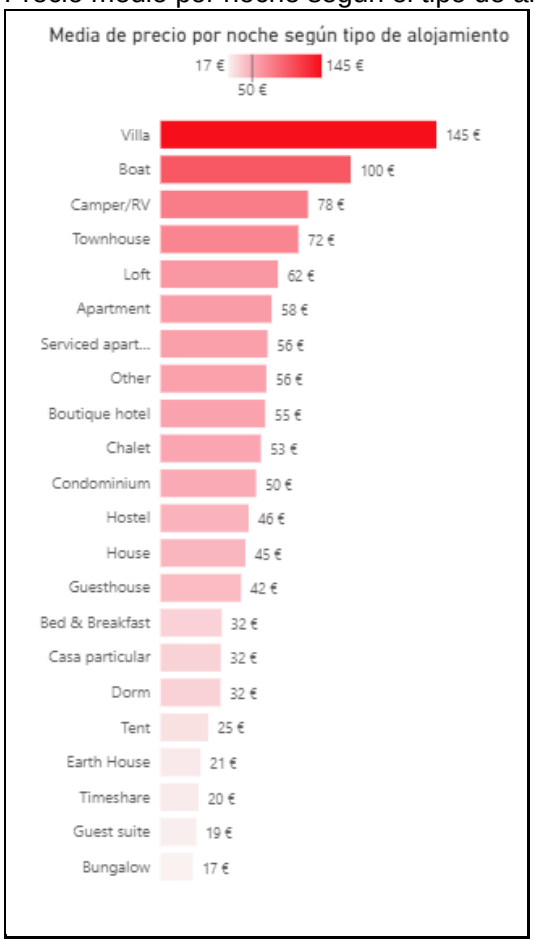
c. Mapa de Madrid basado en los campos: ID, Zipcode, Price, latitud y longitud. El tamaño de la burbuja indica el precio por noche, mientras más grande la burbuja, más elevado es el precio del alojamiento.
También ofrece la posibilidad de dibujar la zona de preferencia directamente en el mapa:



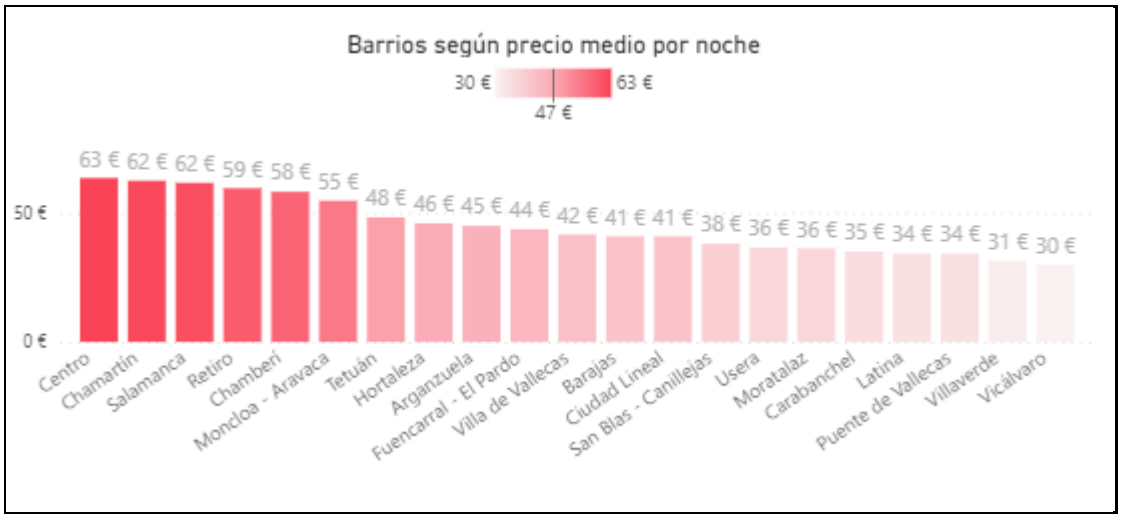
Ejemplo con filtros aplicados:

Barrio: Barajas
Precio maximo: 80€ la noche
Resultado:

d. Precio medio por noche según el tipo de alojamiento



e. Precio medio por noche según la ubicación (barrio)



d. Tabla con detalles de precio y referencia del Host

Precio por noche	Precio por una semana	Precio semanal final	Precio por un mes	Precio mensual final	Gastos de limpieza	Fianza	Host Name	Host URL
70 €	490 €	490 €	2,100 €	2,100 €	24 €	150 €	Violeta	https://www.airbnb.com/users/show/99999180
42 €	294 €	294 €	1,260 €	1,260 €	30 €	100 €	Elisa	https://www.airbnb.com/users/show/99998670
32 €	224 €	224 €	960 €	960 €	20 €	0 €	Amanda	https://www.airbnb.com/users/show/99994045
45 €	315 €	315 €	1,350 €	1,350 €	0 €	0 €	Amanda	https://www.airbnb.com/users/show/99994045
80 €	560 €	560 €	2,400 €	2,400 €	0 €	350 €	Arizonica Espacios Para Vivir	https://www.airbnb.com/users/show/99991901
100 €	700 €	700 €	3,000 €	3,000 €	0 €	350 €	Arizonica Espacios Para Vivir	https://www.airbnb.com/users/show/99991901

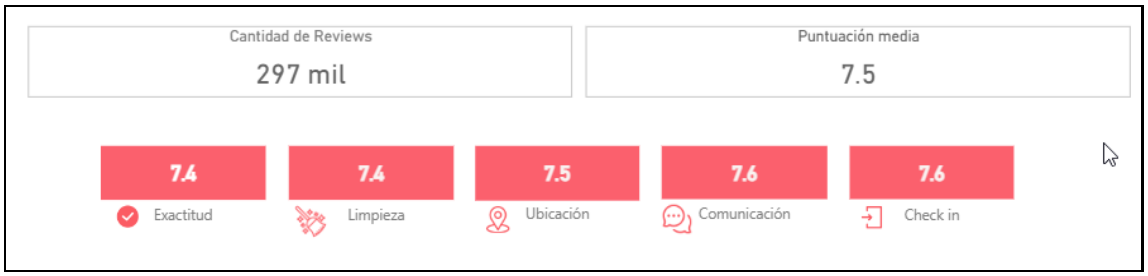
- Precio por noche
- Precio por una semana = Precio por noche * 7 días
- Precio semanal final = Precio con descuento aplicado (en caso de tener)
- Precio por un mes = Precio por noche * 30 días
- Precio mensual final = Precio con descuento aplicado (en caso de tener)
- Gastos de limpieza
- Fianza
- Referencia del host: Nombre y URL del anuncio en AirBnB

Ejemplo de caso con descuento por tiempo de alquiler

Precio por noche	Precio por una semana	Precio semanal final	Precio por un mes	Precio mensual final	Gastos de limpieza	Fianza	Host Name	Host URL
12 €	84 €	70 €	360 €	280 €	10 €	100 €	Miguel	https://www.airbnb.com/users/show/17371009

e. Reviews:

Tarjeta 1: Total Reviews
Tarjeta 2: Puntuación media
Tarjetas inferiores: Desglose de valoraciones



Modelo de Regresión Lineal Airbnb Madrid

Cargamos el data frame que utilizaremos para crear el modelo de regresión lineal. Estos datos ya han sido seleccionados y limpiados anteriormente.

Tenemos un total de 44 columnas en el data frame. El objetivo será crear el modelo de regresión lineal para predecir el precio de un inmueble en función de ciertas características. Para realizar el modelo tenemos que evaluar las características que aportarán valor a nuestro modelo, estas pueden ser: la ubicación "Neighbourhood Group Cleansed", tipo de habitación "Room Type", promedio de puntajes de reseñas "Reviews Mean", número de huéspedes "Accomodates", número de baños "Bathrooms" , número de habitaciones "Bedrooms" y el número de camas "Beds".

En el siguiente data frame podemos observar que la zona Centro es la que mayor frecuencia de registros tiene, lo utilizaremos más adelante para la aplicación del modelo.

Var1 <fctr>	Freq <int>
Centro	6377
Chamberí	885
Arganzuela	785
Salamanca	760
Tetuán	444
Retiro	413
Moncloa - Aravaca	402
Latina	378
Carabanchel	351
Chamartín	340
Ciudad Lineal	302
Puente de Vallecas	217
Hortaleza	173
Fuencarral - El Pardo	147
Usera	144
San Blas - Canillejas	116
Villaverde	80
Barajas	74
Moratalaz	74
Villa de Vallecas	44
Vicálvaro	33

21 rows

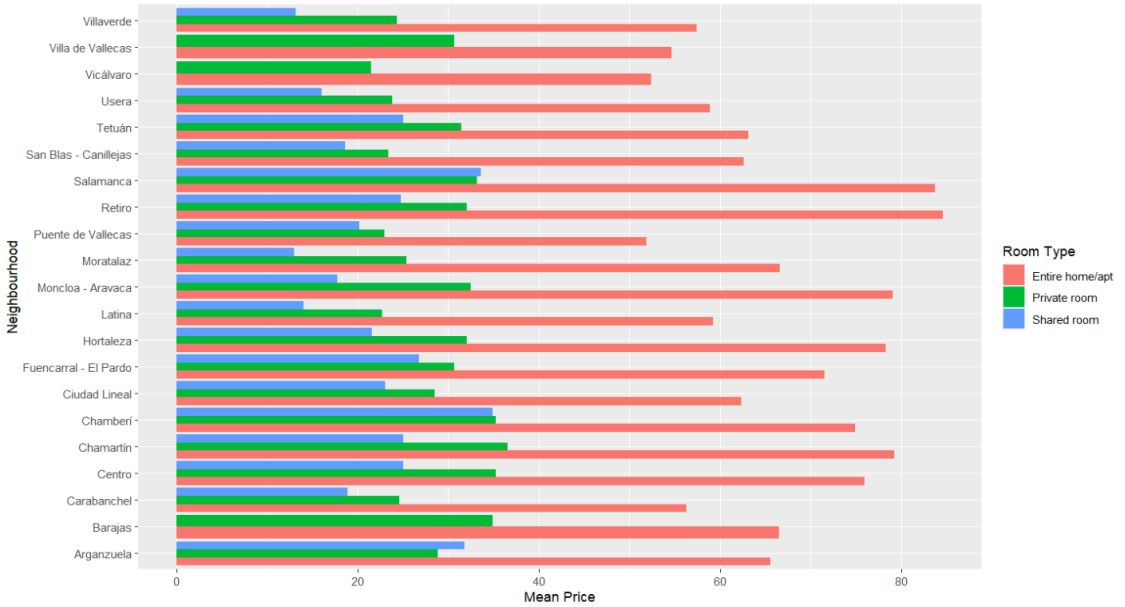
Agrupamos los precios de los inmuebles según su ubicación y tipo de inmueble.

A tibble: 60 × 3 Groups: Neighbourhood Group Cleansed [21]

Neighbourhood Group Cleansed <chr>	Room Type <chr>	mean_price <dbl>
Arganzuela	Entire home/apt	65.57895
Arganzuela	Private room	28.88631
Arganzuela	Shared room	31.75000
Barajas	Entire home/apt	66.57143
Barajas	Private room	34.88333
Carabanchel	Entire home/apt	56.27350
Carabanchel	Private room	24.59459
Carabanchel	Shared room	18.91667
Centro	Entire home/apt	75.93341
Centro	Private room	35.27394
Centro	Shared room	25.01493
Chamartín	Entire home/apt	79.28641
Chamartín	Private room	36.51515
Chamartín	Shared room	25.00000
Chamberí	Entire home/apt	74.91552
Chamberí	Private room	35.28804
Chamberí	Shared room	34.87500
Ciudad Lineal	Entire home/apt	62.33636
Ciudad Lineal	Private room	28.49474
Ciudad Lineal	Shared room	23.00000
Fuencarral - El Pardo	Entire home/apt	71.57447
Fuencarral - El Pardo	Private room	30.63158
Fuencarral - El Pardo	Shared room	26.80000
Hortaleza	Entire home/apt	78.28302

1-24 of 60 rows

En la siguiente gráfica se pueden observar los valores promedios de barrio según su tipo de habitación.



Evaluar variables a incluir con toda la data y observamos cuáles son las que aportan mayor valor y mejoran el coeficiente de determinación ajustado.

```
Call:
lm(formula = log(Price) ~ Accomodates + Bedrooms + Beds + `Room Type` +
    `Neighbourhood Group Cleansed` + `Reviews Mean` + `Cancellation Policy`,
    data = df_airbnb)

Residuals:
     Min       1Q   Median       3Q      Max
-1.86896 -0.22289 -0.01177  0.21931  1.81852

Coefficients:
                                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                                     3.842562   0.016717 229.863  < 2e-16 ***
Accomodates                                     0.075809   0.003444  22.011  < 2e-16 ***
Bedrooms                                        0.117509   0.005705  20.598  < 2e-16 ***
Beds                                           -0.034604   0.004119  -8.402  < 2e-16 ***
`Room Type`Private room                        -0.674654   0.007958 -84.776  < 2e-16 ***
`Room Type`Shared room                         -1.093774   0.026796 -40.819  < 2e-16 ***
`Neighbourhood Group Cleansed`Barajas           0.104502   0.041533   2.516 0.011878 *
`Neighbourhood Group Cleansed`Carabanchel       -0.207922   0.021947  -9.474  < 2e-16 ***
`Neighbourhood Group Cleansed`Centro            0.173077   0.013112  13.200  < 2e-16 ***
`Neighbourhood Group Cleansed`Chamartín         0.195331   0.022192   8.802  < 2e-16 ***
`Neighbourhood Group Cleansed`Chamberí          0.144105   0.016781   8.588  < 2e-16 ***
`Neighbourhood Group Cleansed`Ciudad Lineal     -0.052532   0.023122  -2.272 0.023107 *
`Neighbourhood Group Cleansed`Fuencarral - El Pardo  0.032713   0.030715   1.065 0.286875
`Neighbourhood Group Cleansed`Hortaleza         0.094471   0.028691   3.293 0.000995 ***
`Neighbourhood Group Cleansed`Latina            -0.235242   0.021399 -10.993  < 2e-16 ***
`Neighbourhood Group Cleansed`Moncloa - Aravaca  0.077999   0.021001   3.714 0.000205 ***
`Neighbourhood Group Cleansed`Moratalaz         -0.131404   0.041528  -3.164 0.001559 **
`Neighbourhood Group Cleansed`Puente de Vallecas  -0.235441   0.026189  -8.990  < 2e-16 ***
`Neighbourhood Group Cleansed`Retiro            0.150211   0.020762   7.235 4.93e-13 ***
`Neighbourhood Group Cleansed`Salamanca         0.181344   0.017419  10.411  < 2e-16 ***
`Neighbourhood Group Cleansed`San Blas - Canillejas -0.143656   0.033995  -4.226 2.40e-05 ***
`Neighbourhood Group Cleansed`Tetuán            0.017722   0.020284   0.874 0.382314
`Neighbourhood Group Cleansed`Usera             -0.184004   0.030946  -5.946 2.82e-09 ***
`Neighbourhood Group Cleansed`Vicálvaro         -0.276906   0.060683  -4.563 5.09e-06 ***
`Neighbourhood Group Cleansed`Villa de Vallecas  -0.053437   0.052867  -1.011 0.312148
`Neighbourhood Group Cleansed`Villaverde        -0.273116   0.040308  -6.776 1.29e-11 ***
`Reviews Mean`                                  -0.009525   0.000816 -11.673  < 2e-16 ***
`Cancellation Policy`moderate                   -0.030850   0.008028  -3.843 0.000122 ***
`Cancellation Policy`strict                     -0.020970   0.007745  -2.708 0.006787 **
`Cancellation Policy`super_strict_30            0.092149   0.241399   0.382 0.702670
`Cancellation Policy`super_strict_60            0.634502   0.171041   3.710 0.000208 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3411 on 12508 degrees of freedom
Multiple R-squared:  0.6616,    Adjusted R-squared:  0.6608
F-statistic: 815.2 on 30 and 12508 DF,  p-value: < 2.2e-16
```

Tabla de correlación

	Price	Accomodates	Bathrooms	Bedrooms	Beds	Reviews Mean
Price	1.0	0.6	0.1	0.4	0.4	0.1
Accomodates	0.6	1.0	0.2	0.6	0.8	0.1
Bathrooms	0.1	0.2	1.0	0.3	0.2	-0.1
Bedrooms	0.4	0.6	0.3	1.0	0.6	0.0
Beds	0.4	0.8	0.2	0.6	1.0	0.1
Reviews Mean	0.1	0.1	-0.1	0.0	0.1	1.0

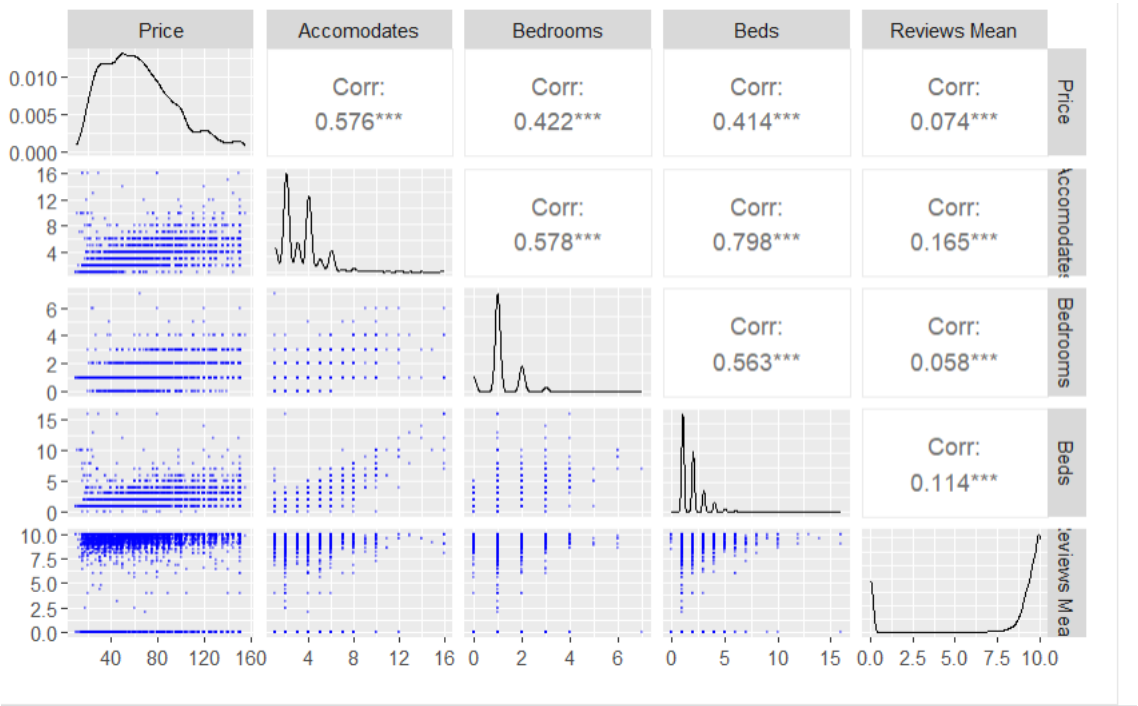
Una vez identificado las variables predictoras, seleccionaremos el barrio “Centro”, para ello creamos un subset.

ID <dbl>	Listing Url <chr>	Host ID...3 <dbl>	Neighbourhood Group Cleansed <chr>	City <chr>	State <chr>	Zipcode <chr>	Country Code <chr>
15141125	https://www.airbnb.com/rooms/15141125	96019257	Centro	Madrid	Comunidad de Madrid	28005	ES
9470166	https://www.airbnb.com/rooms/9470166	9885245	Centro	Madrid	Comunidad de Madrid	28012	ES
17444981	https://www.airbnb.com/rooms/17444981	118059488	Centro	Madrid	Comunidad de Madrid	28012	ES
3284565	https://www.airbnb.com/rooms/3284565	1892467	Centro	Madrid	Community of Madrid	28012	ES
499911	https://www.airbnb.com/rooms/499911	2467212	Centro	Madrid	Comunidad de Madrid	28012	ES
1346747	https://www.airbnb.com/rooms/1346747	7306349	Centro	Madrid	Community of Madrid	28005	ES
3097553	https://www.airbnb.com/rooms/3097553	15327748	Centro	Madrid	Community of Madrid	28012	ES
13440784	https://www.airbnb.com/rooms/13440784	76707968	Centro	Madrid	Comunidad de Madrid	28005	ES
7818234	https://www.airbnb.com/rooms/7818234	5239042	Centro	Madrid	Comunidad de Madrid	28012	ES
1386096	https://www.airbnb.com/rooms/1386096	6643556	Centro	Madrid	Community of Madrid	28012	ES
14180827	https://www.airbnb.com/rooms/14180827	18942409	Centro	Madrid	Comunidad de Madrid	28012	ES
8011473	https://www.airbnb.com/rooms/8011473	8831188	Centro	Madrid	Comunidad de Madrid	28012	ES
13221821	https://www.airbnb.com/rooms/13221821	74180884	Centro	Madrid	Comunidad de Madrid	*NA*	ES
1942585	https://www.airbnb.com/rooms/1942585	1528801	Centro	Madrid	Comunidad de Madrid	28012	ES
9460773	https://www.airbnb.com/rooms/9460773	15208964	Centro	Madrid	Comunidad de Madrid	28012	ES
16311700	https://www.airbnb.com/rooms/16311700	19725037	Centro	Madrid	Comunidad de Madrid	28012	ES
5399733	https://www.airbnb.com/rooms/5399733	3272228	Centro	Madrid	Comunidad de Madrid	28005	ES
14773153	https://www.airbnb.com/rooms/14773153	36963267	Centro	Madrid	Comunidad de Madrid	28005	ES
1666184	https://www.airbnb.com/rooms/1666184	8824421	Centro	Madrid	Community of Madrid	28005	ES
17986327	https://www.airbnb.com/rooms/17986327	15258781	Centro	Madrid	Comunidad de Madrid	28012	ES
7796518	https://www.airbnb.com/rooms/7796518	39840488	Centro	Madrid	*NA*	28013	ES
12809312	https://www.airbnb.com/rooms/12809312	2009482	Centro	Madrid	Comunidad de Madrid	28012	ES
4196358	https://www.airbnb.com/rooms/4196358	11488818	Centro	Madrid	Comunidad de Madrid	28013	ES
13183219	https://www.airbnb.com/rooms/13183219	73741764	Centro	Madrid	Comunidad de Madrid	28005	ES
15508358	https://www.airbnb.com/rooms/15508358	99632258	Centro	Madrid	Comunidad de Madrid	28013	ES
11050879	https://www.airbnb.com/rooms/11050879	57354901	Centro	Madrid	Comunidad de Madrid	28005	ES
5547854	https://www.airbnb.com/rooms/5547854	9193333	Centro	Madrid	Comunidad de Madrid	28012	ES

1-27 of 6,377 rows | 1-8 of 44 columns

Previous 1 2 3 4 5 6 ... 38 Next

Evaluamos nuevamente la tabla de correlación donde notamos que la variable Bathrooms no tiene correlación con la variable Price, por lo tanto, lo dejamos de considerar.



Separaremos los datos de train y test. Para ello, tomamos el 70% de los datos para utilizarlos entrenamiento del modelo y el resto para evaluar la calidad del modelo.

```{r}
set.seed(1234)
idx_1<- sample(1:nrow(df_centro), nrow(df_centro)*0.7)
train.airbnb<- df_centro[idx_1,]
test.airbnb<- df_centro[-idx_1,]
```

Evaluamos el modelo de regresión lineal con las variables a incluir de la data de entrenamiento.

```
Call:
lm(formula = Price ~ Accomodates + Bedrooms + Beds + `Room Type`,
    data = train.airbnb)

Residuals:
    Min      1Q  Median      3Q     Max
-86.182 -12.999  -2.917   9.445 114.415

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             48.1616     0.8777  54.875  < 2e-16 ***
Accomodates              4.6678     0.3484  13.397  < 2e-16 ***
Bedrooms                11.3484     0.5794  19.587  < 2e-16 ***
Beds                    -1.7274     0.4272  -4.044 5.35e-05 ***
`Room Type`Private room -31.5334     0.8021 -39.315  < 2e-16 ***
`Room Type`Shared room  -49.2710     2.8620 -17.216  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.53 on 4457 degrees of freedom
Multiple R-squared:  0.5435,    Adjusted R-squared:  0.543
F-statistic:  1061 on 5 and 4457 DF,  p-value: < 2.2e-16

            (Intercept)             Accomodates                Bedrooms                    Beds
              48.161588                4.667785               11.348429               -1.727405
`Room Type`Private room  `Room Type`Shared room
             -31.533418               -49.271012
```

Ejemplo precio alquiler en el Centro para 2 personas, 1 habitación, 2 camas, Tipo Private Room.

Luego de aplicar la fórmula de la regresión lineal obtenemos como resultado: "El precio de alquiler según las características mencionadas es de 37.94 EUR".

Lo cual se asemeja a la realidad si nos fijamos en la base de datos y filtramos con los mismos valores que se ha solicitado en el modelo.

Calculamos las figuras de calidad de training y testing con lo cual se puede observar que los valores se asemejan el uno del otro.

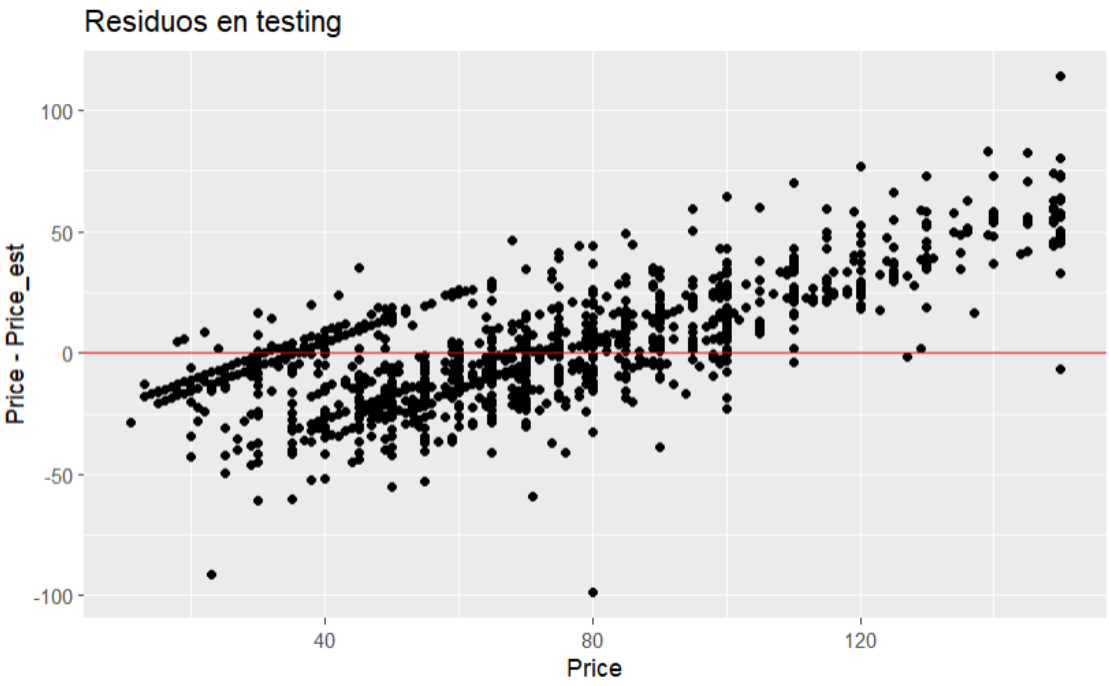
- Training:

```
     RMSE   Rsquared        MAE
20.5176342  0.5435058 15.4028822
```

- Test:

```
      RMSE   Rsquared        MAE
21.2706213  0.5051226 16.0392047
```

Luego de evaluar diferentes variables se ha mejorado el valor del coeficiente de determinación, aun así, no llega a tener un valor para que el modelo sea aceptable, pasamos a evaluar el residuo o error del modelo el cual se asemeja a la campana de Gauss.



¿Que se haría igual y que se haría diferente?

Tomaríamos esta misma base de datos para trabajar, ya que tiene mucha información que se puede extraer para trabajar y sacar nuevas conclusiones sobre posibles búsquedas.

Para ampliar el desarrollo de esta herramienta, haríamos nuevos cálculos basados en los precios de alojamiento de otras ciudades y países, para ampliar las posibilidades de búsqueda a lugares fuera de Madrid.

Conclusiones y lessons learned

A través de los datos, a día de hoy es posible analizar y predecir el comportamiento que un usuario tendrá en la red, conocer qué piensan los clientes y usuarios sobre una marca o un producto, y cuáles son sus necesidades reales sobre la adquisición de productos o servicios.

El desarrollo de este proyecto nos ha dejado como enseñanza, la gran importancia que tiene aprovechar los datos y utilizarlos para identificar nuevas oportunidades. Eso, a su vez, conduce a movimientos de negocios más inteligentes, operaciones más eficientes, mayores ganancias y al alcance de grandes objetivos para el desarrollo de herramientas que aporten algo que resulte útil para la sociedad.