Informe sobre el Análisis de Airbnb Listings en Madrid

Proyecto Final

Giorgia Scuderi

Griselda Vilar

Katerina Margariti

Yesenia Andraca

Febrero 2023

Fase 1. Definición del Dataset y validación de datos

1.1. Exploración inicial de los datos.

El dataset elegido para realizar el análisis son los listings-Airbnb.

Como primera tarea realizamos el análisis inicial, revisando las columnas que contiene el dataset, podemos observar las siguientes:

```
df.columns
✓ 0.7s
Index(['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'],
      dtype='object')
```

Con el siguiente método **.info()** comprobamos que el dataframe contiene:

- Un total de 89 columnas, 21 de tipo float64, 12 de int64 y 56 de tipo object, y 14,779 entradas.

Tras el análisis de cada una de las columnas, se decide hacer la primera eliminación de 46 columnas que no se utilizarán en ningún momento, considerando que son columnas que no nos aportan a lo que de primera instancia queremos analizar, y entre otras muchas de las columnas contienen información nula.

Tras la primera eliminación de columnas, verificamos los valores únicos de la columna “City”, comprobando que la mayoría de entradas corresponden a Madrid, con lo cual se decide realizar un nuevo dataframe solo con los datos de Madrid.

```
#En la columna "City" se seleccionan los datos cuyo valor sea igual a "Madrid"
df_madrid = df[(df['City'] == 'Madrid')]
```

Quedando con un total de 13,207 líneas.

1.2. Comprobación de columna Zipcode.

Se decide realizar la comprobación de la columna “Zipcode”, porque se considera que será una columna de vital importancia para nuestra visualización de datos en Tableau.

La columna cuenta con 439 datos nulos, de los cuales se intentará comprobar si en la columna “Street” existe algún código postal de los que no contengan la información.

En una nueva columna (“df_ziocode_n”) extraemos el Zipcode de la columna Street, indicando en nuestro código que nos extraiga los valores que no sean de formato str, y se realiza una comparación de la nueva columna con la columna “Zipcode” de nuestro dataframe.

```
#Comparar columna zipcode con la nueva columna creada en el paso anterior
df_madrid["comparar_zipcodes"]=df_madrid["Zipcode"]==df_madrid["df_zipcode_n"]
df_madrid[["Zipcode", "df_zipcode_n", "comparar_zipcodes"]]
```

✓ 0.2s MagicPython

	Zipcode	df_zipcode_n	comparar_zipcodes
3	28005	28005	True
4	28013	28013	True
7	28005	28005	True
8	28013	28013	True

Con lo cual se llega a la conclusión de que nuestra primera suposición es falsa, ya que las mismas filas NaN del Zipcode corresponden a las mismas de la columna df_zipcode_n.

Tras comprobaciones se detecta que la columna contiene códigos erróneos, ya que un código postal debe estar constituido por 5 cifras, y no es el caso de esta. Con lo cual se deciden hacer los cambios de los Zipcode correctos.

Al analizar y comprobar que no se pueden extraer Zipcode de alguna otra columna, se decide eliminar los 439 datos de Zipcode NaN, ya que solo representan el 3.32%, dejando un total de 12,768 entradas.

1.3. Normalización de los valores de las tablas.

Se realiza la comprobación de espacios, tildes y a la unificación de valores de diversas columnas que nos servirán más adelante, entre las cuales están:

Columna “Host Location” se unifica por “Madrid, Spain”

Columna “State” por “Comunidad de Madrid”.

Otra columna importante para nuestro análisis, es la columna de Price, con lo cual se trata de revisar si es viable considerar el promedio de la columna Price tomando en cuenta la columna “Neighbourhood Cleansed”.

```
df_madrid_neig_clean=(
    df_madrid.groupby('Neighbourhood Cleansed')
    .agg({'Price':'mean', 'ID':'count'})
    .sort_values(by='Price', ascending=False)
)
df_madrid_neig_clean
```

✓ 0.2s

Neighbourhood Cleansed	Price	ID
El Plantío	280.000000	3
Palomas	142.833333	12
Recoletos	135.606250	161
Fuentelareina	127.666667	3
Castellana	115.505747	88

Comprobamos que al tomar en cuenta esta columna, el precio en alguno de los barrios sale disparado, ya que solo contiene información de 3 filas, con lo cual no es viable, ya que todos los datos NaN de Zipcode donde el barrio sea “El Plantío” saldrán disparados, y esto nos puede afectar a la media de los alquileres.

Con lo cual se decide comprobar con la columna “Neighbourhood Group Cleansed”

```
df_madrid_neig_gpclean=(
    df_madrid.groupby("Neighbourhood Group Cleansed")
    .agg({'Price':'mean', 'ID':'count'})
    .sort_values(by='Price', ascending=False)
)
df_madrid_neig_gpclean
```

✓ 0.1s

Neighbourhood Group Cleansed	Price	ID
Salamanca	85.416565	823
Chamartín	73.894737	342
Centro	72.837199	6600
Chamberí	72.492927	920
Retiro	70.369727	405
Moncloa - Aravaca	65.069136	406
Hortaleza	57.795580	181
Barajas	53.551282	78
Tetuán	52.277136	433
Fuencarral - El Pardo	51.834532	139

Comprobamos y decidimos que al utilizar esta columna, la media de los precios no es tan alta, ya que la cantidad mínima son 43 datos comparado. Así que se decide crear una nueva columna “Price_new”, sustituyendo los NaN por la media del precio por “Neighbourhood Group Cleansed”

Para los valores NaN de las columnas “Weekly Price” y “Monthly Price”, se decide multiplicar el valor de la columna “Price_new” por 7 que corresponde a los días de la semana, y en otra nueva columna por 30 para los valores del mes.

En las columnas “Bathrooms”, “Bedrooms”, y “Beds”, se deciden sustituir los NaN por la media tomando en cuenta el tipo de alojamiento, siempre redondeando, ya que en el caso de Bathrooms, nos encontramos en el caso que pueden existir .5, ya que considera conforme no tiene ducha y por estos casos se dan valores de 1.5,2.5 entre otros.

Se deciden crear nuevas columnas, para que en nuestra métricas en caso que comprobemos que la calidad de datos no es buena, se pueda siempre regresar a las columnas originales, y no perder dato alguno.

Para los valores de tipo Object, se decide que a los valores NaN sustituir por “Not defined”.

```
#Se realiza el cambio a "Not defined", de los valores cuya columnas no son de
tipo Float64 o int64
df_madrid2["Name"]=df_madrid2["Name"].fillna("Not defined")
df_madrid2["Description"]=df_madrid2["Description"].fillna("Not defined")
df_madrid2["Host Since"]=df_madrid2["Host Since"].fillna("Not defined")
df_madrid2["Host Verifications"]=df_madrid2["Host Verifications"].fillna("Not
defined")
df_madrid2["Neighbourhood"]=df_madrid2["Neighbourhood"].fillna("Not defined")
df_madrid2["Amenities"]=df_madrid2["Amenities"].fillna("Not defined")
```

Para los valores de tipo numérico, no se realiza esta sustitución ya que, en la revisión de nuestro dataset en SQL, se comprueba que a la hora de cargar nuestra base de datos las columnas de tipo numérico al tener “Not defined”, da error ya que esta sentencia no es de valor numérico, es por esa razón que solo se decide sustituir las columnas mostradas en la imagen.

1.4. Creación de la columna Superhost.

Nuestro análisis hemos decidido enfocarlo a la creación de la nueva columna Superhost, para así comprobar si al ser superhost o no, influye en el precio final.

Con lo cual con la columna “Features” se extraen los datos del host, en donde extraemos si contine “Host Is Superhost”, en nuestra columna nueva nos indicará con un “Yes” en caso que si es Superhost y “No” en caso contrario.

```
df_madrid3['Is Super Host'] = np.where(df_madrid3['Type_Host'] == 'Host Is
Superhost', True, False)
df_madrid3['Is Super Host YES/NO'] = np.where(df_madrid3['Type_Host'] == 'Host
Is Superhost', 'Yes', 'No')
df_madrid3[["Type_Host", "Is Super Host", "Is Super Host YES/NO"]]
```

✓ 1.1s MagicP

	Type_Host	Is Super Host	Is Super Host YES/NO
3	Host Has Profile Pic	False	No
4	Host Has Profile Pic	False	No
7	Host Is Superhost	True	Yes
8	Host Has Profile Pic	False	No
9	Host Has Profile Pic	False	No

Se procede a eliminar las columnas que no necesitaremos para continuar con el análisis exploratorio de datos, el modelado y la visualización de las métricas, contando con el nuevo archivo con 12,768 entradas y un total de 42 columnas.

Fase 2. Análisis Exploratorio y modelado

El proceso se ha dividido en 3 partes: “Análisis exploratorio y calidad del dato en R”, “Análisis exploratorio y Visualización de datos” y “regresiones lineales”.

2.1. Análisis exploratorio y calidad del dato en R.

Para evaluar la calidad de los datos de la base de datos original se procede a realizar un análisis exploratorio. Esta fase es importante en la medida que permite conocer si las hipótesis establecidas en un principio son las adecuadas, garantizar mejores resultados y una mayor validez estadística, así como conocer de qué manera influyen en los análisis y resultados. Para realizar este cometido, se ha realizado una evaluación de los valores “Nulos” o “Missing data” con el objetivo de evaluar la calidad de la base, identificar datos erróneos o incompletos y comprobar la existencia de patrones de relación en la base de datos. Finalmente, evaluar la pérdida de datos y calidad de cada variable.

Después de esta primera fase se procede a excluir variables con un elevado porcentaje de valores perdidos, imputar valores, completar y recuperar datos de variables clave para el estudio, así como corregir o modificar datos erróneos. Uno de los primeros pasos ha consistido en contabilizar los valores faltantes, principalmente de las variables numéricas, como valores NAN.

Para ello, se procede a representar los datos mediante gráficos que nos ayuden a identificar las mejores variables y aquellas con menor calidad para el estudio además de aplicar unas primeras funciones que permitan contabilizar e identificar los mismos.

Para aquellas variables con valores por encima del 5% de los valores perdidos se procede a evaluar si estas deben ser eliminadas del dataset final o si es pertinente recuperar la información.

Para las variables con un porcentaje de valores perdidos por encima del umbral se procede a evaluar con Python la posibilidad de eliminar o calcular los valores. En particular se valoran las siguientes: “Weekly.Price”, “Monthly.Price”, “Square.Feet”, “Security.Deposit”, “Review.Scores.Rating”, “Review.Scores.Accuracy”, “Review.Scores.Communication”, “Review.Scores.Location”, “Review.Scores.Cleanliness”, “Review.Scores.Value”, “Review.Scores.Checkin”, “Reviews.per.Month” y “Host.Response.Rate”.

Para las variables con valores de pérdida de datos por encima del 30%, se procede a eliminar del estudio: “Square.Feet”, “Host.Acceptance.Rate”, “License” y “Jurisdiction.Names” (realizado en Python). Se decide recuperar el dato mediante cálculos en Python de variables que se consideran importantes, es decir, realizando una estimación para cada una. Asimismo, se procede a completar las variables “Beds” y “Bedrooms”.

Posteriormente al análisis de la estructura, composición y contenido de la base de datos procedemos a su visualización y exploración. Para ello realizamos distintos gráficos que nos permiten ver mejor la relación entre variables y su calidad. Para esta tarea se realizan los siguientes gráficos, tanto para las variables “integer” y numéricas como para las variables “string” o “character”.

Gráfico 1. Proporción de datos faltantes y para comparar la calidad de las variables.

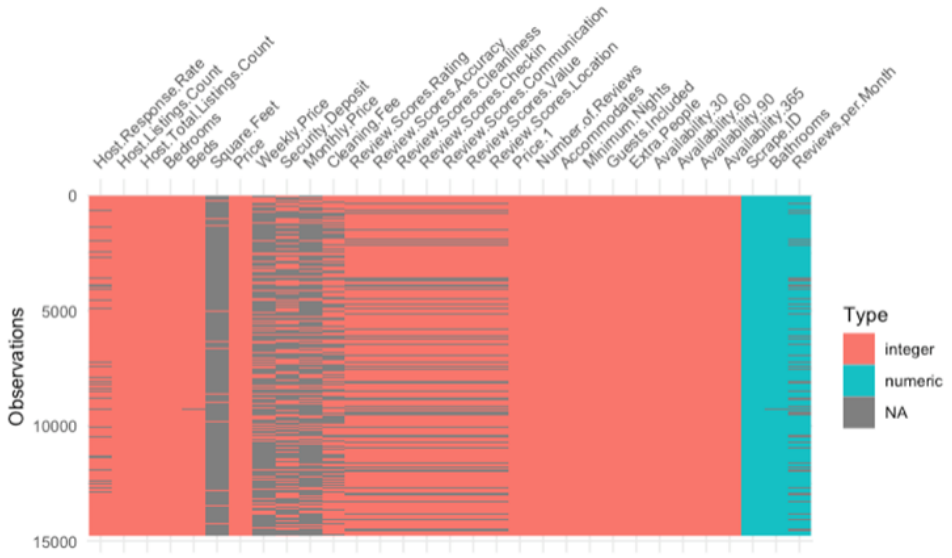


Gráfico 2. Porcentajes de pérdidas de datos sobre el total de la composición del data frame.

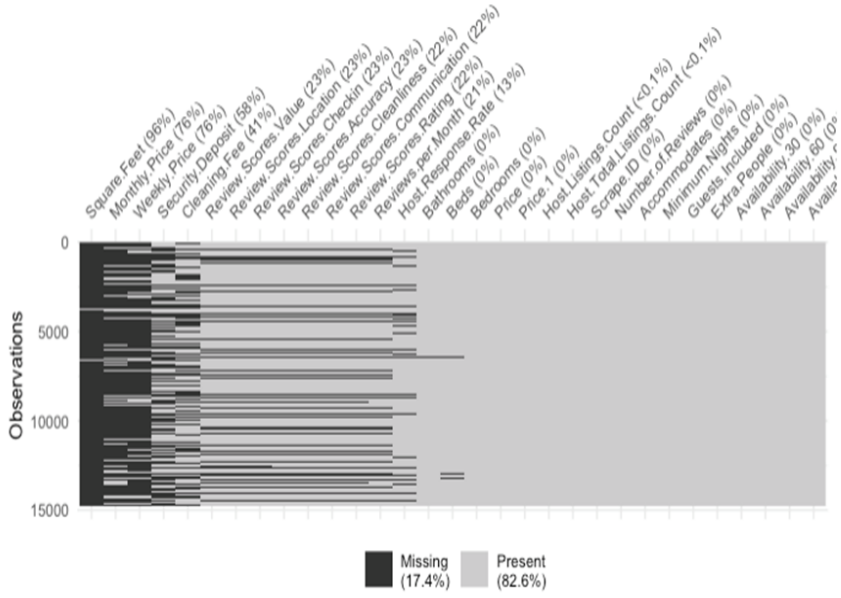


Gráfico 3. Clúster y datos pérdidos.

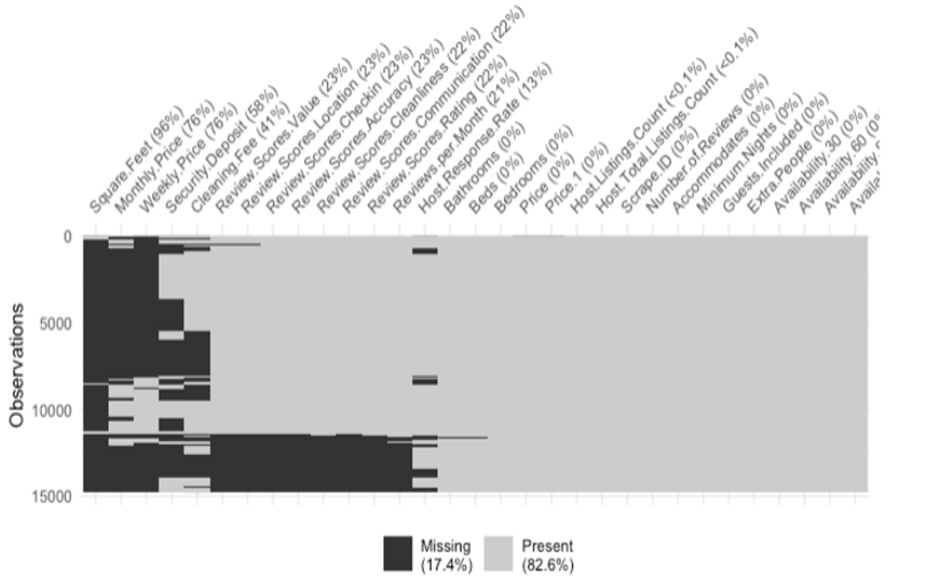
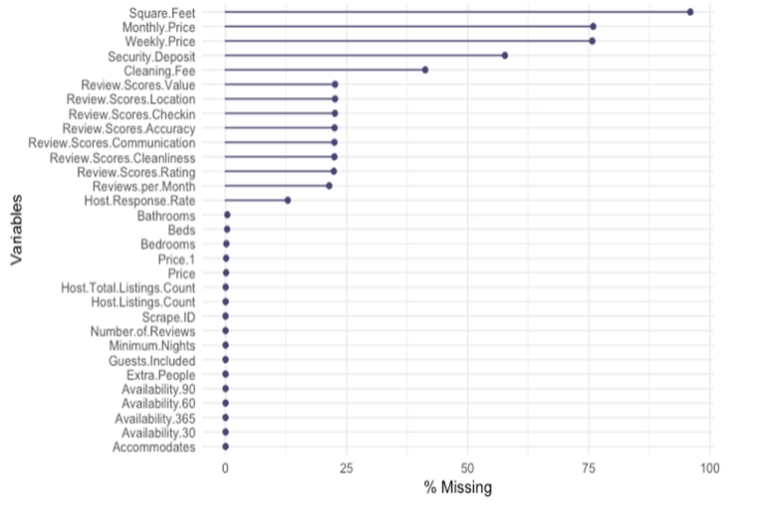
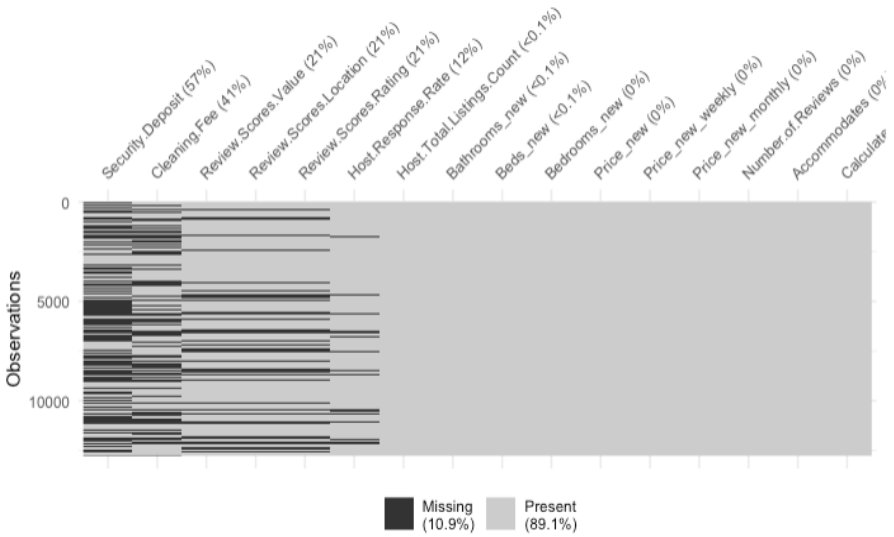


Gráfico 4. Número de valores perdidos por variable.



La misión de esta fase ha sido comprobar la calidad de los datos para tomar decisiones sobre valores nulos e identificar las variables más importantes. Qué

valores rellenar, qué valores omitir, qué valores unificar o aplicar funciones o algoritmos. Esto justifica las acciones de limpieza implementadas en una segunda fase de exploración y análisis realizado en Python sobre el conjunto de datos total y para la obtención del dataset final del proyecto.



2.2. Análisis exploratorio y visualización de datos.

Esta fase se ha centrado en comprobar la existencia de valores dispares o “outliers”, así como determinar el grado de distribución y dispersión de los datos. Este análisis sirve de base para finalizar las labores de limpieza y normalización de datos. También ayuda a identificar valores que modifiquen las estimaciones estadísticas. Esta comprobación es importante porque un único valor atípico puede transformar la media.

1. Distribución de datos y análisis de las variables seleccionadas, para comprobar los valores de las variables. Para esta tarea hemos realizado una inspección preliminar del dataset a través de las siguientes funciones:

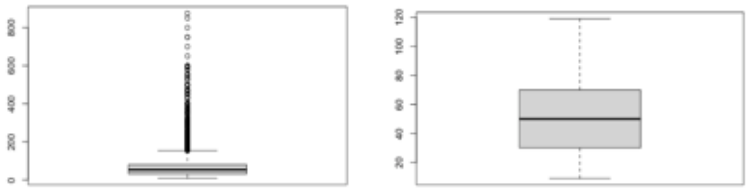
- Summary().
- Búsqueda del valor máximo y mínimo de cada variable.
- La posición en la que se encuentra ese valor máximo.
- Valor mínimo para cada variable.

- Posición valor mínimo.
- Medias para variables.
- Valor trim.
- Varianza.
- Desviación típica.
- Medianas.
- Cuartiles.

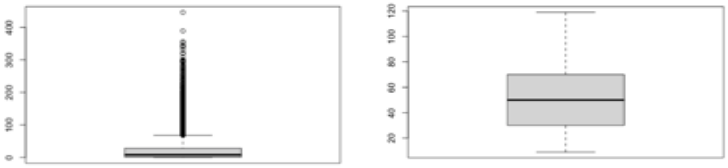
Para conservar los datos de las variables clave y principales se emplea la utilización de funciones que permitan omitir los valores nulos o NAN.

Para complementar estas tareas se representan una serie de gráficos Boxplot que nos ayudan a construir visualmente esta información y resultados, comparar el comportamiento de las variables y realizar unas primeras transformaciones y manejos de los datos para comprobar el comportamiento de los mismos.

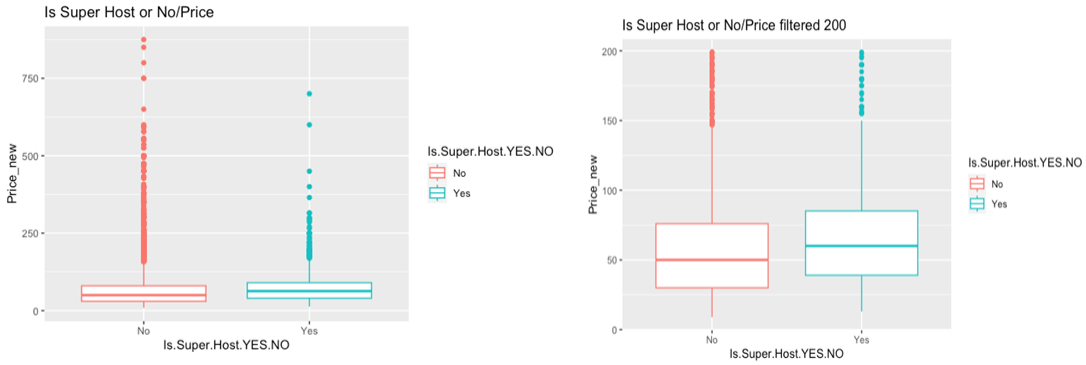
Gráfico de cuartiles y la mediana variable: “Price”. Este gráfico muestra que en los datos se encuentran muchos valores alejados de la mediana. Al tratarse de la variable precio en la fase de limpieza se revisa la existencia de datos erróneos pero se mantienen los valores extremos por su relevancia. En el Boxplot de la izquierda se muestran los datos sin filtrar y a la derecha con filtro >120.



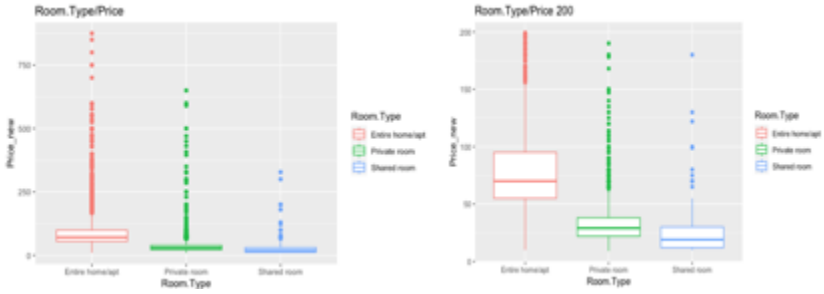
Este mismo caso también se observa en la variable “Number.of.Reviews”. También se muestran los gráficos para los valores de la variable filtrando por >70 para ver mejor la distribución de los datos y la mediana. Esta misma función se aplican a otras variables como “Bedrooms_new”.

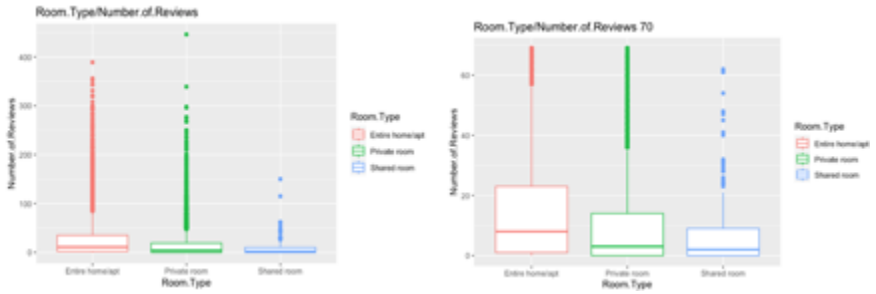


Posteriormente se realizaron Boxplot comparativos que muestran cuartiles y la mediana para la variable “Is.Super.Host.YES.NO” según precio. Se aprecia la mediana ligeramente más elevada en el caso de los Super Host.

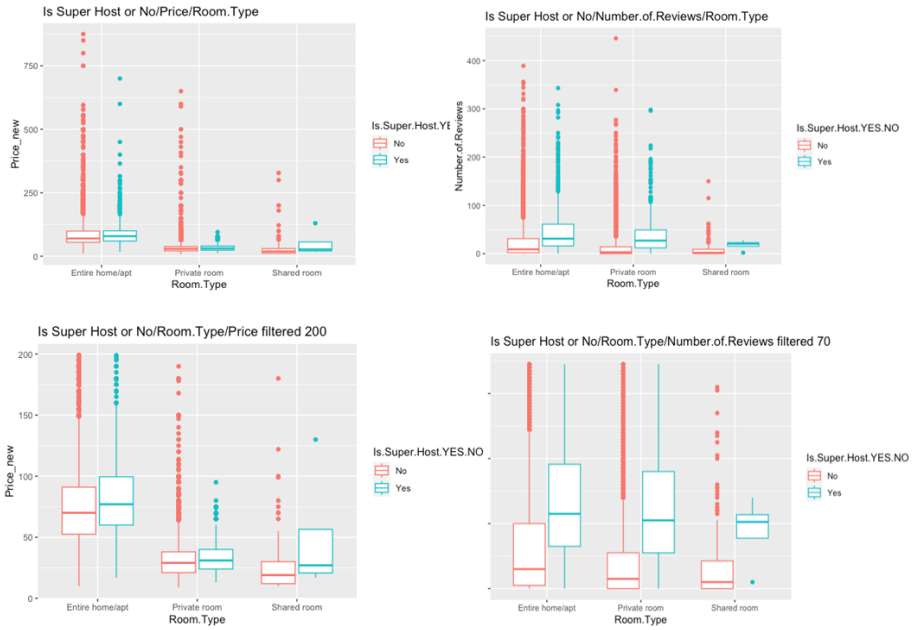


También se revisan el precio y número de reviews según tipo de habitación. Se observa que el tipo de habitación con una mediana de precio más elevada es “Entire home”, seguido de “Private Room” y “Shared Room”. Este mismo patrón también se repite para la variable “Number.of.Reviews”.





También se realizan gráficos Boxplot para comprobar cómo se comportan los datos para la variable de “Host” y “Super Host” respecto a la mediana de precio y número de reviews por tipo de habitación. Se observa la misma tendencia ligeramente superiores en “Super Host”.



2.3. Matrices de correlación.

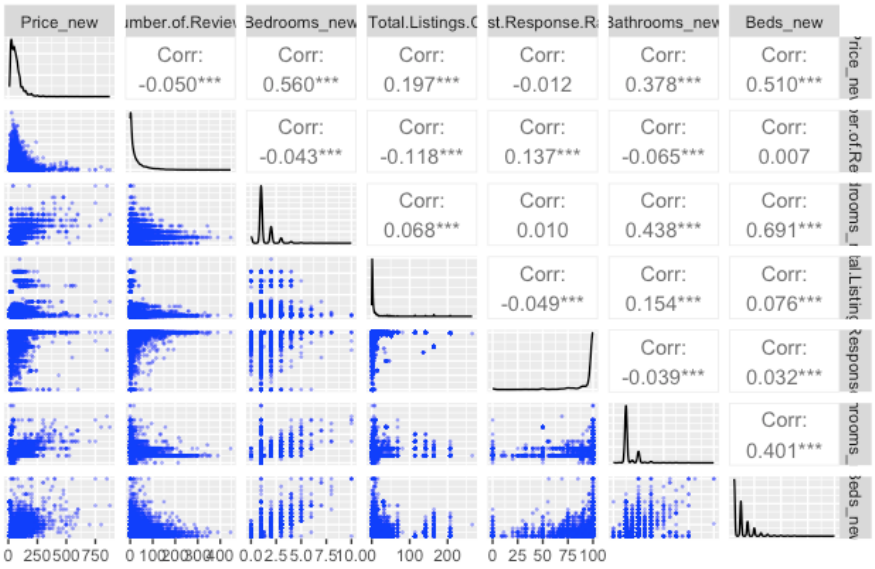
Se generan dos gráficos para la visualización de las matrices de correlación. La correlación refleja la covarianza entre dos variables y grado de dispersión entre los datos en un rango de (-1/1). Para ello aplicamos la función cor() para las variables numéricas de nuestro dataset y visualizamos estos resultados en dos gráficos de "ggpairs". En la matriz se observan valores cercanos a 1 para las variables: "Bedrooms_new"(0,560), "Beds_new"(0,510), "Bathrooms_new"(0,378), por otra parte presentan valores cercanos a 0 en "Host.Total.Listings.Count"(0,197) y como no significativas encontramos "Number.of.Review" (-0,050) y "Host. Response.Rate" (-0,012) y con un posible indicio de regresión a la inversa con el precio.

En la segunda matriz encontramos como resultados valores cercanos a 1 para las variables "Price_new_weekly" (0,983) y "Price_new_monthly" (0,945) pero estas variables se definen de forma dependiente entre sí. Otras variables con valores cercanos a 1 son "Cleaning.Fee"(0,620), "Accommodates"(0,638), "Security.Deposit"(0,413) y valores cercanos a 0 se encuentran las variables "Review.Scores.Location"(0,128), "Review.Scores.Rating"(0,068) y "Review.Scores.Value"(0,037).

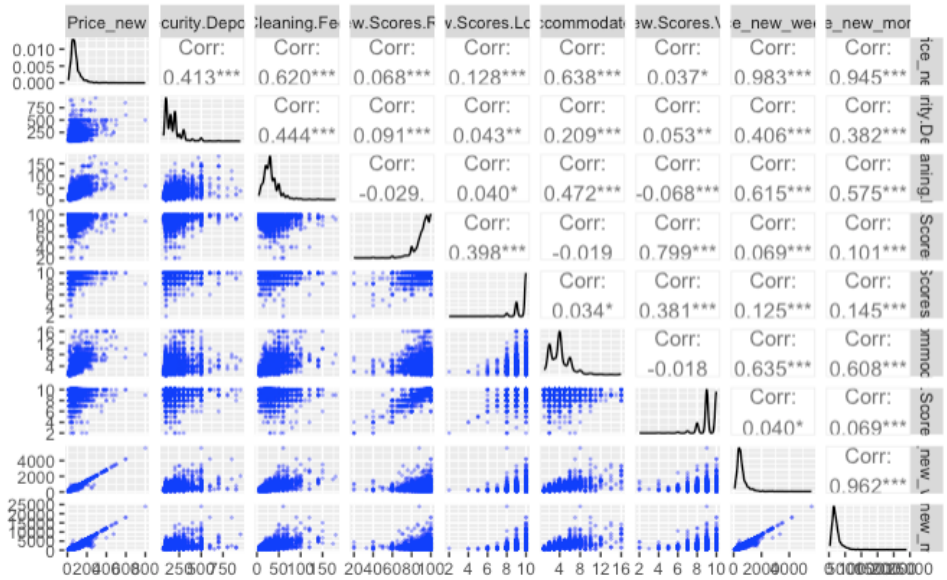
De acuerdo con estos datos las variables más adecuadas para generar el modelo predictivo del precio serían "Cleaning.Fee"(0,620), "Accommodates"(0,638), "Security.Deposit"(0,413), "Bedrooms_new"(0,560), "Beds_new"(0,510) y "Bathrooms_new"(0,378), también "Host.Total.Listings.Count"(0,197) y "Review.Scores.Location"(0,128) representan un porcentaje representativo. Se podrían valorar el resto de variables con valores significativos de correlación si mejoran el modelo predictivo, para ello estas variables serán también analizadas para comprobar su grado de significación en el estudio.

Los gráficos que nos permiten ver las relaciones entre variables, identificar posibles tendencias o regresiones lineales y comprobar el indicador de correlación para identificar las variables más interesantes.

Matriz de correlación 1.



Matriz de correlación 2.

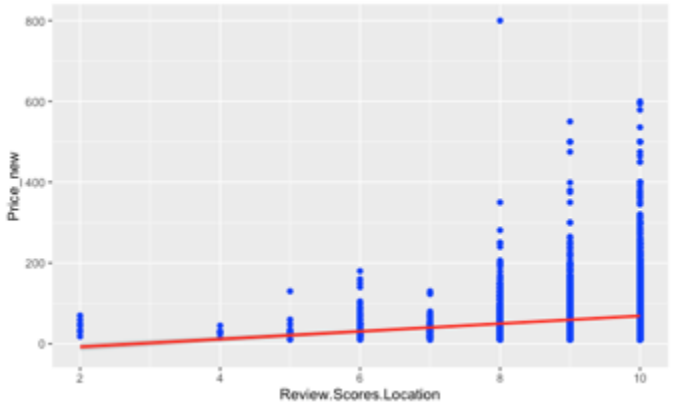


2.4. Regresiones lineales.

Se ha realizado una regresión lineal con cada una de las variables que han tenido una correlación significativa con la variable precio. Estas regresiones lineales nos permiten comprobar si existe una relación entre ambas y cómo es la tendencia de la relación.

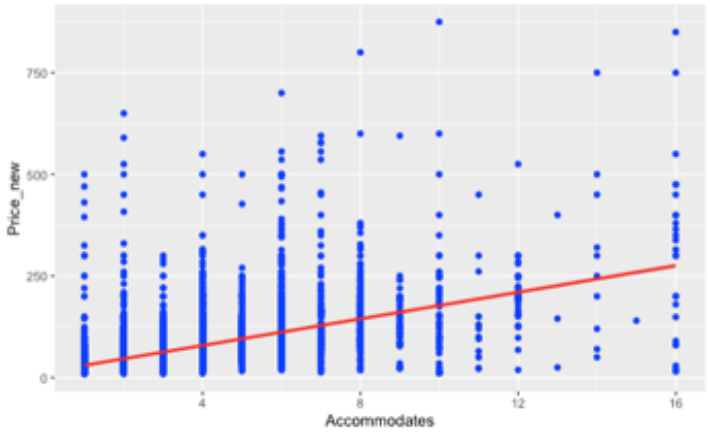
Para ello se ha realizado un diagrama de puntos o “scatterplot” para cada variable que utiliza el método “lm” y una función para calcular el precio en función de la variable escogida. Para cada regresión se genera un gráfico, se aplica una función de correlación cor(x,y) que indica el grado de variación conjunta y lineal de dos variables, y se construye un modelo con la función lm(dataframe, formula=Precio~variable) y también la función predict(modelo, dataframe) sobre un conjunto de datos para observar las predicciones que realizan los modelos. Además aplicamos la función summary() a cada modelo para comprobar el R^2 y los pvalor (Pr).

1. REGRESIÓN LINEAL PRECIO SEGÚN REVIEW.SCORES.LOCATION



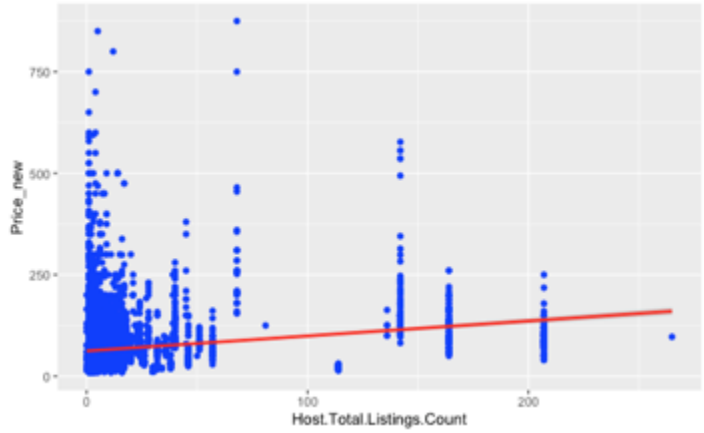
El gráfico muestra una función no muy predictiva pero se marca una tendencia positiva. Una correlación de 0.1491001 y un R^2 de 0.02223. El pvalor (Pr) marca valores bajos y significativos (***).

2. REGRESIÓN LINEAL PRECIO SEGÚN ACCOMMODATES



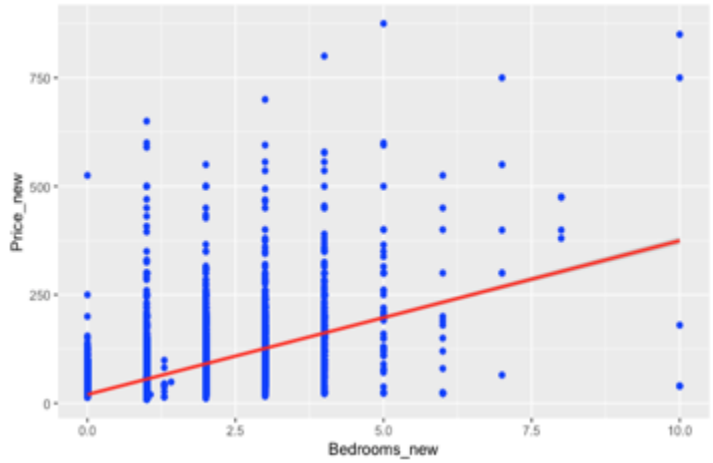
El gráfico muestra una función una regresión positiva. Una correlación de 0.5850059 y un R^2 de 0.3422. El pvalor (Pr) marca valores bajos y significativos (***).

3. REGRESIÓN LINEAL PRECIO SEGÚN HOST TOTAL LISTINGS



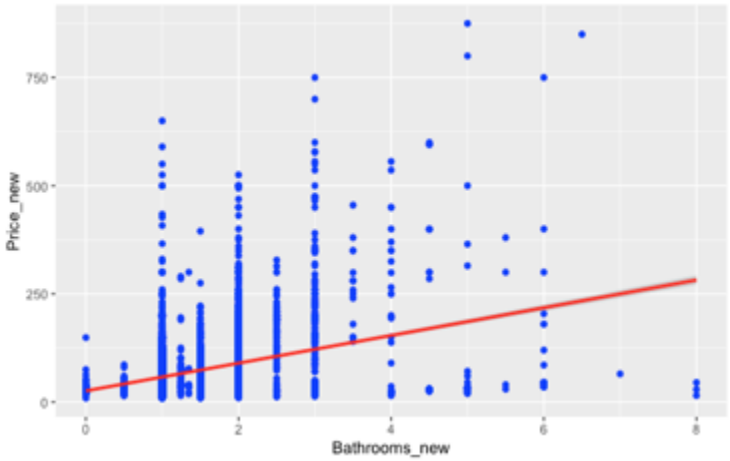
El gráfico muestra una función no muy predictiva pero se marca una tendencia positiva. Una correlación de 0.1831523 y un R^2 de 0.03354. El pvalor (Pr) marca valores bajos y significativos (***).

4. REGRESIÓN LINEAL PRECIO SEGÚN BEDROOMS_NEW



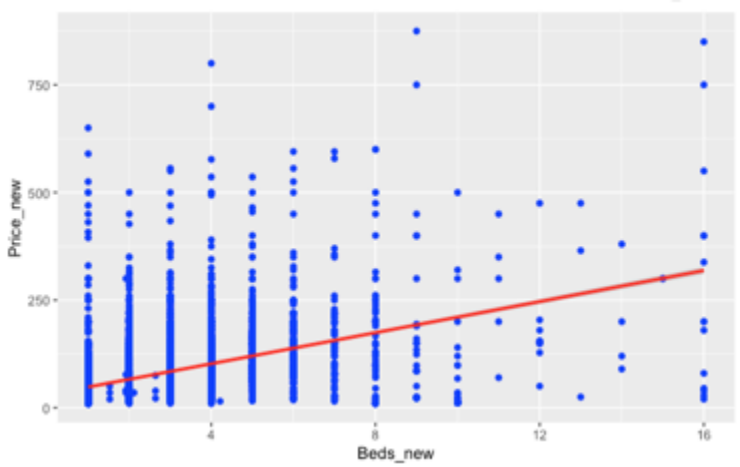
El gráfico muestra una función que marca una regresión positiva. Una correlación de 0.5266968 y un R^2 de 0.2774. El pvalor (Pr) marca valores bajos y significativos (***).

5. REGRESIÓN LINEAL PRECIO SEGÚN BEATHROOMS_NEW



El gráfico muestra una función que marca una regresión positiva. Una correlación de 0.3442813 y un R^2 de 0.21185. El pvalor (Pr) marca valores bajos y significativos (***).

6. REGRESIÓN LINEAL SEGÚN BEDS_NEW

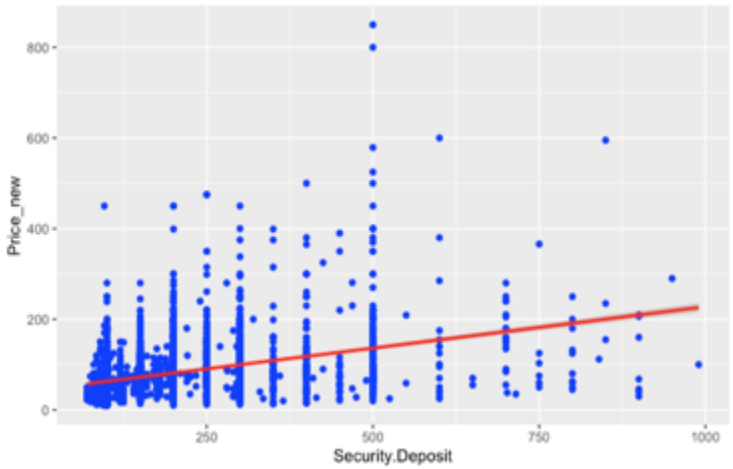


El gráfico muestra una función que marca una regresión positiva. Una correlación de 0.4889465 y un R^2 de 0.2391. El pvalor (Pr) marca valores bajos y significativos (***).

7. REGRESIÓN LINEAL SEGÚN CLEANING.FEE

El gráfico muestra una función que marca una regresión positiva. Una correlación de 0.6526313 y un R^2 de 0.4259. El pvalor (Pr) marca valores bajos y significativos (***).

8. REGRESIÓN LINEAL SEGÚN SECURITY DEPOSIT



El gráfico muestra una función que marca una regresión positiva. Una correlación de 0.367005 y un R^2 de 0.1347. El pvalor (Pr) marca valores bajos y significativos (***).

Esta fase nos ha servido para comprobar la calidad de las relaciones entre las variables y determinar las variables más predictivas respecto al precio. Estas son las que utilizamos para realizar la regresión lineal múltiple y un modelo de predicción del precio. Tras el análisis y la aplicación de estas funciones se observa que aquellas variables relacionadas con las “reviews” y que aparentemente tenían una relación de peso en el estudio no muestran relaciones evidentes aunque sí son significativas en relación con el precio.

Por otra parte, las variables “Number.of.Reviews” , “Host.Response.Rate”, “Review.Scores.Rating”, “Review.Scores.Value” y “Calculated.host.listings.count” que muestran valores de R^2 y correlación muy bajas pero cuyos pvalor (Pr) se marcan como significativos, se guardan como variables a utilizar en caso de que puedan ser útiles para mejorar el modelo predictivo ya que tienen cierto grado de relación con el precio.

2.5. Regresiones lineales múltiples.

Se realiza un modelo lineal múltiple utilizando las variables que han demostrado una mayor relación con el precio y más significativas en su correlación. Además incluimos las variables que minimicen el error cuadrático medio, es decir, la diferencia entre lo predicho y lo real.

Para ello, se realiza un modelo de regresión lineal mediante la función “lm” y con fórmula de Precio en relación con las siguientes variables: “Review.Scores.Location”, “Accommodates”, “Host.Total.Listings.Count”, “Bedrooms_new”, “Bathrooms_new”, “Beds_new”, “Security.Deposit” y “Cleaning.Fee”.

```
Call:
lm(formula = Price_new ~ Review.Scores.Location + Accommodates +
    Host.Total.Listings.Count + Bedrooms_new + Bathrooms_new +
    Beds_new + Security.Deposit + Cleaning.Fee, data = df_multipleregresion)

Residuals:
    Min      1Q  Median      3Q     Max
-164.10  -15.38   -1.18   12.77  552.26

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)              -94.635561   7.821695 -12.099  < 2e-16 ***
Review.Scores.Location     7.376812   0.808497   9.124  < 2e-16 ***
Accommodates               8.283121   0.526757  15.725  < 2e-16 ***
Host.Total.Listings.Count  0.045703   0.018444   2.478   0.0133 *
Bedrooms_new               5.951063   0.961700   6.188 6.75e-10 ***
Bathrooms_new             24.581939   1.273407  19.304  < 2e-16 ***
Beds_new                  -3.236074   0.653030  -4.955 7.53e-07 ***
Security.Deposit           0.071121   0.006022  11.809  < 2e-16 ***
Cleaning.Fee               0.787023   0.037584  20.940  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.74 on 3745 degrees of freedom
  (9014 observations deleted due to missingness)
Multiple R-squared:  0.6242,    Adjusted R-squared:  0.6234
F-statistic: 777.4 on 8 and 3745 DF,  p-value: < 2.2e-16
```

El modelo resultante indica un R^2 de 0.6242. Este por tanto podría mejorarse mediante la modificación de algunas de las variables incluidas en él. Para ello se realizan distintas comprobaciones para mejorar la calidad del modelo mediante la

exclusión e inclusión de variables significativas que han sido previamente analizadas mediante la matriz de correlación y las regresiones lineales, las funciones de correlación y la comprobación de la calidad de los modelos, tomando como indicadores de referencia el R^2 y pvalor (Pr). También aplicamos la función confint() para comprobar los intervalos de confianza del modelo e implementamos la función predict para dos casos de datos según las variables escogidas para el modelo.

```{r}
predict(modelmultiple, data.frame(Review.Scores.Location=c(8, 10), Accommodates=c(4, 2), Host
.Total.Listings.Count=c(3, 5), Bedrooms_new=c(2, 1), Bathrooms_new=c(2, 1), Beds_new=c(4,1),
Security.Deposit=c(150, 80), Cleaning.Fee=c(60, 20)))
```

```
        1         2
103.65972  44.65436
```

Para evaluar la calidad del modelo calculamos las figuras de calidad dividiendo nuestros datos el conjunto de datos en train y test y mediante la librería “caret” de ML a la que le pasamos los datos predichos y los reales que nos calcula el RMSE, Rsquared y MAE. Es un buen indicativo si el Rsquared es similar en ambos resultados y nos sirve para comparar cómo se comporta el modelo en ambos casos.

```{r}
set.seed(1234)
idx <- sample(1:nrow(df_multipleregresion2), nrow(df_multipleregresion2)*0.7)
train.df_multipleregresion2 <- df_multipleregresion2[idx,]
test.df_multipleregresion2 <- df_multipleregresion2[-idx,]
#para comprobar nrow(train.df_multipleregresion2), nrow(test.df_multipleregresion2)
```

Se comprueba, tras la implementación de distintos modelos con combinaciones de variables diversas (todas indicadas en los análisis de las matrices de correlación y las regresiones lineales) que mediante la inclusión de las variables “Security.Deposit” y “Cleaning.Fee” el error cuadrático medio se minimiza. Esto indica que se debería valorar si son variables correctas para realizar las predicciones de precio ya que ambas variables cuentan con valores bajos en la

base de datos con una pérdida del 50% de los valores. Si estas se excluyen el R^2 se sitúa por debajo del 0,50 por lo que el modelo no mejora.

Se debería considerar mejorar la calidad de las variables relacionadas con "Reviews" pues se muestran como significativas pero también con una pérdida de datos relevante y podrían llegar a mejorar el modelo en el futuro.

```
Call:
lm(formula = Price_new ~ Review.Scores.Location + Accommodates +
    Bedrooms_new + Beds_new + Bathrooms_new + Bathrooms_new +
    Review.Scores.Rating + Cleaning.Fee, data = df_multipleregresion2)

Residuals:
    Min      1Q  Median      3Q     Max
-184.33  -14.19   -1.79   11.97  586.58

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           -86.61138    5.65657 -15.312  < 2e-16 ***
Review.Scores.Location   4.89788    0.60632   8.078 7.82e-16 ***
Accommodates             8.72456    0.37983  22.969  < 2e-16 ***
Bedrooms_new             7.34949    0.70948  10.359  < 2e-16 ***
Beds_new                -2.85030    0.47843  -5.958 2.70e-09 ***
Bathrooms_new           17.86807    0.87995  20.306  < 2e-16 ***
Review.Scores.Rating     0.28366    0.04983   5.693 1.31e-08 ***
Cleaning.Fee             0.97925    0.02089  46.887  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 30.79 on 6309 degrees of freedom
  (6451 observations deleted due to missingness)
Multiple R-squared:  0.634,     Adjusted R-squared:  0.6336
F-statistic:  1562 on 7 and 6309 DF,  p-value: < 2.2e-16
```

Finalmente se logra realizar una pequeña mejora de calidad del modelo eliminando la variable "Security.Deposit" con R^2 0.634, pero debe valorarse si es óptimo eliminarla ya que no mejora el modelo. Por su parte, las variables "Number.of.Review" y "Host. Response.Rate" presentan posibles indicios de regresión a la inversa y podrían ser significativas con una mayor calidad o muestra de los datos, que nos permitieran comprobar más tendencias y patrones en este tipo de variables.

Fase 3. Visualización de las métricas

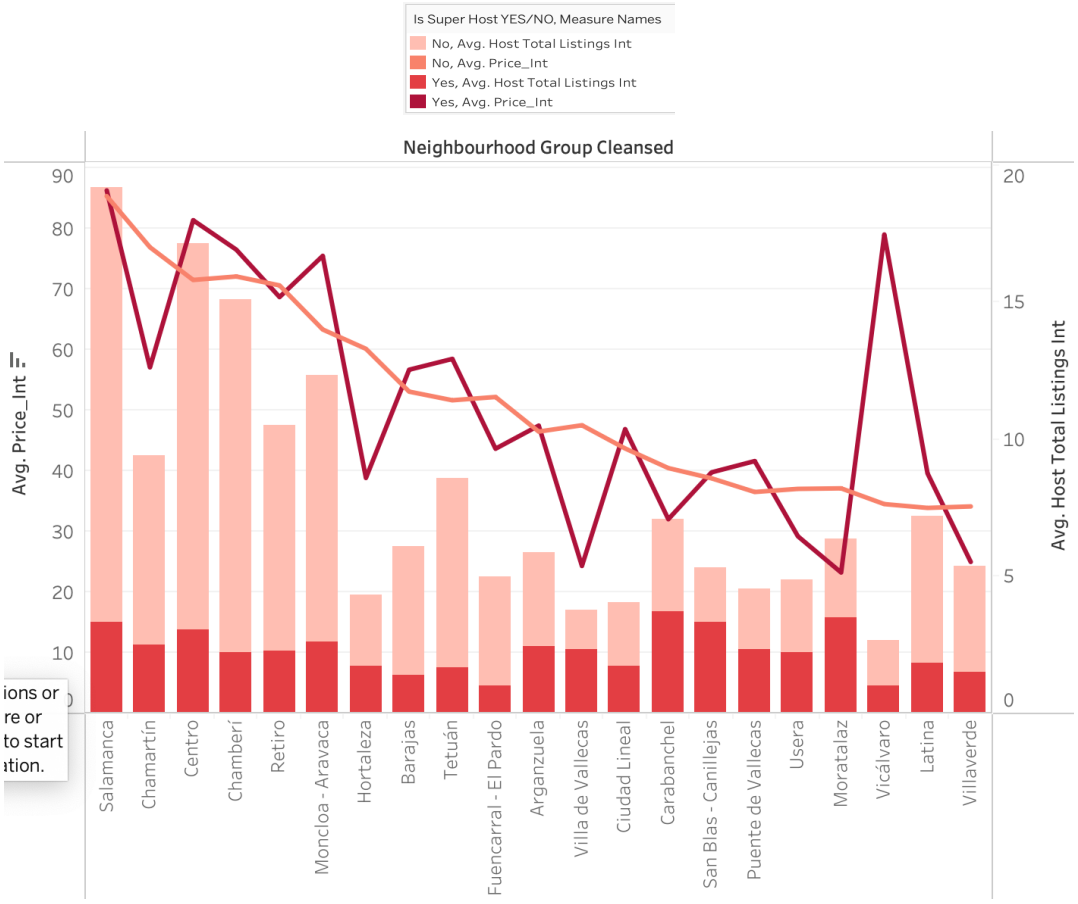
Para la fase de visualización de métricas y exploración/visualización de datos conectamos el dataset final del proyecto con Tableau y preparamos el campo calculado.

Campo Calculado por Reviews Scoring:

AVG([Review Scores Location Int])+AVG([Review Scores Value Int])

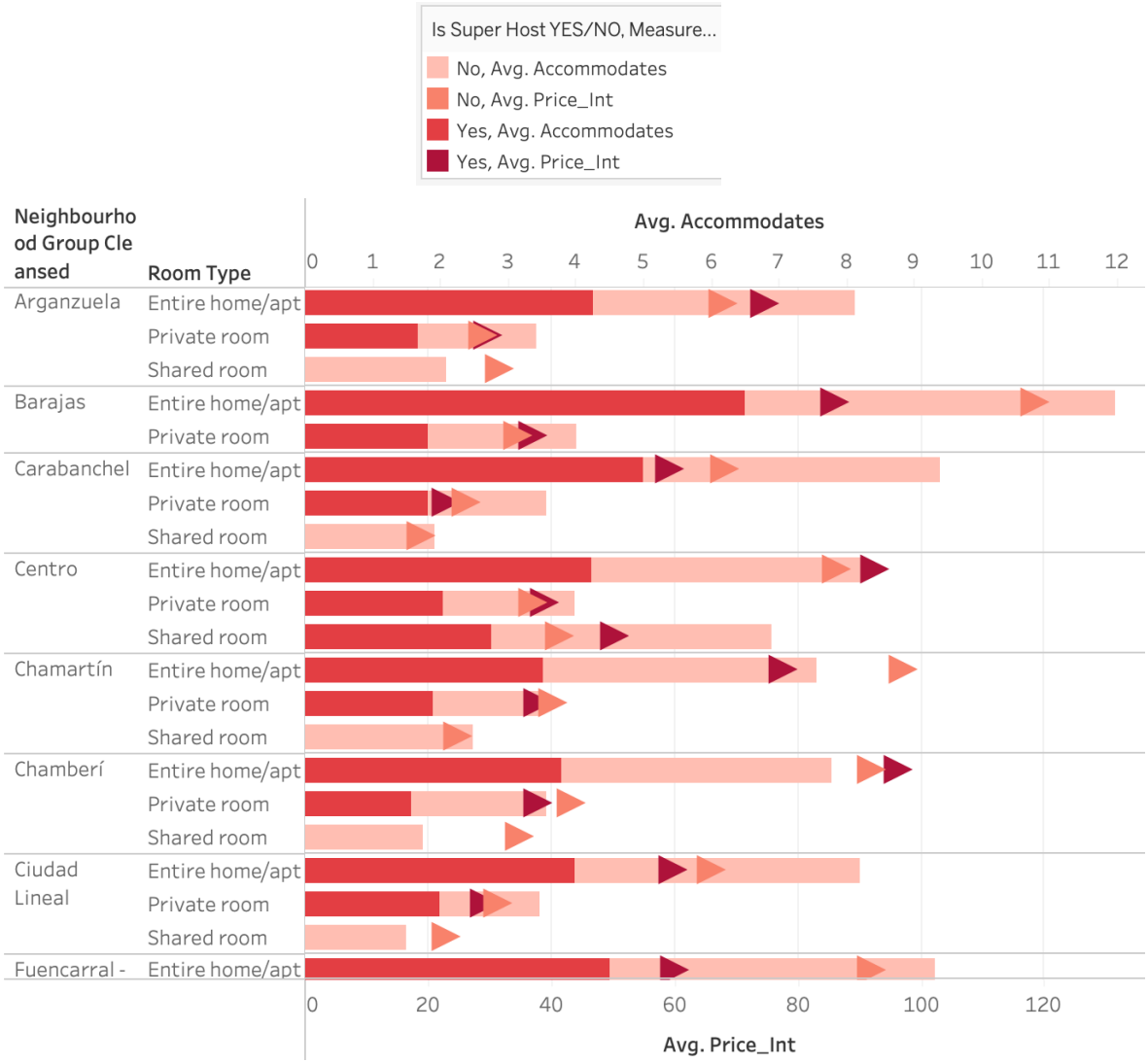
1. Gráfico: **Price/Host Total Listings Neighbourhood** donde comparamos el promedio de precio y el promedio total listings con la diferencia entre hosts y super hosts por cada barrio.

Con este gráfico podemos ver que los precios de hosts normales tienen una inclinación casi normal. Cuantos más hosts hay, mayor es el precio. Pero con los super hosts vemos precios diferentes, ya sea más altos o más bajos que el promedio, eso depende del precio que cada superhost establece para su Airbnb.



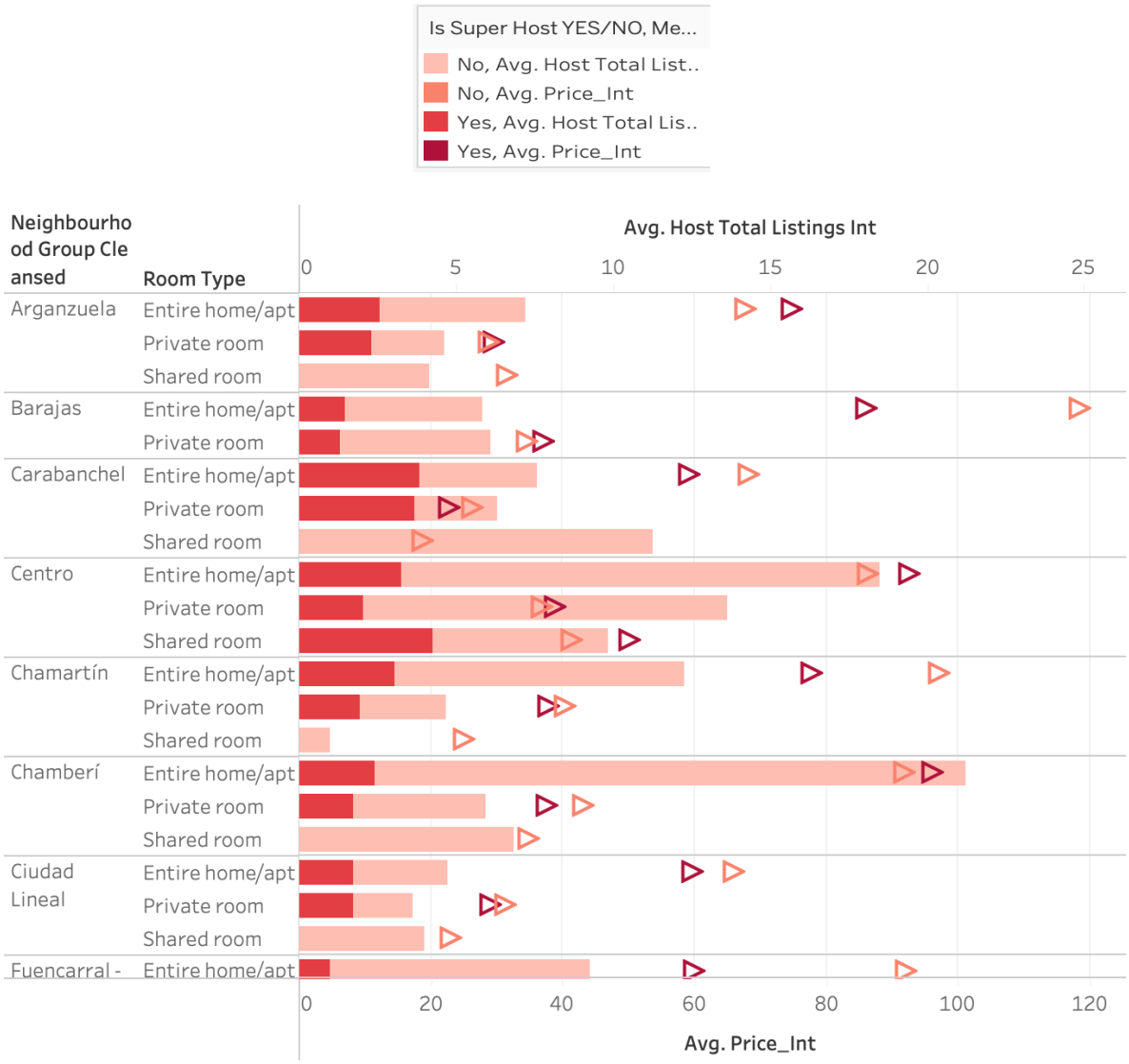
2. Gráfico: **Price/Accomodate Neighbourhood,** donde comparamos cada barrio con los diferentes “Room Types” y cuantos invitados pueden alojarse, con el promedio de precio entre hosts y superhosts.

Aquí podemos ver que las “Entire Homes” son las preferidas y con los precios medios más altos en comparación con el resto de tipos de habitaciones, ya que pueden alojarse más huéspedes.



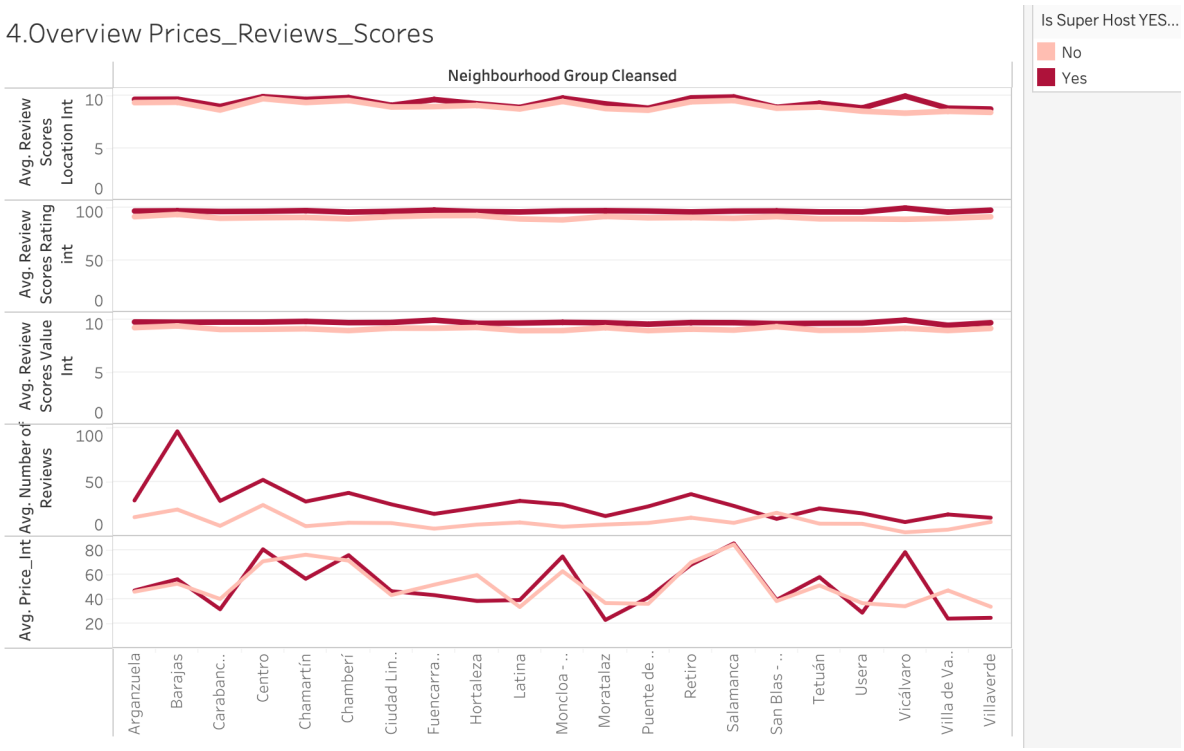
3. Gráfico: **Price/Listings Neighbourhood** donde comparamos el precio promedio por barrio y los listados totales de alojamientos, ya sea Super Host o Host con los “Room Types” específicos.

Aquí podemos ver que las “Entire Homes” son un poco más caras en comparación con las habitaciones privadas o compartidas.



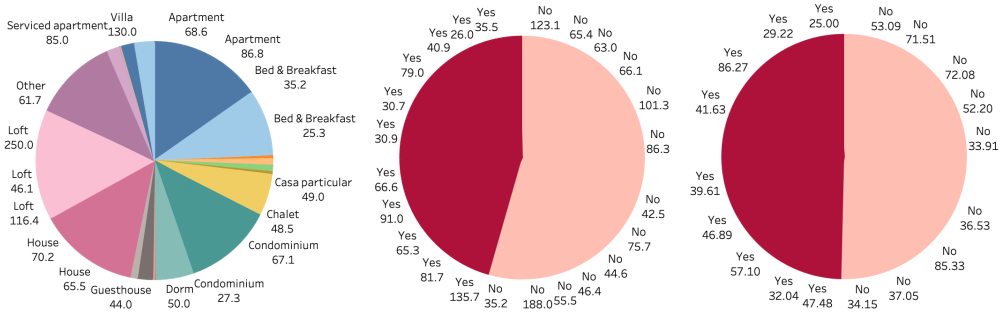
4. Gráfico: **Overview Prices_Reviews_Scores** donde comparamos Host y Super Host en función de su precio promedio, el número promedio de reseñas que recibieron y las puntuaciones promedio de sus reseñas sobre la ubicación y la calificación.

Aparte del precio como mencionamos antes que en algunos barrios tienen precios diferentes ya sea más bajos o más altos que los hosts normales, en las 3 otras categorías vemos una tendencia similar entre los 2 hosts.



5. Gráfico: **Price Correlation**, comparamos el precio promedio para cada una de las 4 categorías: “Avg. Bedrooms_New”, “Avg. Bathrooms_New”, “Avg. Beds_New” y “Avg. Accommodates” tras comprobar en el análisis de R que influyeron más en el precio.

6. A continuación visualizamos **3 Pie Charts** diferentes para utilizar en las descripciones de nuestros gráficos. Creamos un Pie Chart para los diferentes “Property Types” por código postal y su precio promedio. Un gráfico entre Hosts y Super Hosts por código postal y uno por barrio.

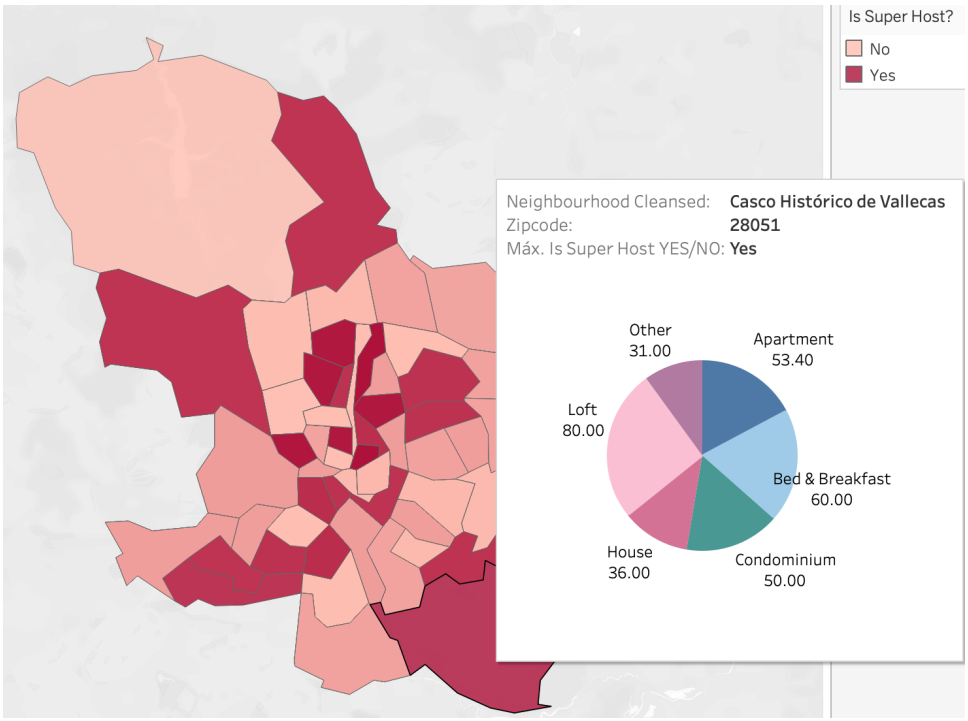


7. También creamos una **nube de palabras** con la correlación del precio promedio por barrio. Cuanto más grande el tamaño de la letra, el barrio presenta un promedio de precio más elevado, por ejemplo, se puede observar que El Plantío es el barrio más caro.

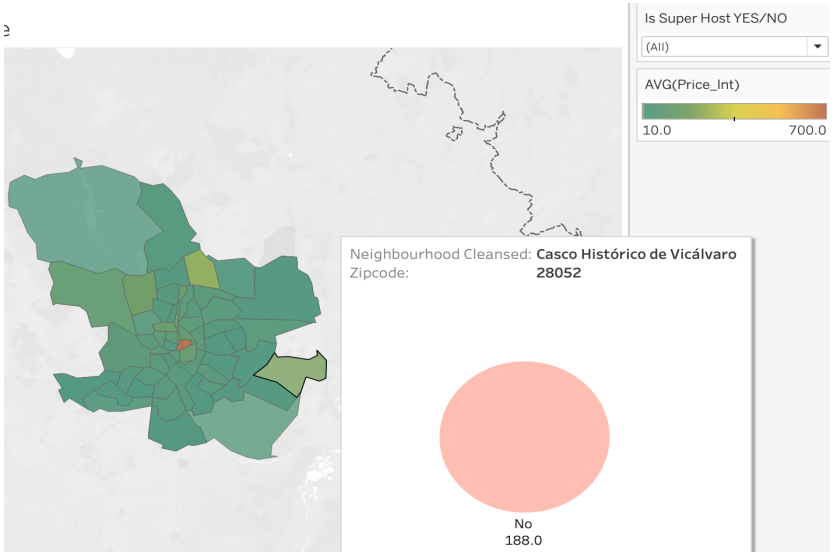
8. **Mapa de Madrid** que está codificado por colores por Hosts y Super Hosts y cuando se desplaza el cursor sobre el barrio que desea también puede ver el tipo de propiedades y su precio promedio.

KEEPCODING Tech School

Glovo

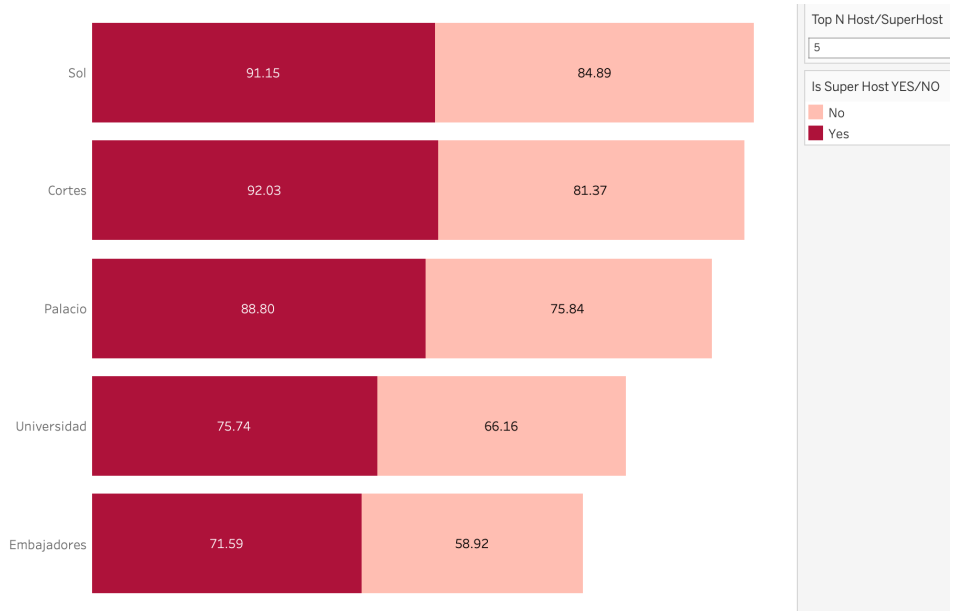


9. **Mapa Price** que muestra los precios por barrio en promedio, los colores van desde los más económicos (verde) hasta los más caros (amarillo). También se incluye un filtro donde muestra los barrios con Super hosts o Hosts normales.

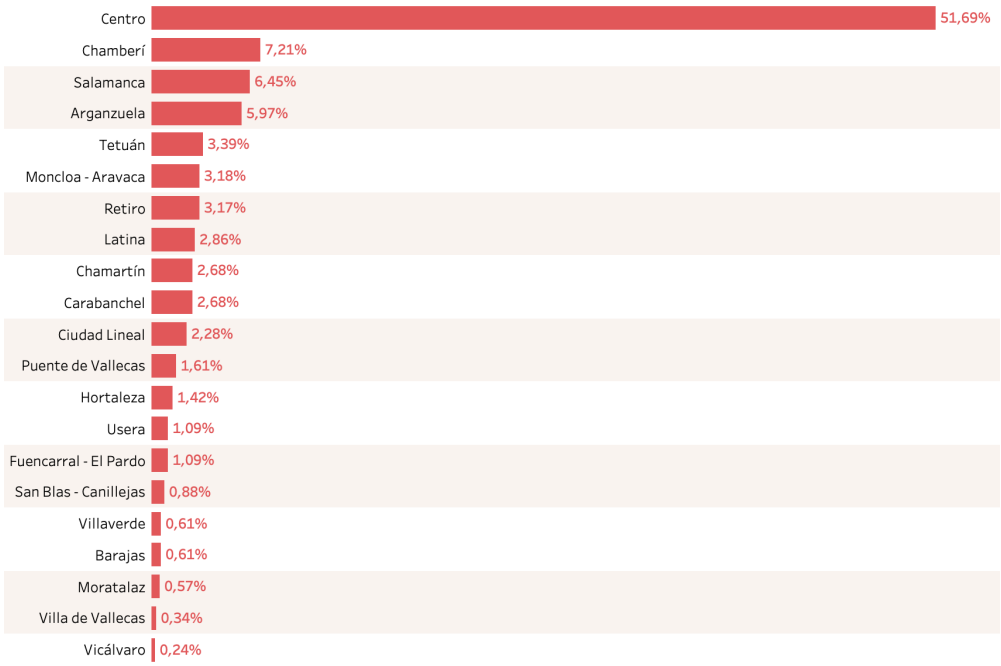


10. Filtro **Top N** de barrios que puede filtrar para ver el ránking que tienen el precio promedio más alto entre Hosts y Super hosts. Por ejemplo, en el Top 5 vemos que

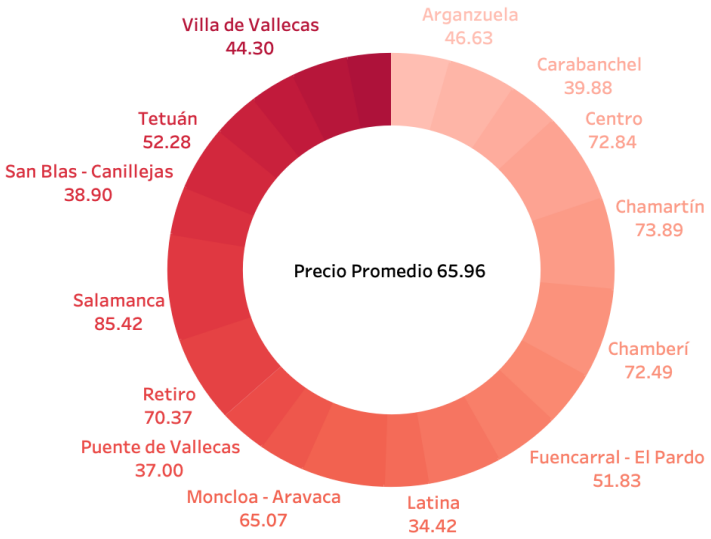
el barrio Sol es el más caro, con un precio promedio de 91 euros por Super Hosts y 85 euros para Hosts.



11. **Gráfico de porcentajes**: Se muestra el % de alquileres que tiene cada barrio. Por ejemplo podemos ver que el Centro tiene la mitad de los alquileres de Airbnb en Madrid con un 51,69%.



12. **Pie Chart** que ordena alfabéticamente todos los barrios y sus precios medios. Además, si desplazas el cursor sobre el gráfico puede verse el precio promedio en ese barrio entre un Host y un Super Host.



13. Agregamos un **conteo** simple de todos los Hosts y Super Hosts, un conteo de los alojamientos, el precio medio diario y el resumen de todo el número de reviews como datos generales para nuestro dashboard.

Alojamientos	Sum Reseñas
12.768	300.508
Precio Promedio Diario	**Super Hosts**
65.96	No 11.245
	Yes 1.523

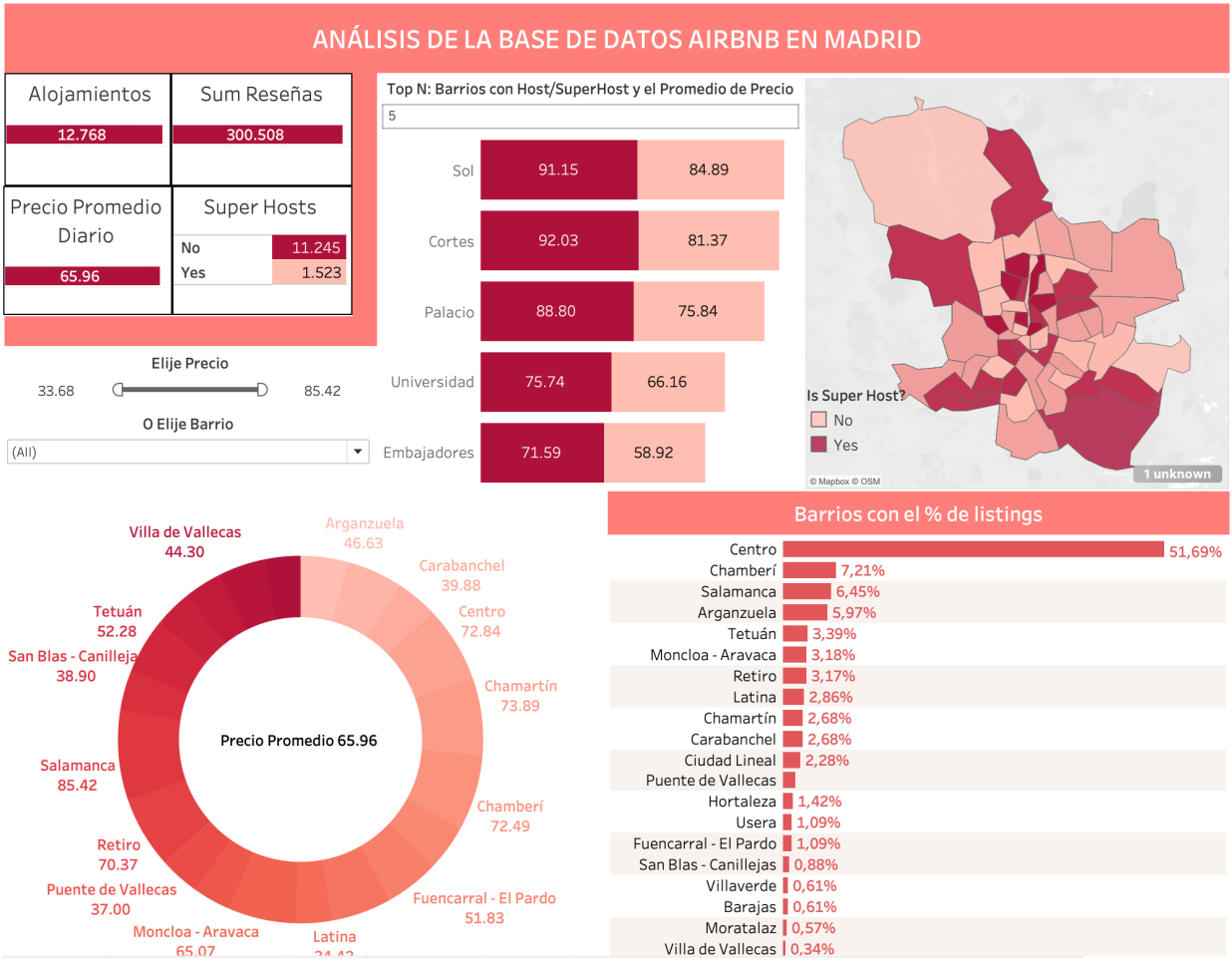
14. Finalmente creamos nuestro **dashboard** donde en la esquina superior izquierda tenemos los 4 cuadros de los datos generales. Podemos ver que hay 12.768 alquileres en nuestra base de datos. Todos tienen una suma de 300.508 reseñas. El precio medio en Madrid por noche es de 65,96 euros y los Hosts son 11.245 mientras que los Super Hosts son 1.523.

En el gráfico TOP N. En este se puede filtrar por los Top Barrios y ver un promedio de precio entre el Host y Super hosts. Por ejemplo en Sol se puede observar que los Super Hosts tienen un precio medio de 91,15 euros mientras que los Host tienen 84,89 euros.

A continuación también tenemos nuestro mapa interactivo, donde puedes elegir directamente la ubicación que deseas y muestra la información en caso de que tengan más Hosts o Super Hosts y ver los precios promedio por tipo de habitación. Por ejemplo en Aravaca tenemos más Super Hosts y en este barrio el precio promedio de alquilar de una Camper/RV es de 125 euros al día mientras que alquilar una casa es de 32 euros.

En la parte inferior de nuestro mapa agregamos la gráfica que muestra todos los barrios y el % de alquileres que tiene cada uno. Se muestra que la mitad de los anuncios de nuestra base de datos se encuentran en el centro de Madrid.

Y finalmente, como último gráfico, tenemos nuestro Pie Chart en el que el usuario puede ver todos los barrios alfabéticamente y codificados por colores, desde rojo claro hasta rojo más oscuro. En el centro de la gráfica se muestra el precio promedio considerando todos los barrios, y en la parte externa muestra alguno de los barrios con el precio promedio de cada uno de ellos. También se pueden utilizar los filtros de la parte superior para filtrar por el rango de precios que se desee para que pueda encontrar los barrios que son más baratos para reservar o filtrar directamente por barrios y ver el listado de precios.



Fase 4.- Conclusiones y “lessons learned”

Durante el proceso de limpieza y normalización de datos, nos centramos en diferentes KPI específicos para poder llevar adelante el modelado y visualización de las métricas finales.
La presencia de valores NaN nos llevó a aplicar algunas opciones para dar más uniformidad a los valores de las distintas columnas.

Se ha llevado a cabo un procesos de limpieza y comprobación de columnas, por ejemplo la del "Zipcode", y creaciones de nuevas columnas, por ejemplo "Price new", “ Price new weekly”, “is Super Host” etc. Todo esto con el fin de poder crear un conjunto de datos final y efectivo para los siguientes pasos de análisis del proyecto.

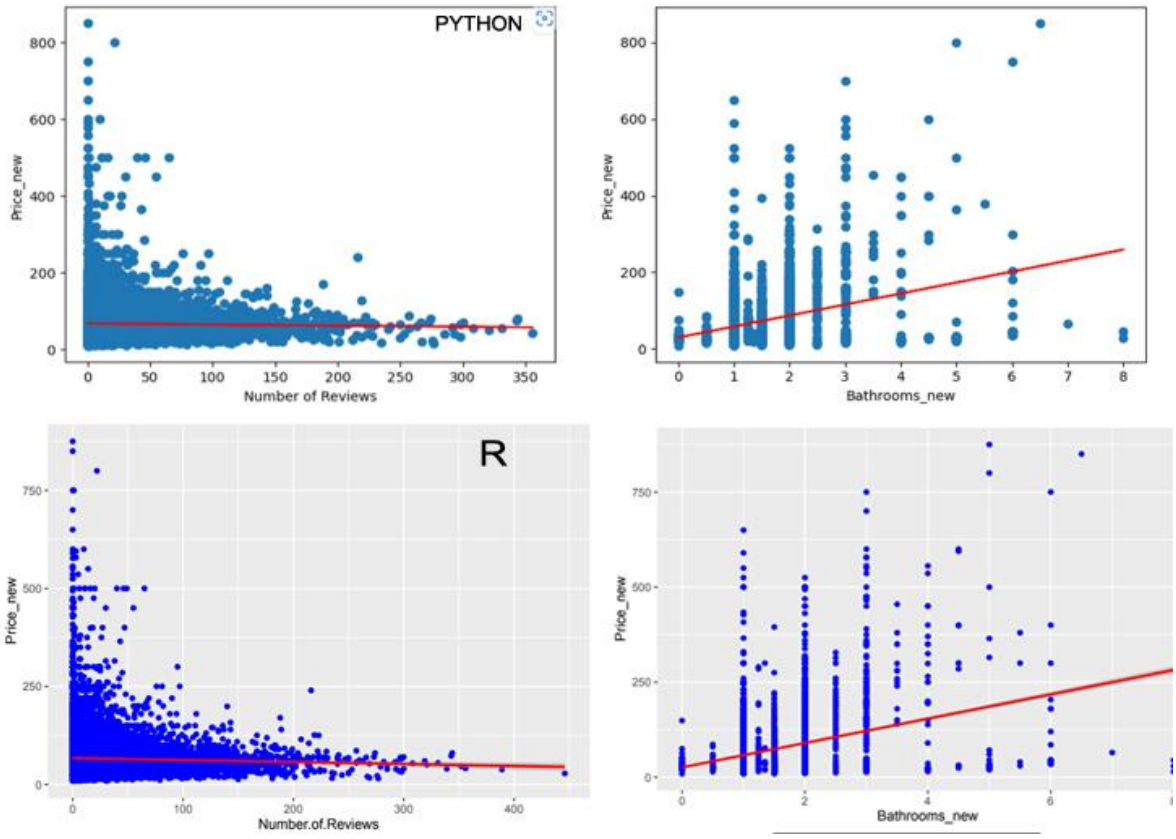
```
df_madridfinal.columns
```

```
Index(['ID', 'Listing Url', 'Name', 'Description', 'Host ID', 'Host Since',
       'Host Response Rate', 'Host Total Listings Count', 'Host Verifications',
       'Street', 'Neighbourhood Cleansed', 'Neighbourhood Group Cleansed',
       'City', 'State', 'Zipcode', 'Market', 'Smart Location', 'Country Code',
       'Country', 'Latitude', 'Longitude', 'Property Type', 'Room Type',
       'Accommodates', 'Bed Type', 'Amenities', 'Security Deposit',
       'Cleaning Fee', 'Number of Reviews', 'Review Scores Rating',
       'Review Scores Location', 'Review Scores Value',
       'Calculated host listings count', 'Geolocation', 'Price_new',
       'Price_new_weekly', 'Price_new_monthly', 'Beds_new', 'Bedrooms_new',
       'Bathrooms_new', 'Is Super Host', 'Is Super Host YES/NO'],
      dtype='object')
```

En el proceso de Análisis Exploratorio y modelado de los datos, se realizaron diferentes pruebas de regresión lineal con variables numéricas tanto con R como con Python Sklearn. Se ha comprobado si existían relaciones de correlación entre las variables analizadas (matriz de correlación) para evaluar las predicciones de precios. Las variables con una mayor relación con el precio y más significativas en su correlación fueron utilizadas para realizar el modelo de regresión lineal múltiple.

Se debería considerar mejorar la calidad de las variables relacionadas porque también si se muestran como significativas, algunas tienen una pérdida de datos relevante que se podría llegar a mejorar cualitativamente y así mejorar el modelo.

Mostramos un gráfico con un ejemplo de regresión lineal simple realizado sobre las mismas columnas: Price_new, Number of Reviews, Bathrooms new, tanto en R como en Python.



Durante el proceso de visualización de datos y creación del dashboard con Tableau se han mostrado algunas informaciones relevantes: las “Entire Homes” son las preferidas y con los precios medios más altos en comparación con las habitaciones privadas o compartidas. En tema Super Host / Host el barrio Sol es el más caro, con un precio promedio de 91 euros por Super Hosts y 85 euros para Hosts. La zona del Centro tiene la mitad de los alquileres de Airbnb en Madrid con un 51,69%.

Consideramos las métricas finales teóricamente correctas, sin embargo, desde el punto de vista de la presencia de valores NaN y consiguiente falta de datos, los

cálculos podrían mejorarse con un tratamiento más centrado en el manejo de los valores nulos y en la mejora de la calidad del dataset.

La creación de este proyecto nos permitió aplicar los conceptos teóricos y prácticos aprendidos durante el curso a un contexto de datos real.

Aprendimos la importancia de la evaluación cualitativa y cuantitativa de los datos y el efecto que tienen los datos nulos durante el proceso analítico para la realización de las predicciones estadísticas.

Las fases de limpieza de los datos y la imputación de los nulos han sido muy importantes para mejorar la calidad de las estimaciones y la aplicación de funciones y obtención de resultados más óptimos.

La exploración y visualización de datos en R ha sido importante para ver en detalle los indicadores y ver cómo se comportan los datos.

Para modificar el guión del proyecto y hacer algunos cambios habríamos probado otras metodologías para rellenar los valores NaN con métodos diferentes. Habríamos realizado más pruebas de análisis para obtener nueva información y comprender mejor los datos, así como habríamos consultado también de otras fuentes Airbnb, una base de datos más amplia y de otras zonas y ciudades para comprobar patrones o tendencias en los datos.