INFORME PROYECTO FINAL

Predicción de Precios de Airbnb en Madrid

1. Introducción

Este informe explica nuestro proyecto sobre el análisis y la predicción de precios de Airbnb en Madrid, destacando cómo organizamos y analizamos los datos para entender y prever cambios en el mercado de alquiler temporal.

Comenzamos definiendo un modelo de entidad-relación para estructurar los datos de manera eficiente, lo que facilita el análisis y la implementación futura de modelos predictivos. A continuación, realizamos un análisis exploratorio para detectar patrones y tendencias que afectan los precios, lo cual es esencial para comprender el mercado.

Luego revisamos la calidad de los datos y ajustamos el tratamiento de estos para asegurar su precisión en el modelado. También incorporamos gráficos que resaltan las métricas clave del análisis, facilitando la interpretación de nuestros hallazgos.

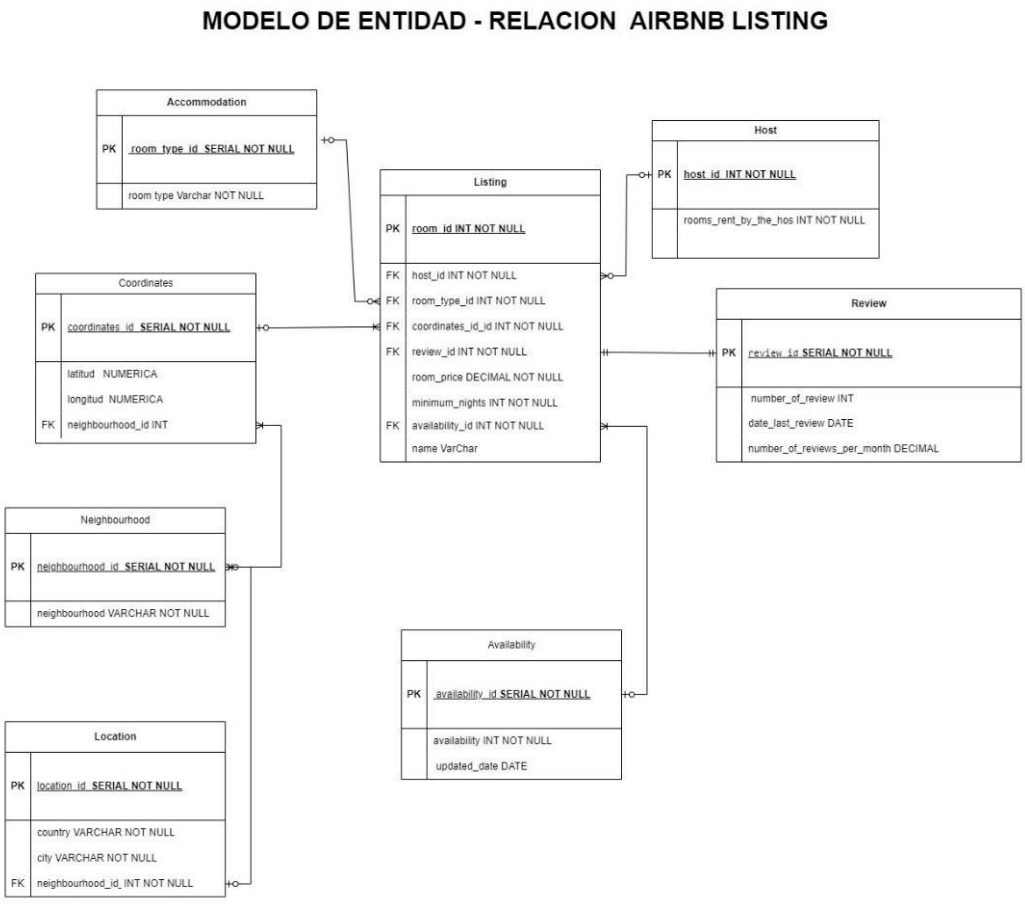
Finalmente, describimos el preprocesamiento y modelado de los datos, detallando las técnicas y modelos que utilizamos para predecir los precios de los listados y evaluando su efectividad y precisión. Este proceso destaca la importancia de seleccionar y transformar adecuadamente las variables y refleja los retos y limitaciones que encontramos.

Con este informe, esperamos proporcionar una visión clara y profunda de los factores que influyen en los precios de Airbnb en Madrid y ofrecer recomendaciones que mejoren la toma de decisiones de los anfitriones en este mercado dinámico.

2. Arquitectura y Validación de los Datos

2.1 Modelo de Entidad-Relación

Para el desarrollo de un modelo de predicción de precios en los listados de Airbnb, hemos diseñado un modelo de entidad-relación (ER) que estructura y organiza los datos de manera eficiente. Este modelo es fundamental para la gestión y análisis de datos, facilitando la realización de predicciones precisas.



Modelo de Entidad – Relación Airbnb Listing

Componentes Principales del Modelo ER:

- **Entidades Clave:**

    - **Accommodation:** Define los tipos de alojamiento disponibles. Incluye campos como **room_type_id** para un identificador único y **room_type** que describe el tipo de

habitación. Esta entidad es crucial para categorizar los diferentes tipos de habitaciones ofrecidas en los listados.

- **Listing:** Es la entidad central del modelo, donde cada registro representa un único listado de Airbnb. Incluye detalles como **room_id, host_id, room_type_id, coordinates_id, review_id, availability_id, room_price** (precio por noche), **minimum_nights** (mínimo de noches requeridas) y **name** (nombre del listado). Este conjunto de atributos facilita la gestión detallada de cada listado.
- **Host:** Representa a los anfitriones de los alojamientos. Incluye **host_id** como identificador único y **rooms_rent_by_the_host**, que indica el número de habitaciones que el anfitrión ofrece en alquiler. Esta entidad ayuda a rastrear la actividad de los anfitriones en la plataforma.
- **Review :** Contiene información sobre las reseñas de los listados, crucial para evaluar la popularidad y la satisfacción del cliente. Atributos como **review_id, number_of_reviews, date_last_review, y number_of_reviews_per_month** permiten analizar el rendimiento de cada listado a través del feedback de los usuarios.
- **Availability:** Gestiona la disponibilidad de los listados con atributos como **availability_id, availability** que indica el estado de disponibilidad actual, y **updated_date** que marca la última actualización de esta información.
- **Coordinates:** Almacena las coordenadas geográficas de cada listado, incluyendo **coordinates_id, latitude y longitude**. También está vinculada a **neighbourhood_id,** que relaciona las coordenadas con un barrio específico.
- **Neighbourhood:** Esta entidad captura los detalles del barrio con **neighbourhood_id y neighbourhood**, facilitando análisis geográficos detallados y la agrupación de listados por áreas.
- **Location:** Captura información más amplia de ubicación con **location_id, country, city, y neighbourhood_id**. Permite una clasificación detallada de los listados según su localización geográfica y facilita segmentaciones por áreas urbanas.

- **Relaciones Principales:**

  - **Accommodation a Listing:** Un tipo de habitación puede estar asociado con múltiples listados.
  - **Host a Listing:** Un anfitrión puede tener varios listados.
  - **Coordinates a Listing:** Cada listado está definido por un conjunto único de coordenadas.
  - **Review a Listing:** Cada listado está vinculado a un conjunto específico de reseñas.
  - **Availability a Listing:** Cada listado detalla una disponibilidad específica.
  - **Neighbourhood a Coordinates:** Un vecindario puede englobar múltiples coordenadas.
  - **Neighbourhood a Location:** Muchos barrios pueden formar parte de la misma localidad.

Este modelo ER proporciona una estructura clara y optimizada para la gestión de los datos de Airbnb, esencial para la realización de análisis detallados y el desarrollo de modelos predictivos eficaces.

3. Análisis Exploratorio

En esta sección, vamos a sumergirnos en un análisis exploratorio del dataset de Airbnb centrado en Madrid. Este análisis nos permitirá descubrir tendencias, comportamientos y patrones en los listados de Airbnb, proporcionándonos datos valiosos sobre cómo se distribuyen las propiedades, qué características son más comunes, y cómo varían los precios en diferentes distritos de la ciudad. Esto nos ayuda a comprender mejor el mercado de alquileres temporales en Madrid y a identificar oportunidades y áreas de mejora para los anfitriones de Airbnb.

3.1. Herramientas y Técnicas Usadas:

R y sus Librerías: Para el análisis utilizamos R, apoyándonos en librerías como **tidyverse** para la manipulación de datos, **ggplot2** para visualizaciones detalladas, **corrplot** para visualizar matrices de correlación, **leaflet** para mapas interactivos y otras. Estas herramientas fueron esenciales para manejar y analizar el dataset, asegurando resultados precisos y visualmente claros.

3.2. Revisión de la Calidad de los Datos y Tratamiento de Datos:

En esta sección del proyecto, nos centramos en asegurar que los datos de Airbnb que estamos analizando sean precisos y confiables.
Para mantener la integridad del conjunto de datos original mientras realizábamos modificaciones, inicialmente creamos una copia denominada **airbnb_data** y trabajamos sobre ella. Tras ello realizamos:

3.2.1. Conversión de variables

Convertimos las columnas **Date.last.review** y **Updated.Date** de texto a formato de fecha, lo cual nos permitió realizar cálculos de tiempo.

```
# Convertir la columna Date.last.review a tipo fecha
airbnb_data$Date.last.review <- as.Date(airbnb_data$Date.last.review, format = "%Y-%m-%d")

# Convertir la columna Updated.Date a tipo fecha
airbnb_data$Updated.Date <- as.Date(airbnb_data$Updated.Date, format = "%Y-%m-%d")
```

Asimismo, transformamos las columnas **Room.type**, **Neighbourhood**, **City**, y **Country** en formatos categóricos para facilitar el análisis por segmentos.

```
# Convertir columnas a factores (categóricas)
airbnb_data$Room.type <- as.factor(airbnb_data$Room.type)
airbnb_data$Neighbourhood <- as.factor(airbnb_data$Neighbourhood)
airbnb_data$City <- as.factor(airbnb_data$City)
airbnb_data$Country <- as.factor(airbnb_data$Country)
```

Ajustamos el conjunto de datos para incluir únicamente entradas correspondientes a Madrid, refinando y depurando la información sobre **Neighbourhood**,**City, Country.**

```
#Verificamos que solo queden los niveles deseados en City y Country
summary(airbnb_data[, c("City", "Country")])
```

```
##      City         Country
##  Madrid:21255   Spain:21255
```

Dividimos la columna de **Coordinates** en dos: **Latitude** y **Longitude**.

```
# Asegurándonos de que la columna Coordinates está en formato de string
airbnb_data$Coordinates <- as.character(airbnb_data$Coordinates)

# Separar la columna Coordinates en dos nuevas columnas: Latitude y Longitude
airbnb_data<- airbnb_data |>
  separate(Coordinates, into = c("Latitude", "Longitude"), sep = ",", convert = TRUE)
```

Nos aseguramos de que el cambio se ha hecho efectivo:

```
str(airbnb_data[, c("Latitude", "Longitude")])
```

```
## 'data.frame':    21255 obs. of  2 variables:
##  $ Latitude : num  40.4 40.4 40.4 40.4 40.4 ...
##  $ Longitude: num  -3.72 -3.69 -3.7 -3.71 -3.65 ...
```

Por último, determinamos que la columna **Location** era redundante ya que duplicaba información de otras columnas (**Country, City** y **Neighbourhood)**, y procedimos a eliminarla para simplificar la estructura de datos.

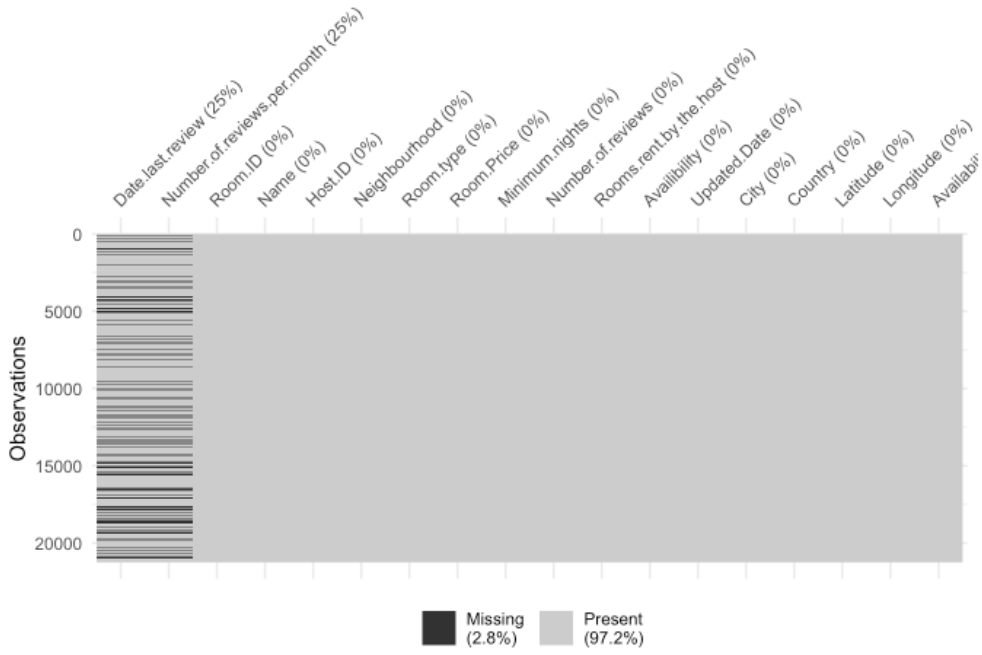
3.2.2. Valores Nulos y Ceros

Se investigó la presencia de valores nulos y ceros en el dataset, creando un dataframe específico para cuantificar estos valores en distintas columnas.

```
##                            NA_Count Zero_Count
## Room.Price                        0          1
## Number.of.reviews                 0       5400
## Date.last.review               5400          0
## Number.of.reviews.per.month    5400          0
## Availibility                      0       5465
```

Recuento de valores nulos y ceros en variables clave

Se detectaron cantidades significativas de ceros en **Number.of.reviews** y valores nulos en **Date.last.review** y **Number.of.reviews.per.month**, indicando listados sin reseñas recientes.



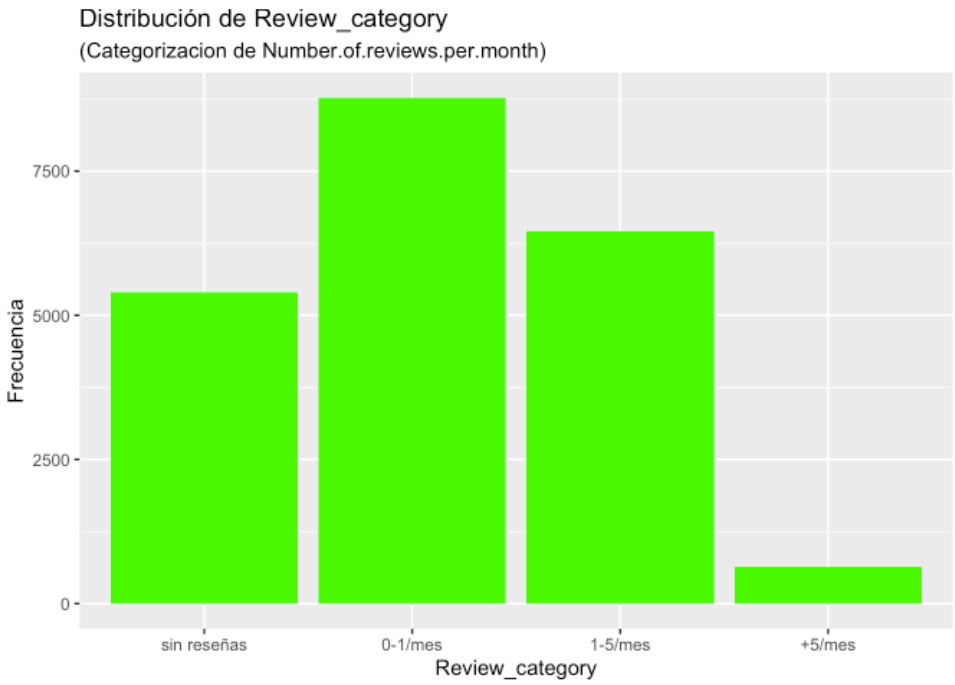
Distribución de datos faltantes y presentes en el dataset

Un registro en **Room.Price** con valor cero se eliminó por ser probablemente un error, y los **nulos** en **Number.of.reviews.per.month** se sustituyeron por ceros para simplificar el análisis.

3.2.3. Categorización de variables

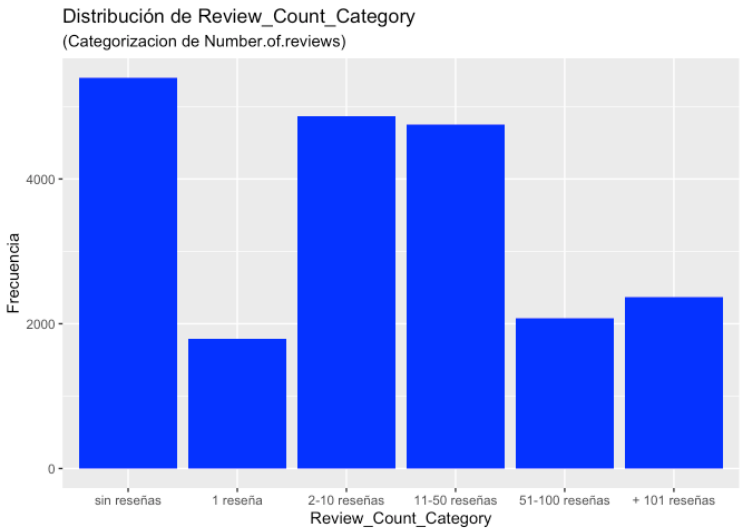
Se categorizaron algunas variables para facilitar el análisis y la visualización de los datos por segmentos, además de permitir una interpretación más intuitiva de la información.

1. **Number.of.reviews.per.month**: Esta variable fue categorizada (en la columna **Review_category**) para diferenciar entre listados según la frecuencia de sus reseñas mensuales. Se dividió en categorías como "sin reseñas", "0-1/mes", "1-5/mes", y "+5/mes". Esta categorización ayuda a identificar rápidamente la popularidad y la frecuencia de interacción de los usuarios con los listados.



Frecuencia de listados de Airbnb por categoría de reseñas mensuales

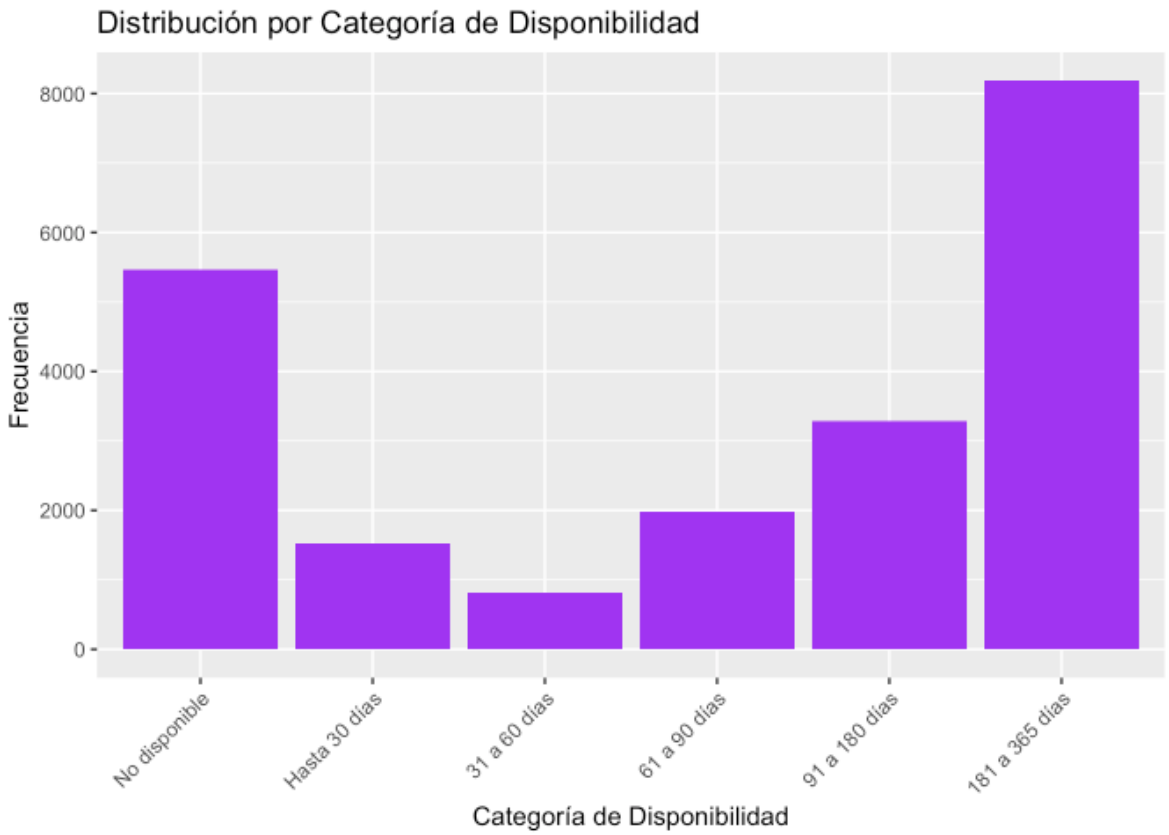
2. **Number.of.reviews**: Similar a la anterior, esta variable fue categorizada (en la columna Review_Count_Category) para reflejar la cantidad total de reseñas que ha recibido cada listado, con categorías que incluyen "sin reseñas", "1 reseña", "2-10 reseñas", "11-50 reseñas", "51-100 reseñas", y "+101 reseñas". Estas categorías permiten agrupar los listados en diferentes niveles de popularidad y experiencia de usuario.



Distribución de listados de Airbnb por total de reseñas recibidas.

3. **Availibility**: Se transformó en una variable categórica (en la columna **Availability_Cat**) para representar la disponibilidad de los listados en rangos definidos, como "No disponible",

"Hasta 30 días", "31 a 60 días", "61 a 90 días", "91 a 180 días", y "181 a 365 días". Esta clasificación es crucial para entender la disponibilidad de los listados a lo largo del año y planificar estrategias de gestión de inventario basadas en la demanda esperada.



Frecuencia de listados según su disponibilidad anual

3.2.4. Gestión de duplicados:

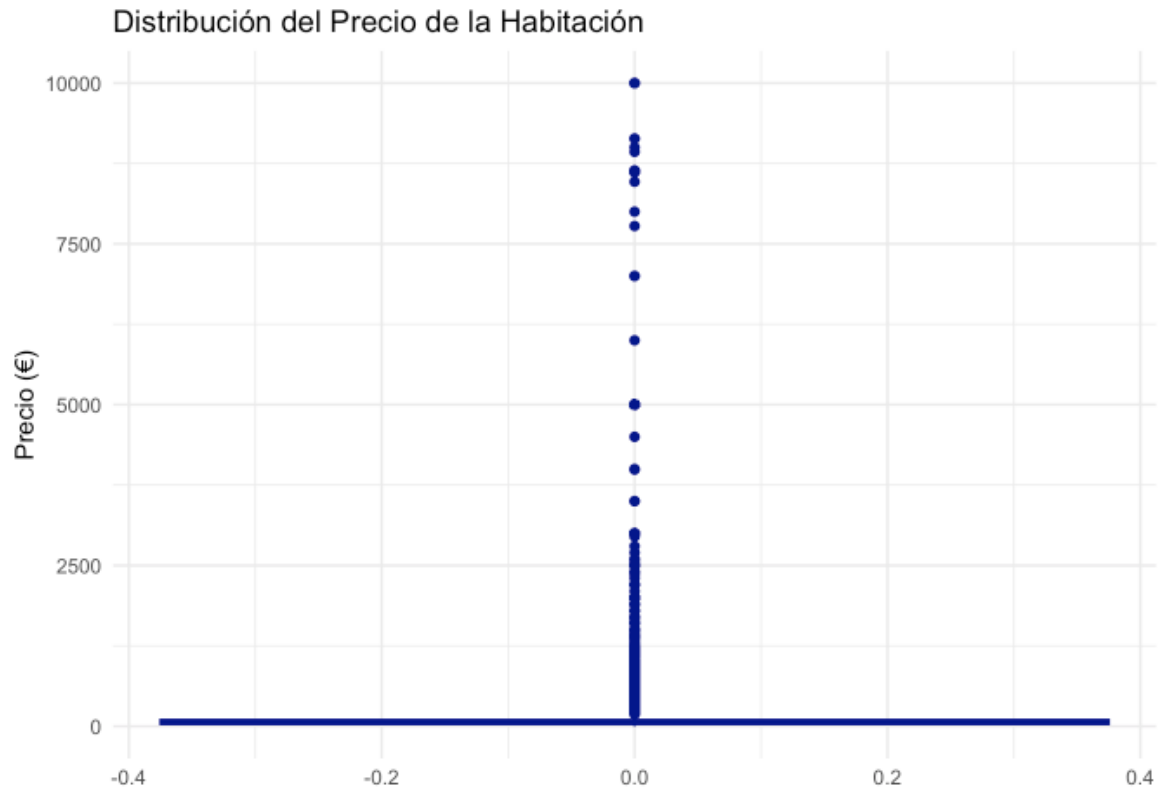
Se confirmó que no había duplicados en el conjunto de datos

```
filas_duplicadas <- airbnb_data[duplicated(airbnb_data), ]
if (nrow(filas_duplicadas) == 0) {
  print("No hay filas duplicadas en el dataset.")
} else {
  print("Se han encontrado filas duplicadas en el dataset.")
}
```

```
## [1] "No hay filas duplicadas en el dataset."
```

3.3. Análisis Descriptivo

Nos enfocamos en detectar y entender los outliers de algunas variables clave usando gráficos de caja. Por ejemplo, el precio de las habitaciones ("Room.Price") mostró una distribución con varios valores extremadamente altos, lo que probablemente representa alojamientos de lujo o ubicaciones premium. Lo mismo hicimos para el número mínimo de noches ("Minimum.nights") y la cantidad de habitaciones que cada anfitrión ofrece ("Rooms.rent.by.the.host"), revelando diferencias en cómo los anfitriones manejan sus políticas de hospedaje y el tamaño de sus operaciones.



Distribución de Room.Price

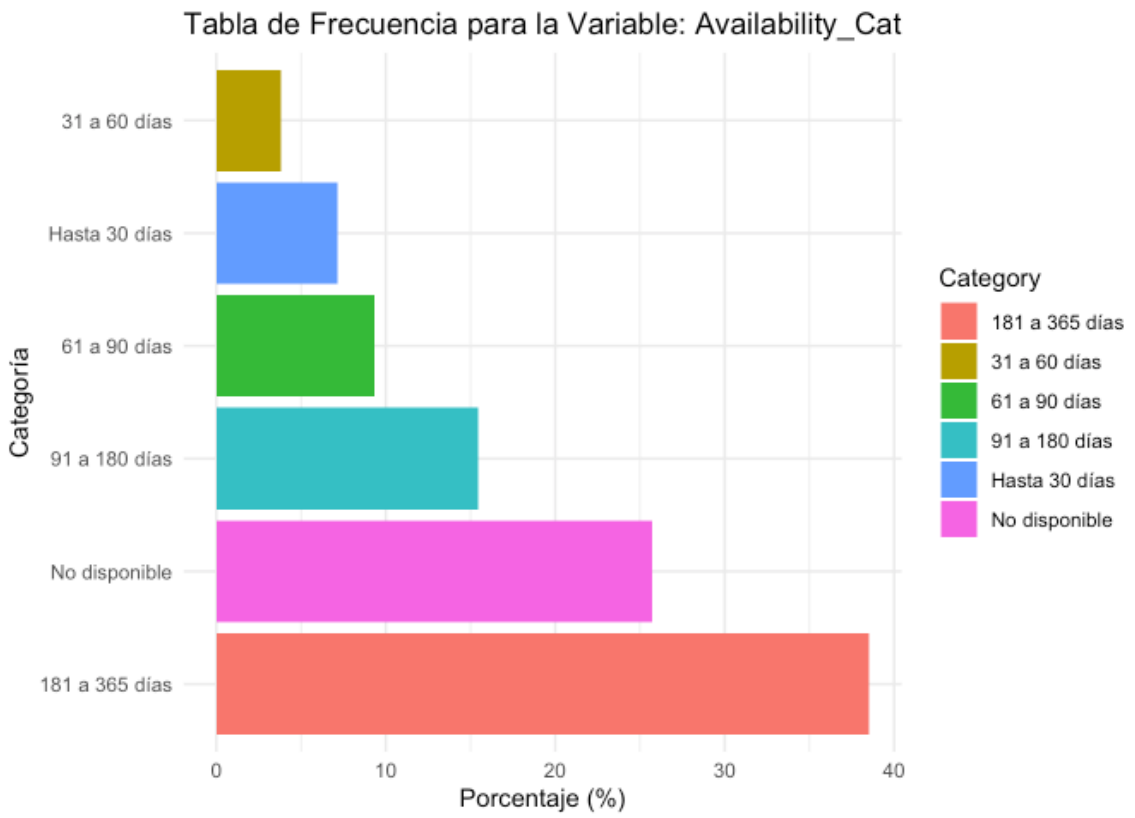
Utilizamos **dplyr** para filtrar y seleccionar las columnas numéricas que queríamos analizar más a fondo. Esto incluyó calcular los cuartiles para determinar los rangos de los outliers y así identificar el porcentaje de datos atípicos en cada variable numérica.

```
## # A tibble: 7 × 2
##   Variable                              Percentage_Outliers
##   <chr>                                 <chr>
## 1 outliers_in_Room.ID                   0.00%
## 2 outliers_in_Host.ID                   0.00%
## 3 outliers_in_Room.Price                11.40%
## 4 outliers_in_Minimum.nights            11.39%
## 5 outliers_in_Number.of.reviews         11.77%
## 6 outliers_in_Number.of.reviews.per.month 6.72%
## 7 outliers_in_Rooms.rent.by.the.host    15.43%
```

Porcentaje de valores atípicos por variable

Los resultados mostraron que tanto **Room.Price**, **Rooms.rent.by.the.host** como **Number.of.reviews** tenían una cantidad significativa de outliers, lo que indica una gran variabilidad en los precios y en la gestión de múltiples propiedades.

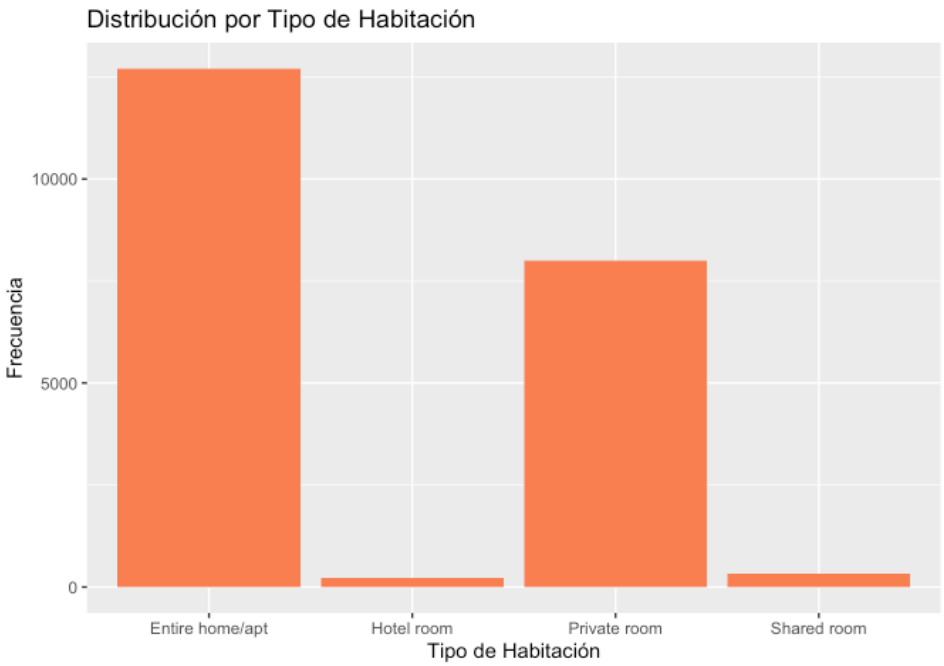
Para las variables categóricas, exploramos cómo se distribuyen las categorías dentro de cada variable.



Porcentaje de listados por rango de disponibilidad

También usamos **ggplot2** para hacer gráficos de otras variables categóricas, mostrando la distribución de los tipos de habitación y las categorías de disponibilidad.

Estos gráficos no solo hicieron los datos más fáciles de digerir visualmente, sino que también destacaron tendencias clave, como la popularidad de apartamentos completos en comparación con habitaciones compartidas o de hotel, y la variabilidad en la disponibilidad a lo largo del año.

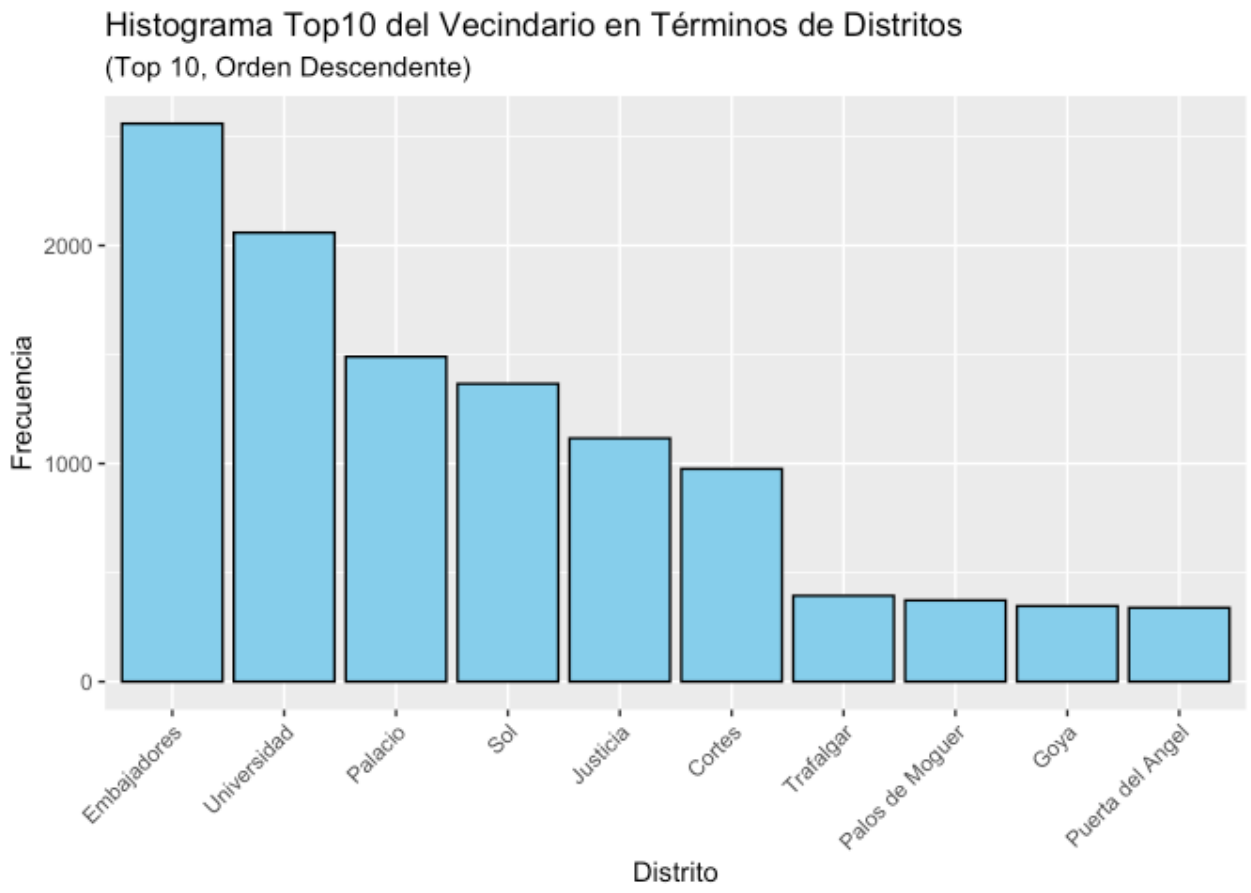


Frecuencia por tipos de alojamiento

3.4. EDA: Análisis Exploratorio de Datos

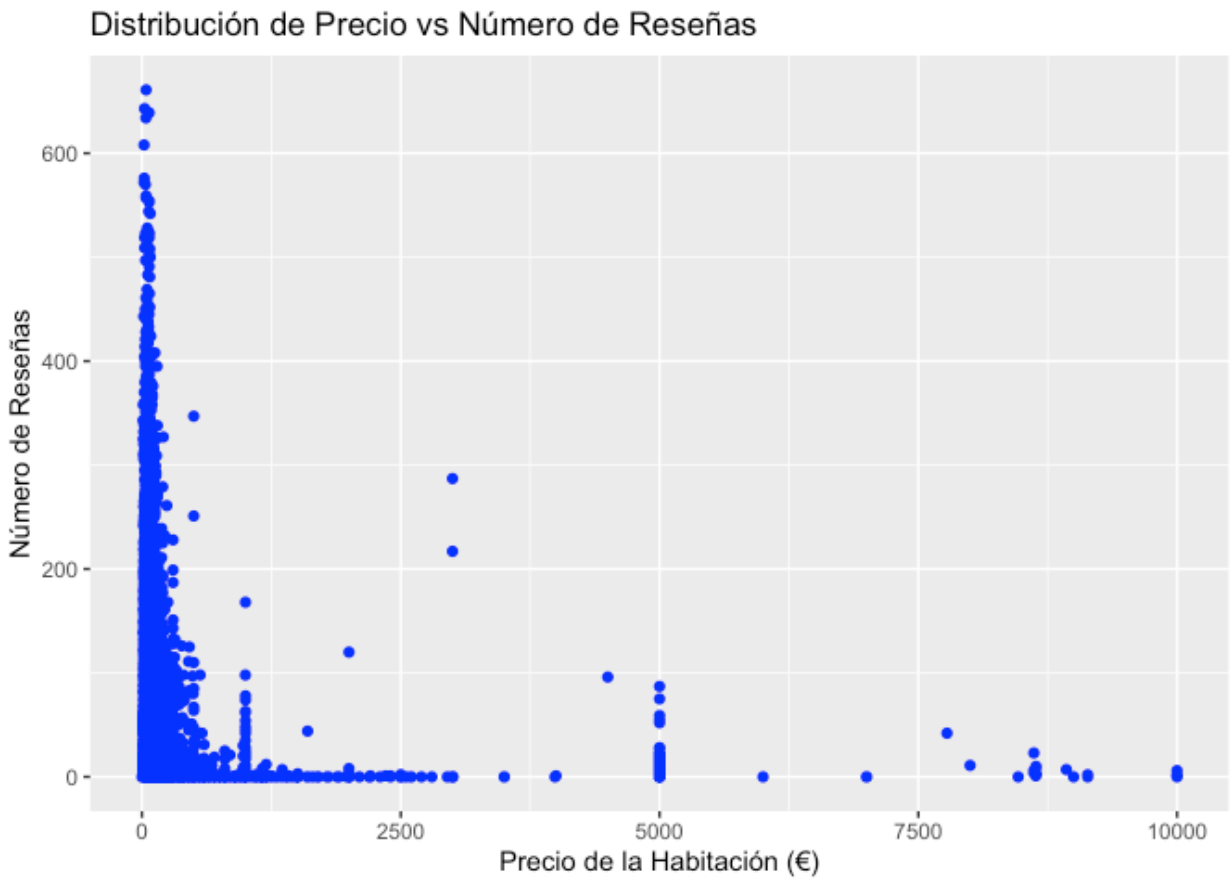
En nuestro análisis exploratorio de datos sobre los listados de Airbnb en Madrid, decidimos enfocarnos en cómo se reparten estos listados por los distintos distritos de la ciudad. Queríamos ver si podíamos descubrir algunos patrones en cuanto a dónde prefieren quedarse los usuarios y cómo eso podría influir en la popularidad de ciertas áreas. Esta información nos ayuda a entender qué zonas son las más buscadas y cómo eso podría impactar en la fijación de precios de Airbnb.

Primero, se calculó la frecuencia de cada distrito y se identificaron los diez distritos más frecuentes. Este histograma mostró claramente que Embajadores es el distrito con más listados, seguido de Universidad y Palacio, indicando áreas de alta demanda o mayor oferta de alojamientos.



Frecuencia de listados por distrito

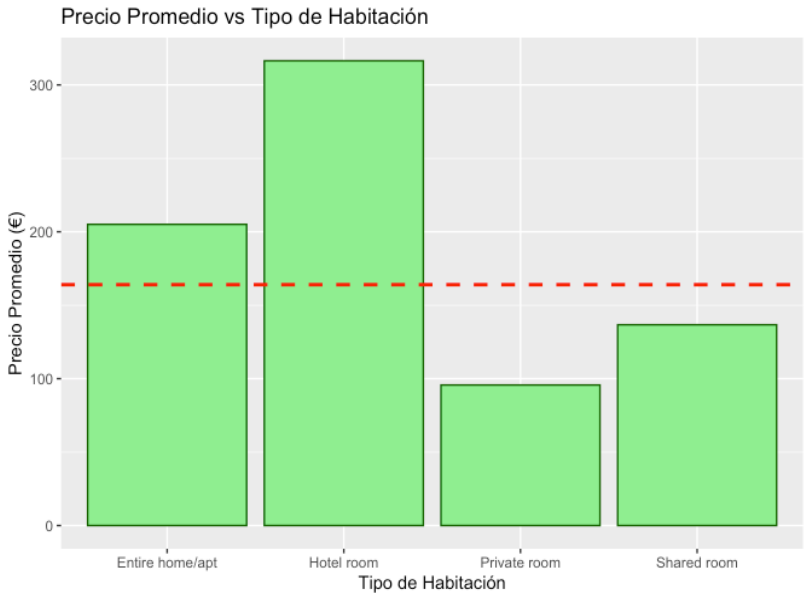
Adicionalmente, se exploró la relación entre el precio de las habitaciones y el número de reseñas recibidas, utilizando un gráfico de dispersión.



Precio versus número de reseñas

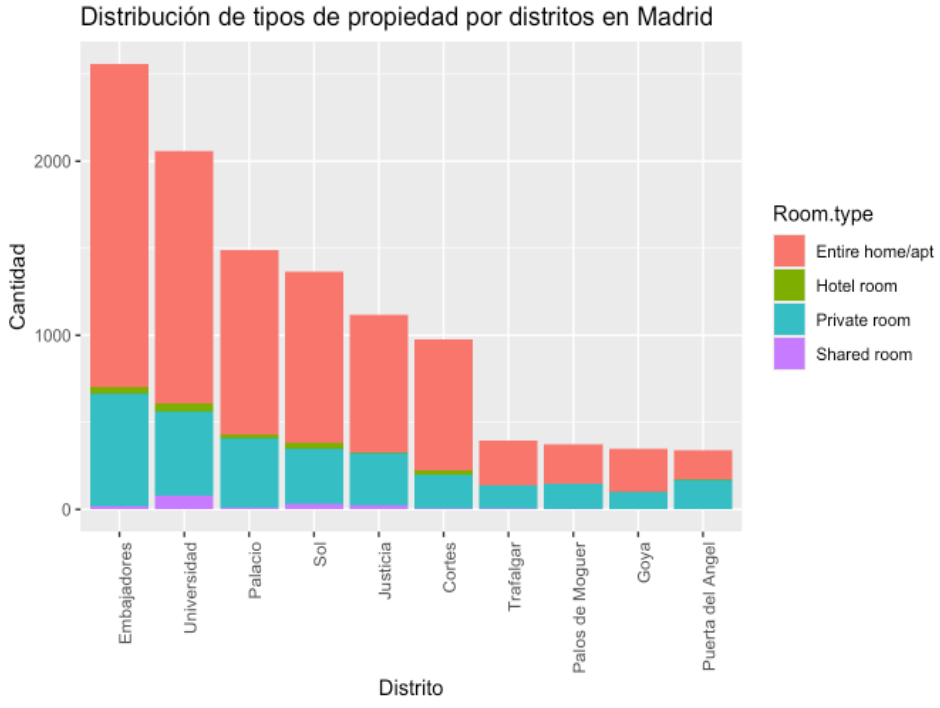
Los resultados indicaron que no hay una correlación directa entre precios altos y un mayor número de reseñas, sugiriendo que factores como la ubicación y la calidad del alojamiento pueden ser más determinantes en la popularidad de los listados.

Se analizó también cómo varían los precios promedio según el tipo de habitación, encontrando que las habitaciones de hotel tienen los precios promedio más altos, seguidos por las casas o apartamentos completos, las privadas y las compartidas, en ese orden.



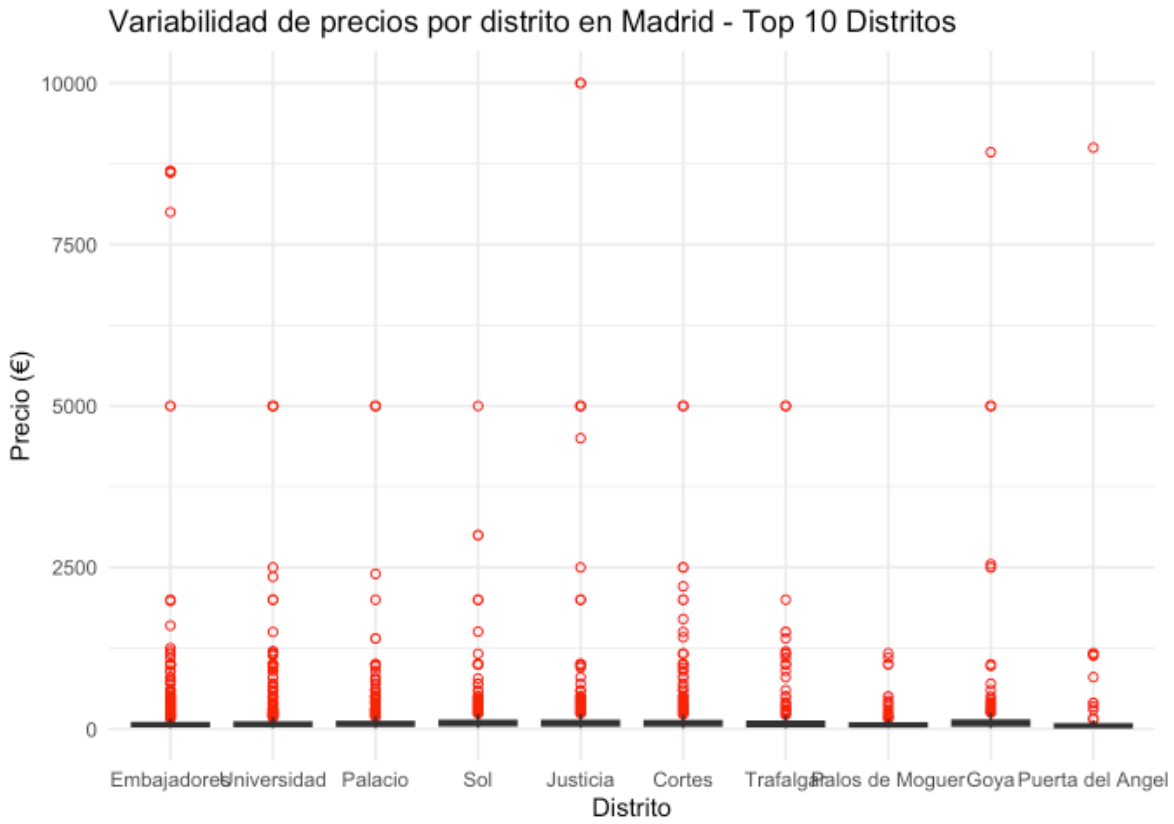
Comparación de precios promedio por tipo de habitación

Finalmente, se realizó una comparación de los tipos de propiedad y su distribución en los principales distritos de Madrid, mostrando una gran diversidad de alojamientos en distritos como Embajadores y Universidad.



Diversidad de tipos de habitación por distrito

También se observó cómo varían los precios en estos distritos, notando una considerable variabilidad y la presencia de valores atípicos en lugares como Embajadores, Universidad y Palacio.



Fluctuación de precios en los principales distritos

Este análisis detallado proporciona una visión clara de las tendencias del mercado de Airbnb en Madrid, útil para ajuste de precios y decisiones de inversión en propiedades.

4. Visualización de las métricas

Ahora procederemos a realizar la visualización de los KPIs que obtuvimos durante nuestro análisis exploratorio.

KPIs:

- Barrios más populares en Airbnb
    - Tipos de habitación por distrito
- Correlación entre precio y número de reseñas
- Precio promedio por tipo de habitación
- Variabilidad de precios por barrio
- Distribución geográfica barrios

4.1 Importar los Datos

Para empezar nos conectaremos a un archivo de texto, en este caso trabajaremos con el dataset que creamos tras nuestro análisis exploratorio, **processed-air-bnb-listings.csv**.

4.2 Verificar los Datos

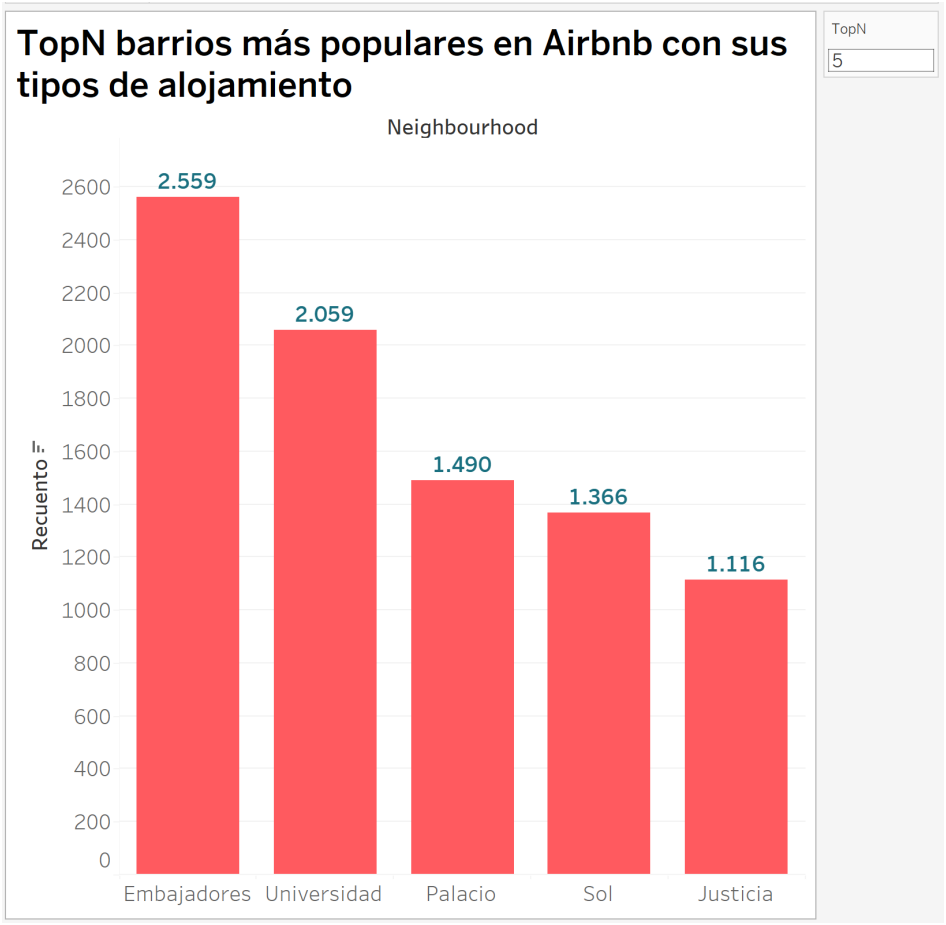
Comprobamos que todos los datos se carguen correctamente y que las variables correspondan con su tipo. Obtenemos que tenemos 20 campos y 21.255 filas.

4.3 Vizualizaciones

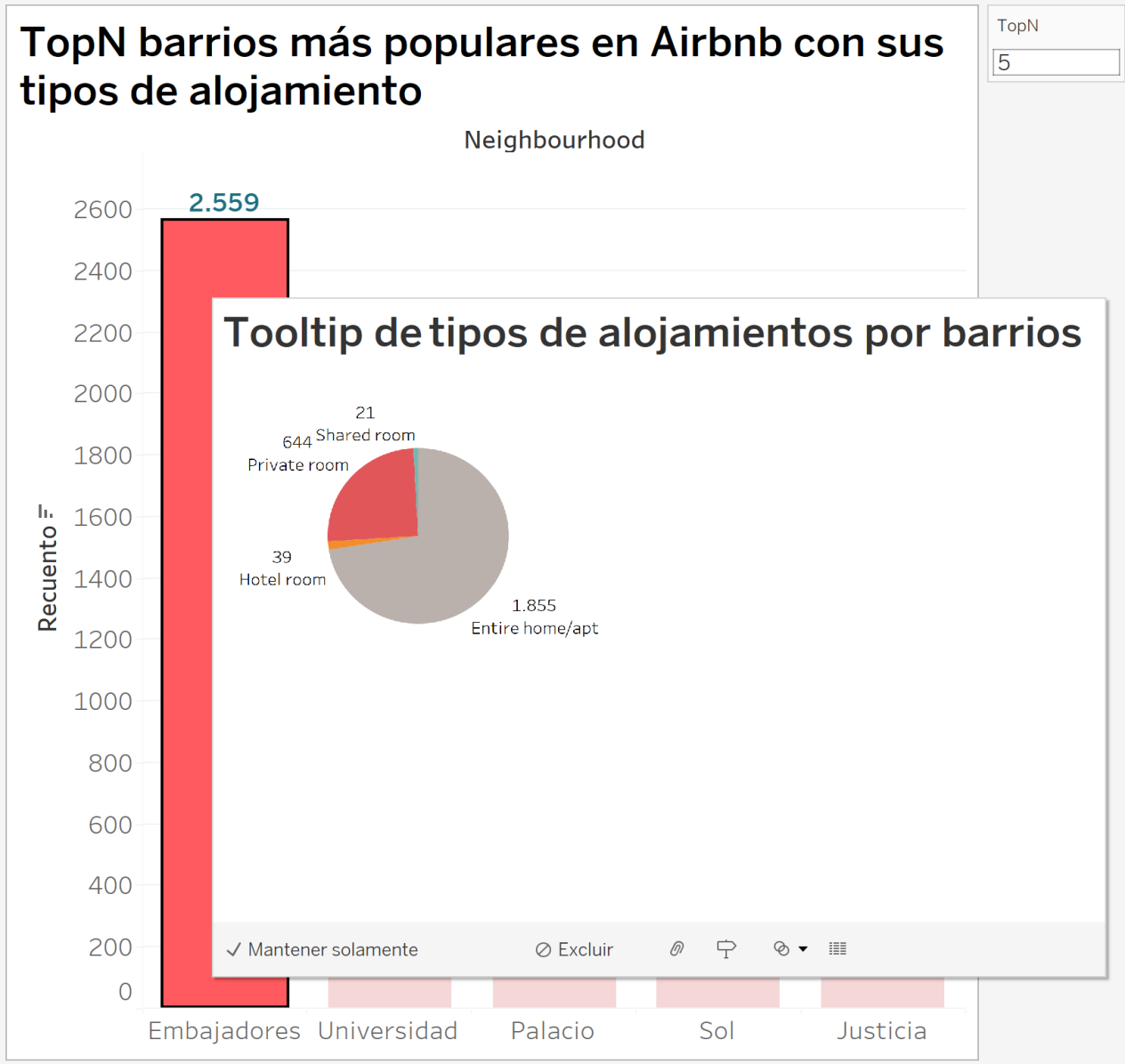
Crearemos los gráficos que representarán nuestros KPIs.

4.3.1 TopN barrios más populares en Airbnb con sus tipos de alojamiento

Creamos un gráfico de barras, utilizando la variable **Neighbourhood** en columnas y el recuento de la misma en filas para obtener la cantidad de veces que salen los barrios en el dataset, dicha cantidad la podemos ver encima de cada columna. Para poderlo todo más claro creamos un parámetro TopN aplicado como filtro a Neighbourhood para poder decidir cuántos barrios queremos ver.



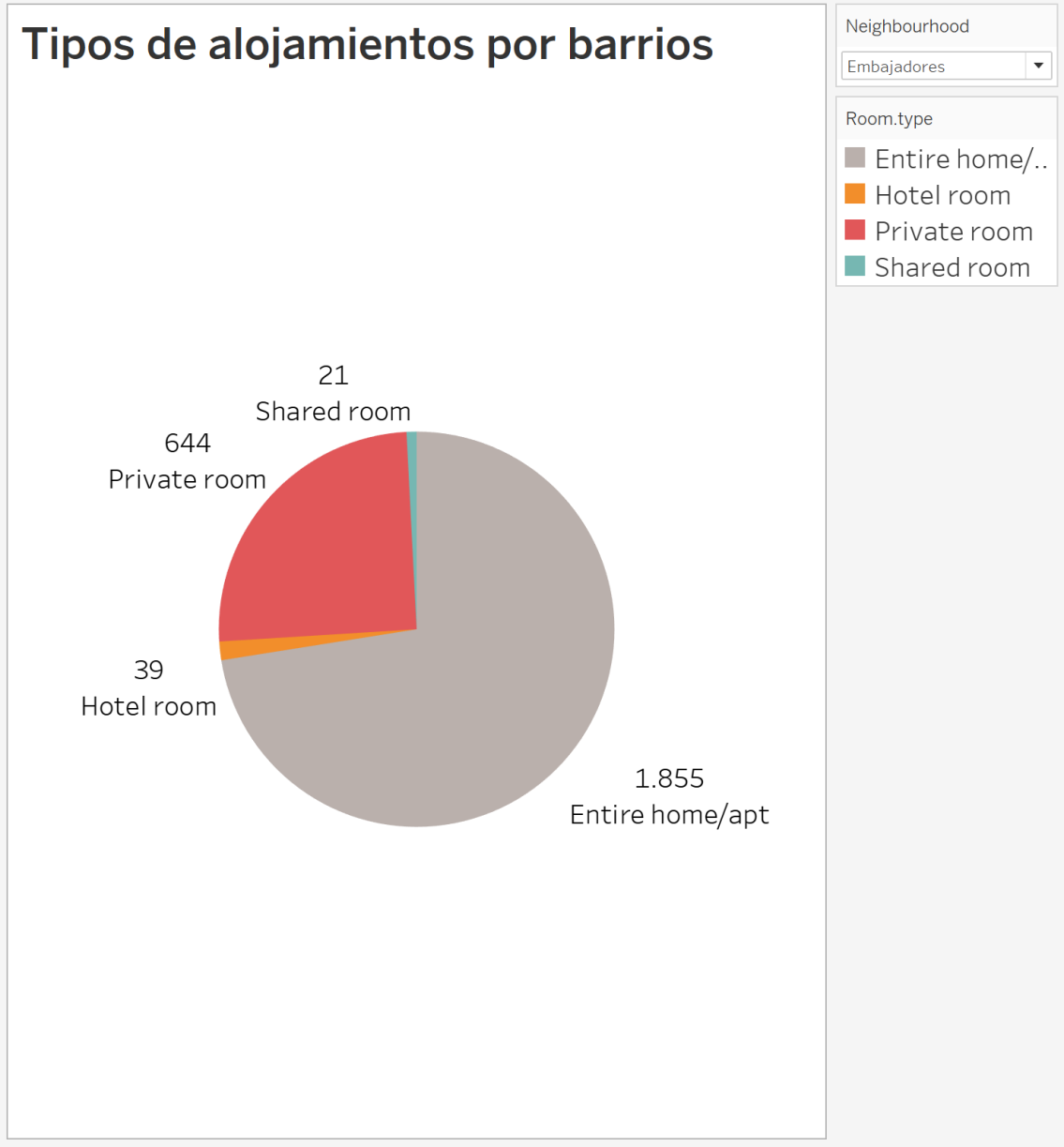
Como descripción emergente podremos ver un tooltip, que explicamos más adelante, en el que se indican los tipos de habitaciones y la cantidad según el barrio.



En la imagen podemos ver cómo se vería en el caso del barrio de Embajadores.

4.3.1.1 Tooltip tipos de alojamientos por barrios

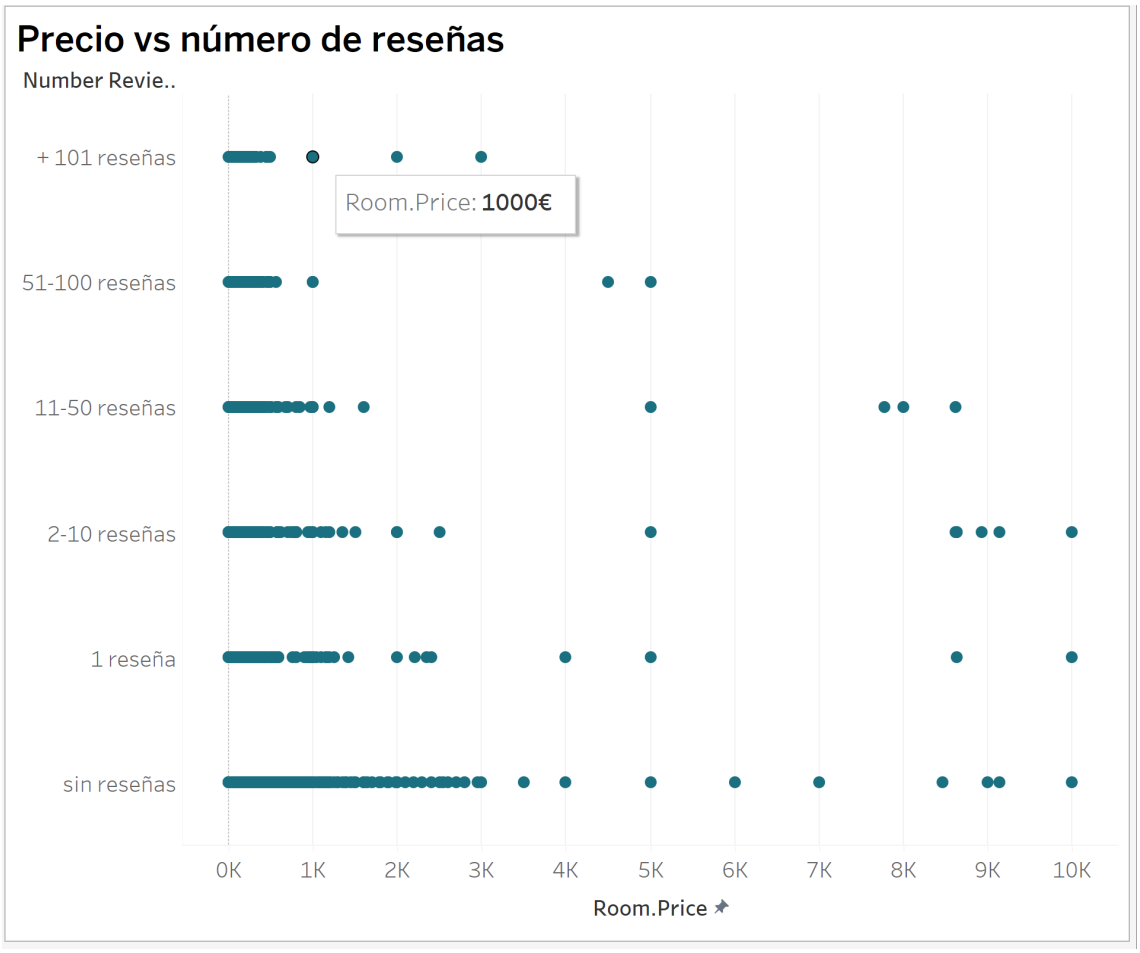
Creamos un gráfico circular, utilizamos las variables **Room.type** y **Neighbourhood**. En este caso para poder crearlo colocamos Room.type en colores y en etiqueta, para que se apliquen los colores según el número de tipos de habitación(Entire home/apt, Hotel room, Private room, Shared room) tendremos cuatro colores y para que salgan los tipo en el gráfico. Luego colocaremos el recuento de la misma en ángulo,para que se divida y por último en detalle para que se vea la cantidad del tipo correspondiente de habitación. Aplicaremos un filtro con Neighbourhood, para poder ver el gráfico correspondiente de cada barrio al seleccionarlo.



En la imagen podemos ver el resultado al elegir el barrio de Embajadores.

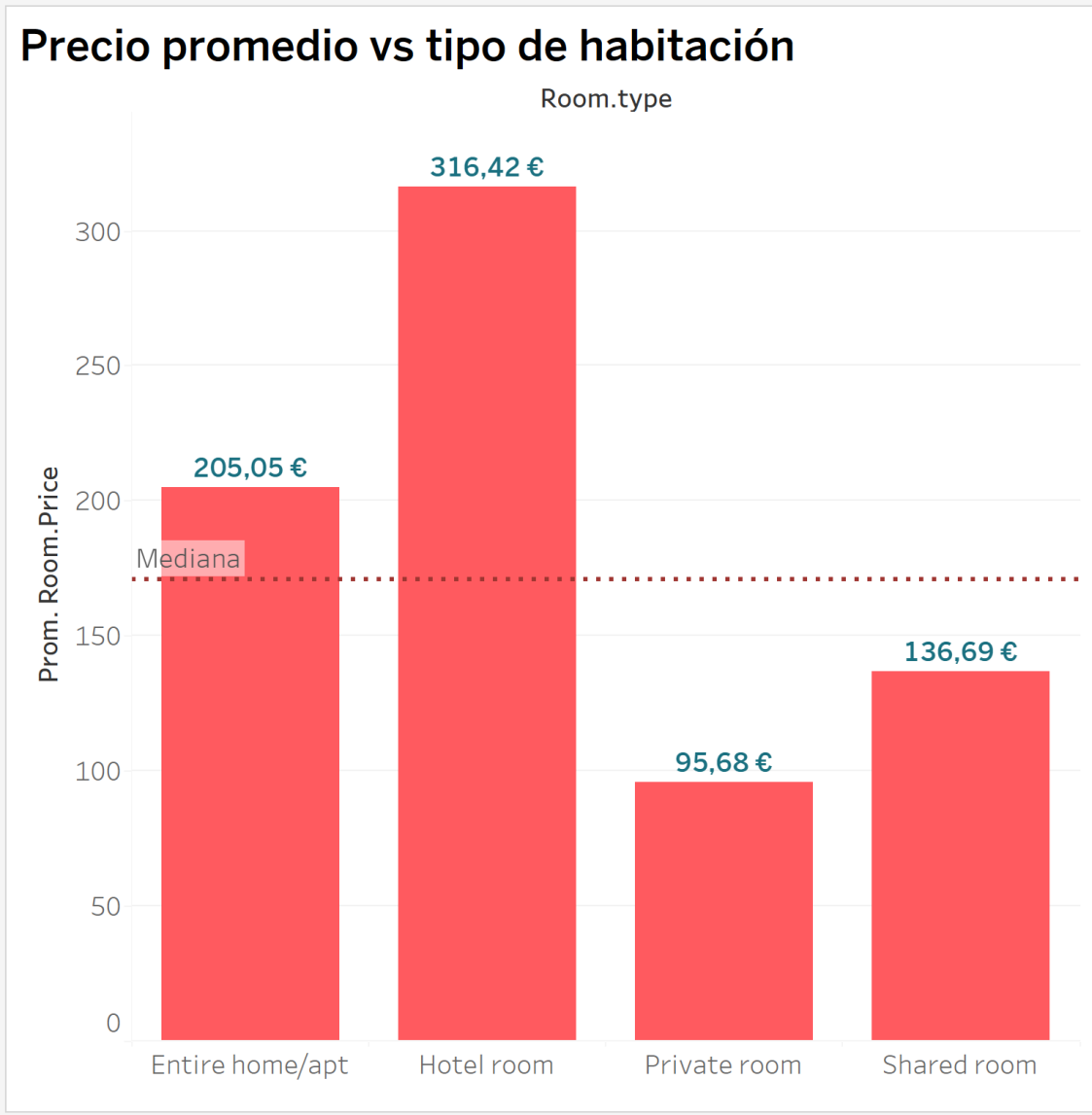
4.3.2 Precio vs número de reseñas

Creamos un gráfico de dispersión categorizado, utilizamos las variables **Room.Price** en columnas y **Number Reviews** en filas. Para obtener los círculos como forma, elegimos en marcas Círculo. Como descripción emergente, cada vez que nos coloquemos encima de uno de los círculos nos dirá el precio.



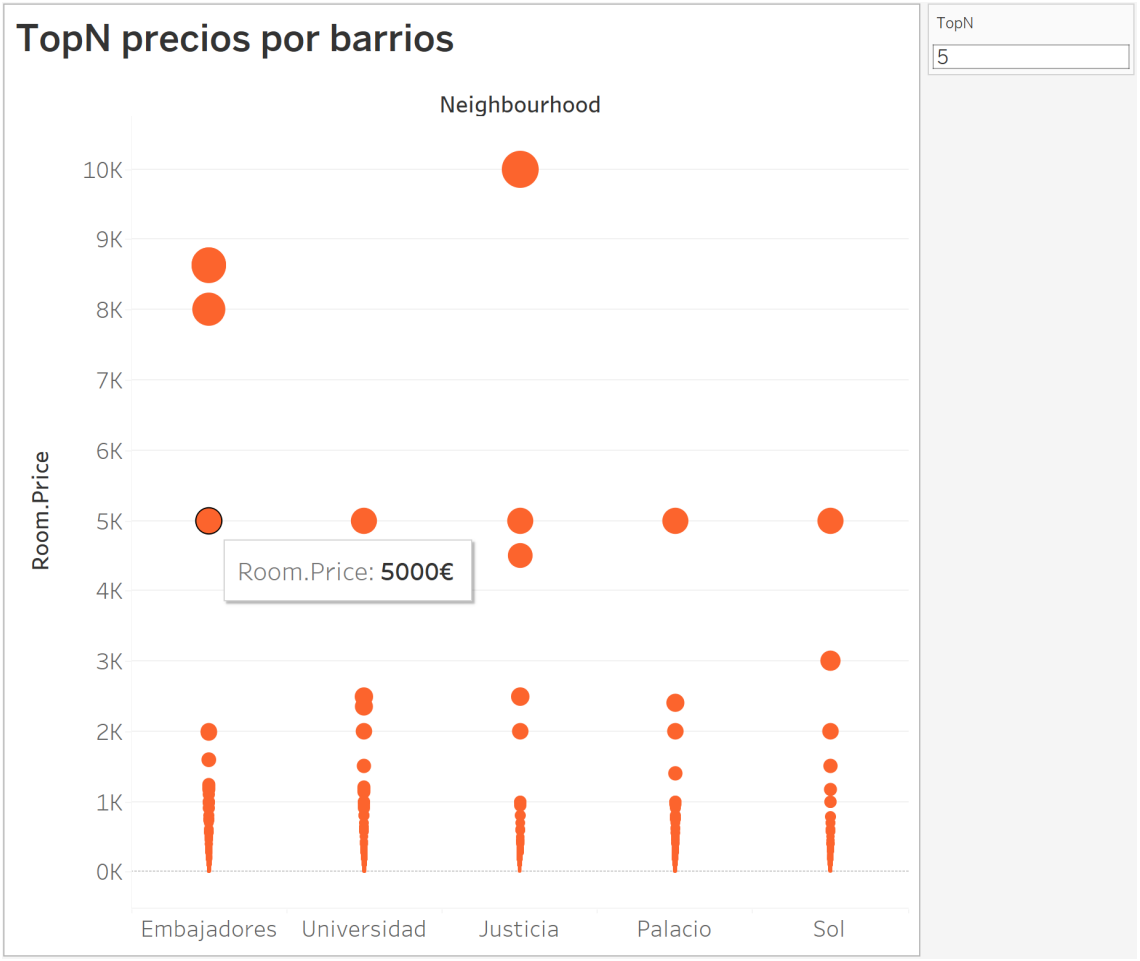
4.3.3 Precio promedio vs tipo de habitación

Creamos un gráfico de barras, utilizamos las variables **Room.type** en columnas y Promedio de **Room.Price** en filas. Añadimos una línea de referencia, que indicará la mediana. Encima de cada barra podemos ver la cantidad, para poder obtener el símbolo del euro(€), creamos un campo calculado llamado **Prom Room.Price,** en el que calculamos el promedio y luego cambiamos el formato de número a moneda.



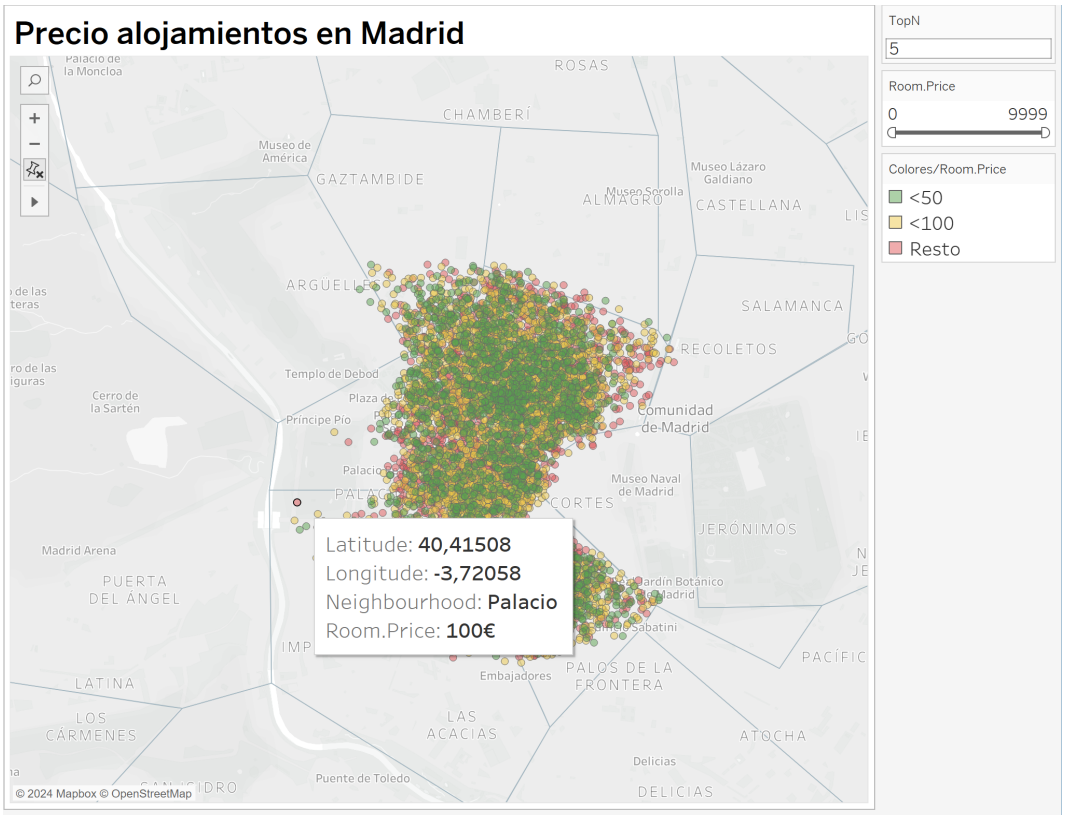
4.3.4 TopN precios por barrios

Creamos un gráfico de dispersión, utilizamos las variables **Neighbourhood** en columnas y **Room.Price** en filas. Para obtener los círculos como forma, elegimos en marcas Círculo. Además colocamos **Room.Price** en tamaño para que los círculos sean más grandes cuanto mayor es el precio. Aplicamos el parámetro TopN creado anteriormente como filtro de Neighbourhood, para poder ver el número de barrios deseados. y como descripción emergente, cada vez que nos coloquemos encima de uno de los círculos nos dirá el precio.



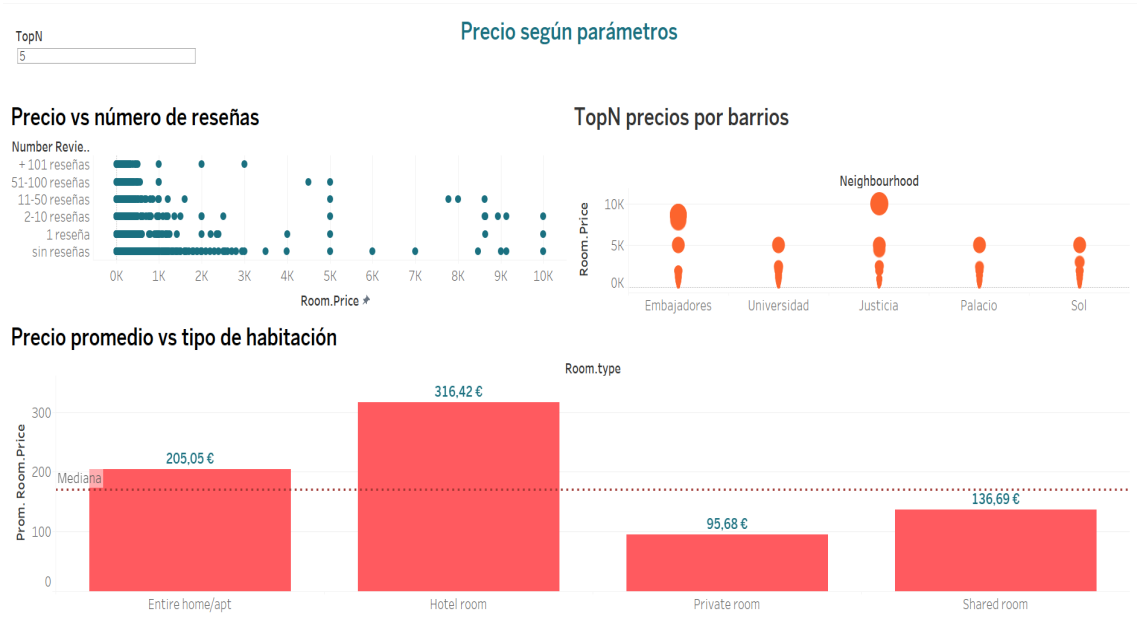
4.3.5 Precio alojamientos en Madrid

Creamos un gráfico geográfico, utilizamos las variables **Longitude** en columnas y **Latitude** en filas. Como filtros volvemos a usar el parámetro TopN para Neighbourhood y creamos un campo calculado llamado **Colores/Room.Price** donde establecemos la división de <50, <100 y Resto. En marcas, colocamos el campo calculado y especificamos los colores, en este caso verde, amarillo y rojo. Por otro lado en tamaño ponemos Room.Price para que varíe el tamaño según el valor. En la descripción emergente podremos ver la latitude, la longitude, el Neighbourhood y el Room.Price.



4.4 Dashboard

En este punto juntamos los gráficos según Mapa (Precio alojamientos en Madrid), Barrios Populares (TopN barrios más populares en Airbnb con sus tipos de alojamiento) y Precios (Precios vs número de reseñas, TopN precios por barrios, Precio promedio vs tipo de habitación).



En la imagen podemos ver el dashboard precio con título Precio según parámetros.

4.5 Historia

Para terminar hemos creado una historia, en primer lugar podemos ver el mapa, de esta forma vemos como están distribuidos los precios según la ubicación, vemos que están muy concentrados en el centro. Luego tenemos barrios populares, el top5 corresponde a Embajadores, Universidad, Palacio, Sol y Justicia y el tipo de habitación que predomina es Entire home/apt. Y por último, en precios podemos hacernos una idea de cómo varía el precio según el tipo de habitación, el barrio y que los sitios con mayores reseñas tienen unos precios menores que los que tienen menos.

5. Pre-procesamiento y Modelado

5.1 Preprocesamiento de Datos:

El preprocesamiento de datos es una fase crucial en cualquier proyecto de análisis de datos y modelado predictivo. En el contexto de nuestro proyecto sobre la predicción de precios de Airbnb en Madrid, se realizaron una serie de pasos para garantizar la calidad y la idoneidad de los datos antes de la construcción de modelos.

Antes del inicio de la selección y transformación de las variables, se corrigieron posibles errores en los datos, como formatos incorrectos o inconsistencias en la codificación de variables. Esto garantiza la coherencia y la fiabilidad de los datos utilizados en el análisis.

5.1.1. Selección de Variables:

La selección de variables es un paso crítico para identificar las características más relevantes para la predicción del precio de los listados de Airbnb en Madrid. Se realizaron análisis de correlación y pruebas de importancia de características para seleccionar las variables más informativas. Se descartaron variables que no contribuían significativamente a la predicción del precio, lo que ayudó a reducir la complejidad del modelo y mejorar su capacidad predictiva.

a. **Análisis Descriptivo de Variables:**

   En nuestro conjunto de datos de Airbnb, realizamos un análisis descriptivo detallado de varias variables relevantes para comprender mejor la distribución y las características de los datos. Aquí están las principales ideas:

   Room.Price: El precio medio de una habitación es de aproximadamente 164, pero la mediana es solo 60. Esto sugiere una distribución sesgada hacia la derecha, con algunos valores extremadamente altos (como el máximo de 9999) que están inflando la media. Además, el 75% de las habitaciones tienen un precio de 100 o menos. Esta amplia variabilidad en los precios podría afectar la capacidad de los modelos de machine learning para generalizar correctamente si no se maneja adecuadamente.

   Minimum.nights: La mayoría de las estancias requieren muy pocas noches, con una mediana de solo 2 noches. Sin embargo, el valor máximo es de 1125 noches, lo que indica que hay algunas estancias extremadamente largas en nuestro conjunto de datos. Esta

variabilidad en la duración de las estancias también puede influir en la predicción de los precios de las habitaciones.

Number.of.reviews: La mediana de la cantidad de reseñas por habitación es de solo 6, lo que sugiere que la mayoría de las habitaciones tienen pocas reseñas. Sin embargo, la media es mucho más alta, aproximadamente 34.875, lo que indica que algunas habitaciones tienen un número muy alto de reseñas. Esta discrepancia entre la mediana y la media podría indicar la presencia de valores atípicos o un sesgo en la distribución de las reseñas.

Number.of.reviews.per.month: Similar a la cantidad total de reseñas, la mayoría de las habitaciones tienen pocas reseñas por mes, pero hay algunas con un número muy alto de reseñas por mes. Esta variable puede proporcionar información adicional sobre la popularidad y la actividad de las habitaciones en la plataforma de Airbnb.

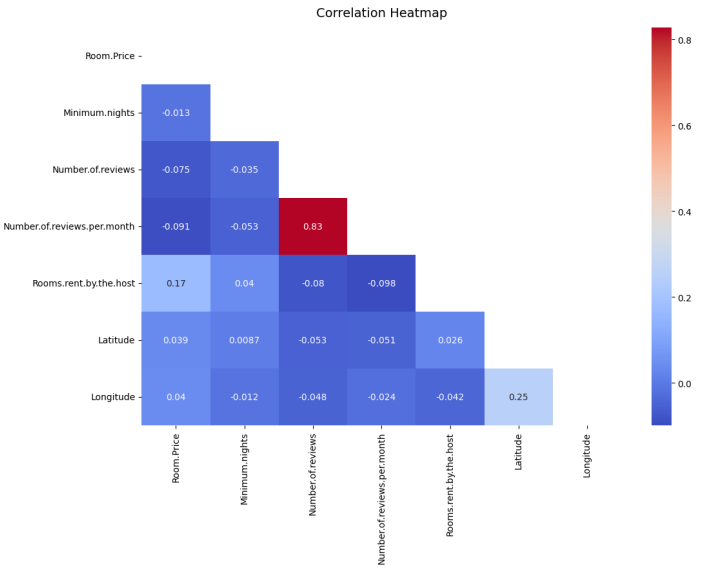
Rooms.rent.by.the.host: La mayoría de los anfitriones alquilan pocas habitaciones, con una mediana de solo 2 habitaciones. Sin embargo, el valor máximo es de 244, lo que sugiere la presencia de algunos anfitriones que alquilan un gran número de habitaciones. Este hallazgo resalta la variabilidad en el número de habitaciones que un anfitrión puede ofrecer y su impacto potencial en los precios.

Latitude y Longitude: Estas variables representan las coordenadas geográficas de las habitaciones. Sin un contexto geográfico adicional, no podemos realizar un análisis significativo de estas variables. Sin embargo, pueden ser importantes para modelos que consideran la ubicación geográfica como un factor en la fijación de precios.

El análisis descriptivo revela una gran variabilidad en los datos, con algunos valores extremos en varias columnas. Esta variabilidad puede influir en la capacidad de los modelos de machine learning para generalizar correctamente. Una estrategia para manejar esta variabilidad es convertir algunas variables numéricas en variables categóricas, especialmente aquellas con una amplia gama de valores únicos o valores extremos. Por ejemplo, convertir "**Number.of.reviews**" y "**Number.of.reviews.per.month**" en categorías puede reducir la complejidad de los datos y ayudar al modelo a capturar patrones más generales.

b. Evaluación de Correlación:

Iniciamos nuestro análisis examinando la relación entre las variables numéricas y el precio de las habitaciones. Observamos que algunas variables, como el número de habitaciones alquiladas por el anfitrión y las coordenadas geográficas (latitud y longitud), mostraron cierta correlación con el precio de las habitaciones. Sin embargo, estas correlaciones fueron moderadas en su mayoría, lo que sugiere que otras variables también podrían influir en el precio. Es esencial tener en cuenta que la correlación, aunque indicativa de asociación, no implica causalidad. Por lo tanto, debemos considerar cuidadosamente otras variables que puedan influir en el precio y evaluar su contribución al modelo de manera integral.



c. Análisis de Varianza (ANOVA):

Procedimos a realizar un análisis de varianza para las variables categóricas, específicamente el tipo de habitación, el número de revisiones, el número de revisiones por mes, entre otras. Descubrimos que estas variables tienen un efecto significativo en el precio de las habitaciones según los valores F y los valores p extremadamente pequeños obtenidos. Por ejemplo, el tipo de habitación mostró un valor F de 68.94 con un valor p de 2.33e-44, lo que indica un efecto altamente significativo en el precio de la habitación. Este hallazgo subraya la importancia de considerar las características específicas de las habitaciones al predecir sus precios, como si son apartamentos completos, habitaciones privadas o compartidas.

d. **Prueba de Hipótesis para Distribución Normal:**

Aplicamos pruebas de hipótesis para verificar si las variables numéricas seguían una distribución normal. Ninguna de las variables pareció seguir esta distribución, como lo indican los valores p extremadamente bajos obtenidos. Esto sugiere que es poco probable que estas variables se distribuyan normalmente en la población. Es fundamental tener en cuenta esta falta de normalidad al seleccionar y ajustar los modelos estadísticos, ya que muchos métodos asumen una distribución normal de los datos.

5.1.2. Transformaciones de Datos

El análisis del dataset limpio y la selección de atributos para predecir los precios de Airbnb en Madrid condujeron a la aplicación de varias técnicas de preprocesamiento de datos. Esto incluyó la normalización de atributos categóricos y numéricos para garantizar que estuvieran en un formato adecuado para el modelado predictivo.

a. Atributos Categóricos:

Utilizamos el *OneHotEncoder* con *ColumnTransformer* para transformar los atributos categóricos seleccionados. Estos incluyen: **Neighbourhood, Room_type, Availability_Cat, Review_category, Review_Count_Category** y **Time_category**.

Se observó una gran variabilidad en los datos de "**Number.of.reviews**" y "**Number.of.reviews.per.month**" durante el análisis exploratorio, por lo que fueron convertidos en atributos categóricos.

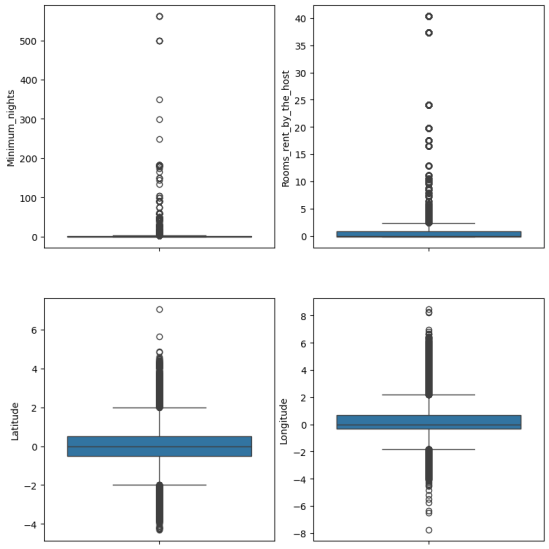
b. Atributos Numéricos:

Dado que las variables numéricas independientes no siguen una distribución normal o gaussiana, se les aplicó normalización. Además, se identificaron outliers en las siguientes columnas: **Minimum.nights, Rooms.rent.by.the.host, Latitude, Longitude.** Para manejar estos outliers, se aplicó *RobustScaler* a los atributos numéricos.

Después de las transformaciones, creamos una copia del DataFrame original y eliminamos la columna '**Time_category**'. El nuevo DataFrame resultante contiene 11 columnas con datos normalizados y transformados.

Además, se creó un DataFrame normalizado con 151 columnas, que incluyen las variables originales y las nuevas columnas generadas después de aplicar *OneHotEncoder* a los atributos categóricos.

A pesar de estas transformaciones, los valores atípicos persistieron en ciertos atributos. Se planteó la posibilidad de probar modelos eliminando estos valores atípicos para mejorar el rendimiento del modelo.



5.2 Modelo:

El objetivo principal de este estudio fue desarrollar un modelo predictivo capaz de estimar el precio de las habitaciones de Airbnb en un área específica. Inicialmente, se optó por aplicar un algoritmo de regresión lineal para predecir el precio de los inmuebles en función de diversas características seleccionadas. Sin embargo, debido a los resultados menos satisfactorios obtenidos con la regresión lineal, se exploraron alternativas para mejorar el rendimiento del modelo. Para ello, se evaluaron otros dos modelos de aprendizaje automático: *Random Forest* y *XGBoost*.

Estos modelos fueron seleccionados por su capacidad para manejar relaciones no lineales y capturar patrones complejos en los datos, aspectos que podrían mejorar la precisión de las predicciones en comparación con la regresión lineal. Además, se realizaron análisis adicionales para comprender mejor el impacto de ciertas características del conjunto de datos en el rendimiento de los modelos, lo que permitió identificar áreas potenciales de mejora y optimización del proceso predictivo.

5.2.1. Regresión Lineal:

La regresión lineal es un modelo simple que asume una relación lineal entre las características y la variable objetivo. En este caso, obtuvimos un R2 Score muy bajo de 0.0678 y un MAE de 180.54. Estos resultados indican que la regresión lineal no es adecuada para capturar la complejidad de los datos y predecir con precisión el precio de las habitaciones.

5.2.2. Random Forest:

Random Forest es un algoritmo de ensamble que combina múltiples árboles de decisión para mejorar la precisión y evitar el sobreajuste. En comparación con la regresión lineal, Random Forest mostró una mejora significativa con un R2 Score de 0.5122 y un MAE de 122.50. Esto sugiere que Random Forest es capaz de capturar relaciones no lineales en los datos y proporcionar predicciones más precisas del precio de las habitaciones.

5.2.3. XGBoost:

XGBoost es una implementación optimizada de Gradient Boosting que ha demostrado ser muy efectiva en una variedad de problemas de predicción. En este estudio, XGBoost superó tanto a la regresión lineal como a Random Forest, con un R2 Score de 0.6821 y un MAE de 96.87. Estos resultados sugieren que XGBoost es capaz de capturar relaciones más complejas en los datos y proporcionar predicciones más precisas del precio de las habitaciones de Airbnb.

5.2.4 Impacto de los Escenarios Planteados:

Para comprender mejor cómo ciertas características afectan el rendimiento de los modelos, se exploraron varios escenarios al eliminar características específicas del conjunto de datos.

	Caso	R2	MAE
0	Regresión Lineal	0.067833	180.541153
1	Random Forest	0.512228	122.504677
2	XGBoost	0.682136	96.868375
3	RF sin coordenadas	0.506223	123.159447
4	XG Boost sin coordenadas	0.679599	96.431223
5	RF sin Barrios	0.604443	107.191553
6	XG Boost sin Barrios	0.684410	97.450871
7	RF sin outliers	0.457007	0.586767

Por ejemplo, al eliminar las coordenadas geográficas (Latitud y Longitud), se observó una disminución en el rendimiento de los modelos, lo que sugiere que esta información es importante para predecir el precio de las habitaciones.

Del mismo modo, al eliminar la información sobre los barrios, se observó una mejora en el rendimiento de los modelos, lo que sugiere que esta característica puede no ser tan relevante para la predicción del precio como se pensaba inicialmente. Esta observación contradice la expectativa inicial de que los barrios tendrían un impacto significativo en el precio de las habitaciones de Airbnb. Es posible que otras características del conjunto de datos proporcionen información más relevante para predecir el precio.

Por otro lado, al eliminar los outliers, se observó una mejora significativa en el MAE, lo que sugiere que los outliers podrían estar introduciendo ruido en los modelos. Este hallazgo respalda la noción de que los outliers pueden distorsionar la capacidad de los modelos para encontrar patrones generales en los datos y hacer predicciones precisas. La exclusión de estos valores atípicos permitió que nuestros modelos se ajustaran de manera más efectiva a los datos restantes, lo que se tradujo en predicciones más certeras del precio de las habitaciones. Sin embargo, es importante tener en cuenta que esta mejora en la precisión se logró a expensas de una pérdida considerable de datos, lo que puede haber afectado la representatividad y la capacidad predictiva general de nuestros modelos. En futuros análisis, podríamos considerar métodos alternativos para abordar los outliers que minimicen la pérdida de datos mientras aún mejoran la precisión del modelo.

En conclusión, XGBoost demostró ser el modelo más efectivo para predecir el precio de las habitaciones de Airbnb en este estudio, superando tanto a la regresión lineal como a Random Forest. Además, se destacó la importancia de características como las coordenadas geográficas y la información sobre los barrios para mejorar la precisión de los modelos. Sin embargo, se debe tener cuidado al manejar outliers, ya que su eliminación puede llevar a una pérdida de información importante. En futuros estudios, se pueden explorar técnicas más avanzadas de procesamiento de datos y modelado para mejorar aún más el rendimiento del modelo predictivo.

5.3 Pipeline:

Para simular un despliegue del modelo y comprender cómo se podría implementar en un entorno de producción, se creó un pipeline completo que incluye tanto el preprocesamiento de datos como el entrenamiento del modelo. Un pipeline es una estructura eficiente y ordenada que encadena

diferentes pasos de procesamiento de datos y modelado en un flujo de trabajo unificado. En este caso, se creó un pipeline para realizar el preprocesamiento de los datos y entrenar un modelo de regresión XGBoost para predecir el precio de las habitaciones de Airbnb.

El proceso de creación del pipeline se dividió en varias etapas:

5.3.1 Preprocesamiento de datos: Se definió un preprocesador que incluye dos transformadores: uno para las características numéricas y otro para las características categóricas. Las características numéricas se normalizaron utilizando RobustScaler, mientras que las características categóricas se codificaron mediante OneHotEncoder.

5.3.2. Definición del modelo: Se configuraron los hiperparámetros del modelo XGBoost para optimizar su rendimiento. Estos hiperparámetros fueron seleccionados previamente mediante técnicas de búsqueda y validación de hiperparámetros.

5.3.3. Creación del pipeline: Se combinaron los transformadores de preprocesamiento y el modelo XGBoost en un único pipeline utilizando la clase Pipeline de scikit-learn.

```
# Se crea una copia del dataframe original con la eliminación de las columnas 'Number_of_reviews' y 'Number_of_reviews_per_month'
df_airbnb_transformed = df_airbnb_clean_vs_1.drop(['Number_of_reviews', 'Number_of_reviews_per_month'], axis=1)

# Definir las columnas numéricas y categóricas
numeric_features = ['Minimum_nights', 'Rooms_rent_by_the_host', 'Latitude', 'Longitude']
categorical_features = ['Neighbourhood', 'Room_type', 'Availability_Cat', 'Review_category', 'Review_Count_Category', 'Time_category']

# Definir los transformadores para las columnas numéricas y categóricas
numeric_transformer = RobustScaler()
categorical_transformer = OneHotEncoder(drop='first')

# Crear el modelo con los hiperparámetros obtenidos
xgbr = xgb.XGBRegressor(
    colsample_bytree=0.9,
    learning_rate=0.1,
    max_depth=5,
    min_child_weight=3,
    n_estimators=200,
    subsample=0.9,
    random_state=42
)

# Combinar los transformadores en un preprocesador
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)])

# Definir la pipeline
pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('regressor', xgbr)
])
✓ 0.0s
```

Una vez creado el pipeline, se utilizó para entrenar el modelo con los datos de entrenamiento y realizar predicciones sobre los datos de prueba. Además, se demostró cómo utilizar el pipeline para hacer predicciones sobre nuevos datos utilizando un ejemplo de un nuevo DataFrame llamado new_data.

```python
# Supongamos que new_data es el nuevo DataFrame
new_data = pd.DataFrame({
    'Minimum_nights': [3],
    'Rooms_rent_by_the_host': [13],
    'Latitude': [40.40],
    'Longitude': [-3.70175],
    'Neighbourhood': ['Embajadores'],
    'Room_type': ['Entire home/apt'],
    'Availability_Cat': ['Hasta 30 días'],
    'Review_category': ['0-1/mes'],
    'Review_Count_Category': ['11-50 reseñas'],
    'Time_category': ['8 semanas - 6 meses']
})

# Usar la pipeline para hacer predicciones
predictions = pipeline.predict(new_data)

print(predictions)
```

✓ 0.0s

```
[164.99863]
```

El código que se presenta a continuación es una extensión de nuestro proceso de predicción de precios para habitaciones de Airbnb. En lugar de ofrecer una predicción puntual del precio, hemos implementado un enfoque que proporciona un rango de precios. Esta decisión se basa en la naturaleza de nuestro modelo de regresión, que tiene un coeficiente de determinación (R2) del 68%. Esto significa que alrededor del 68% de la variabilidad en los precios puede ser explicada por las características que hemos incluido en nuestro modelo.

```python
# Usa la pipeline para hacer predicciones
predictions = pipeline.predict(new_data)

# Calcula el rango de precios basado en el MAE
lower_bound = int(predictions[0] - 96.87)
upper_bound = int(predictions[0] + 96.87)

# Redondea el precio a dos decimales
predicted_price = round(predictions[0], 2)

print(f"El precio predicho es aproximadamente {predicted_price}, con un rango de {lower_bound} a {upper_bound}.")
```

```
El precio predicho es aproximadamente 165.0, con un rango de 68 a 261.
```

Una vez utilizado la pipeline para hacer predicciones sobre un nuevo conjunto de datos, representado por new_data, calculamos un rango de precios basado en el error absoluto medio (MAE) del modelo, que en este caso es aproximadamente 96.87. Sumamos y restamos este valor al precio predicho para obtener un rango de precios. Finalmente, redondeamos el precio predicho a dos decimales y lo imprimimos junto con el rango de precios obtenido.

Este enfoque nos permite ofrecer una estimación más realista del precio de una habitación, teniendo en cuenta la incertidumbre asociada con nuestro modelo. Esto puede ser especialmente útil para los usuarios que deseen tener una idea general del precio esperado, pero también deseen conocer la variabilidad en esa estimación. Este método de presentación de los resultados es parte de nuestro esfuerzo por proporcionar una información más completa y útil a nuestros usuarios.

Es importante destacar que, si bien se utilizó el modelo XGBoost en este pipeline debido a su mejor rendimiento en comparación con otros modelos, como se mencionó anteriormente, la elección del modelo no es concluyente debido al ruido presente en el dataset. Por lo tanto, este ejemplo se presenta como una forma de automatizar el proceso de predicción del precio de las habitaciones, pero se recomienda realizar un análisis más exhaustivo y considerar otras técnicas de modelado para obtener resultados más robustos.

Además, el desempeño del modelo en un entorno de producción real puede verse afectado por diversos factores, como la variabilidad en los datos de entrada y las condiciones cambiantes del mercado. Por lo tanto, es fundamental realizar una monitorización continua del modelo y realizar ajustes según sea necesario para mantener su precisión y fiabilidad a lo largo del tiempo.

Después de completar el proceso de preprocesamiento y modelado de los datos para predecir los precios de las habitaciones de Airbnb en Madrid, hemos llegado a varias conclusiones y reconocido algunas limitaciones importantes en nuestro estudio.

Conclusiones:

- **Selección de modelo:** A pesar de que la tarea asignada inicialmente fue desarrollar un algoritmo de regresión lineal para predecir el precio de los inmuebles en función de las características seleccionadas, se encontró que esta opción no produjo los resultados más satisfactorios. XGBoost, en cambio, demostró ser el modelo más efectivo, superando tanto a la regresión lineal como a Random Forest. Su capacidad para capturar relaciones complejas en los datos resultó en predicciones más precisas y confiables.
- **Importancia de las características:** Se identificó que características como las coordenadas geográficas y la información sobre los barrios fueron importantes para mejorar la precisión de los modelos. Sin embargo, la relevancia de ciertas variables puede variar dependiendo del contexto y del conjunto de datos.
- **Manejo de outliers:** La eliminación de outliers condujo a mejoras significativas en el rendimiento del modelo, pero también resultó en una pérdida considerable de datos. En futuros estudios, podría ser beneficioso explorar métodos alternativos para abordar los outliers que minimicen esta pérdida de información.
- **Pipeline de producción:** La implementación de un pipeline completo que incluye tanto el preprocesamiento de datos como el entrenamiento del modelo es fundamental para simular el despliegue del modelo en un entorno de producción. Esto facilita la automatización del proceso de predicción y garantiza la coherencia en el flujo de trabajo.

Limitaciones:

- **Ruido en el dataset:** Aunque XGBoost mostró el mejor rendimiento en nuestro estudio, la presencia de ruido en el dataset puede haber influido en esta elección. Es importante reconocer que la elección del modelo no es concluyente y que se deben considerar otras técnicas de modelado para obtener resultados más robustos.
- **Generalización de los resultados:** Los resultados obtenidos en este estudio pueden ser específicos para el conjunto de datos y las condiciones del mercado en Madrid. Por lo tanto, es crucial realizar un análisis más exhaustivo y considerar la generalización de los modelos a diferentes contextos geográficos y temporales.

- **Variabilidad en los datos:** El desempeño del modelo en un entorno de producción real puede verse afectado por la variabilidad en los datos de entrada y las condiciones cambiantes del mercado. Es importante tener en cuenta estas variaciones y realizar ajustes continuos en el modelo según sea necesario para mantener su precisión y fiabilidad a lo largo del tiempo.