Tecnológico de Monterrey

Actividad 1.1 Extracción de Características

Pilar Méndez Briones | A01736843

Analítica de datos y herramientas de inteligencia artificial

Alfredo García Suárez

Fecha de entrega

20 de marzo de 2025

Tabla de contenido

INTRODUCCIÓN

Para la extracción de características, ocupé mi dataframe limpio de Lyon, Francia, sin nulos ni outliers, para poder realizar un análisis detallado y sin valores que podrían afectar al mismo.

ANÁLISIS UNIVARIADO

Primer paso

Obtener un análisis univariado de las variables categóricas. Para esto, utilicé la siguiente función:

```python
#Obtengo un análisis univariado de las variables categóricas
freq_tbl (Francia)
```

Python

```
             host_name  frequency  percentage  cumulative_perc
0                David        130    0.013143         0.013143
1            Guillaume        127    0.012840         0.025983
2               Damien        127    0.012840         0.038823
3               Romain         99    0.010009         0.048832
4              Nicolas         97    0.009807         0.058639
...                ...        ...         ...              ...
2373  Cime De L'Aulnes          1    0.000101         0.999596
2374            Yassir          1    0.000101         0.999697
2375     Anne - Judith          1    0.000101         0.999798
2376      Léa & Emerik          1    0.000101         0.999899
2377           Mostafa          1    0.000101         1.000000

[2378 rows x 4 columns]
```

Esto me da el nombre y los valores de la variable categórica, así como su frecuencia, porcentaje y porcentaje acumulado. Datos que nos servirán para el análisis

Segundo paso

Escoger cada variable, eliminando las dos últimas columnas que no nos sirven con drop, quedándonos solo con frequency, ya que es la que nos va a servir para realizar las gráficas.

```python
#Elimino las columnas percentage y cumulative perc para trabajar con las frecuencias
table1 = table1.drop(['percentage','cumulative_perc'],axis=1)
table1
```

	host_location	frequency
0	Lyon, France	7901
1	Paris, France	260
2	France	109
3	Villeurbanne, France	70
4	Rennes, France	63

Tercer paso

Filtrar los valores más relevantes, para que nuestras gráficas no tengan demasiadas categorías que sea imposible visualizarlo. En algunos casos, no fue necesario ya que solo había dos categorías o eran menos de 5.

```python
#Filtro de los valores más relevantes de las variables (mayores a 30)
filtro = table1 [table1['frequency']>30]
filtro = filtro.set_index('host_location')
filtro
```

Python

host_location	frequency
Lyon, France	7901
Paris, France	260
France	109
Villeurbanne, France	70
Rennes, France	63
Caluire-et-Cuire, France	39
Annecy, France	34

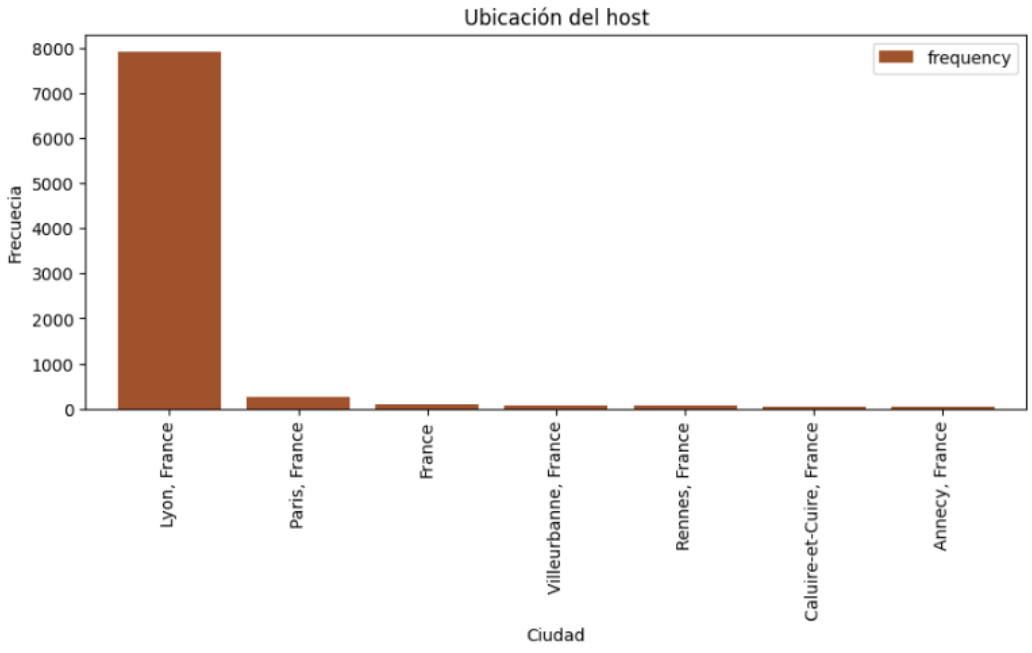
Cuarto paso

En este paso, ahora si viene el análisis de cada variable dependiendo la gráfica.

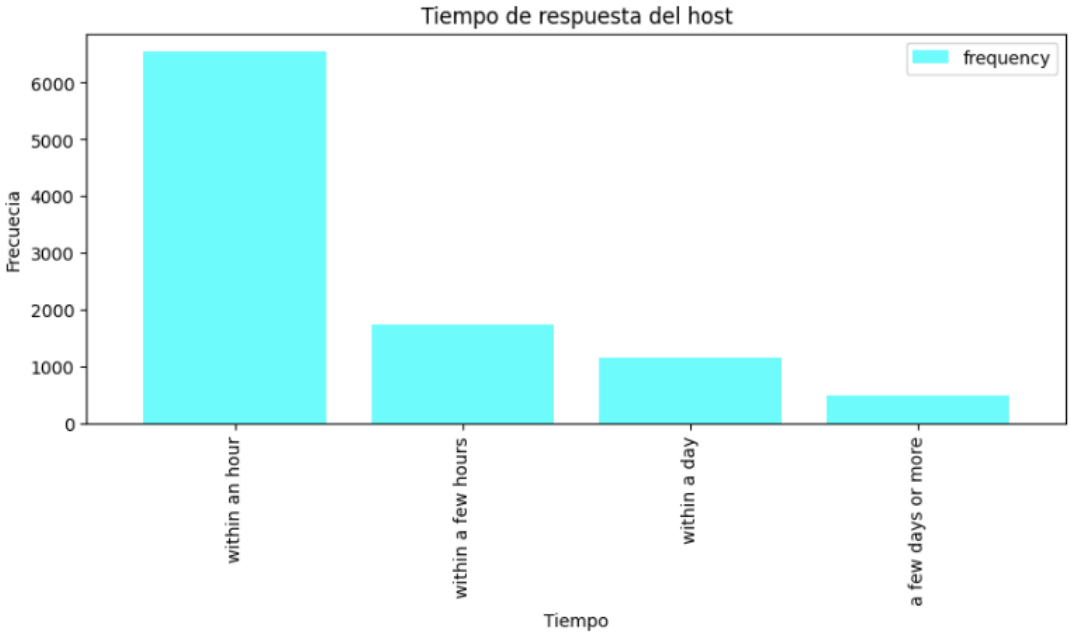
Para las 10 variables categóricas, realicé los tres gráficos, sin embargo, en este reporte, voy a escoger el gráfico que más se le adecue al tipo de variable.

Nota: La frecuencia es el número de registros obtenidos.

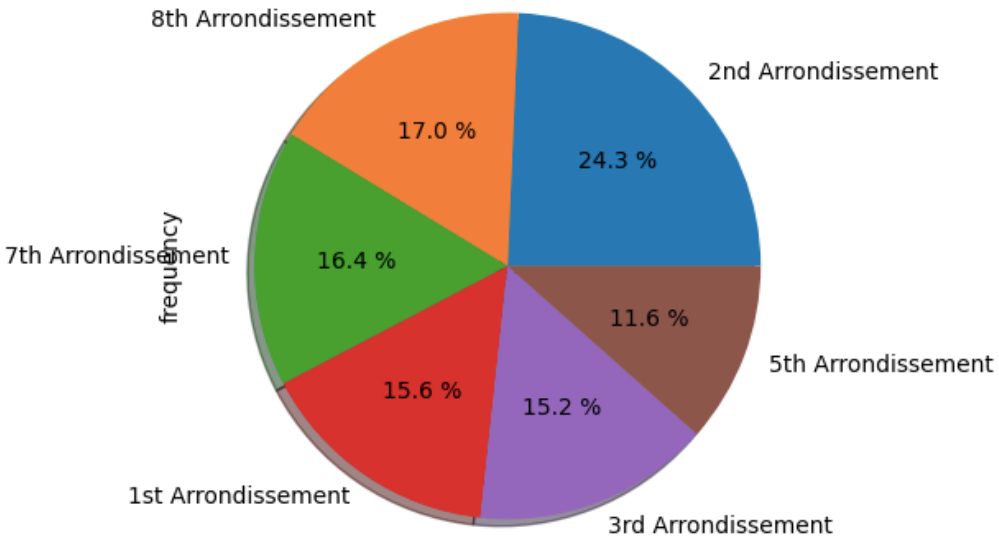
1. Host location: Con la gráfica de barras, podemos observar que la ciudad donde viven más los host que rentan en Lyon es la misma Lyon, esto tiene mucho sentido, ya que al final es más fácil rentar una casa y controlarla en tu misma ciudad, que andar viajando para administrarla desde otra.



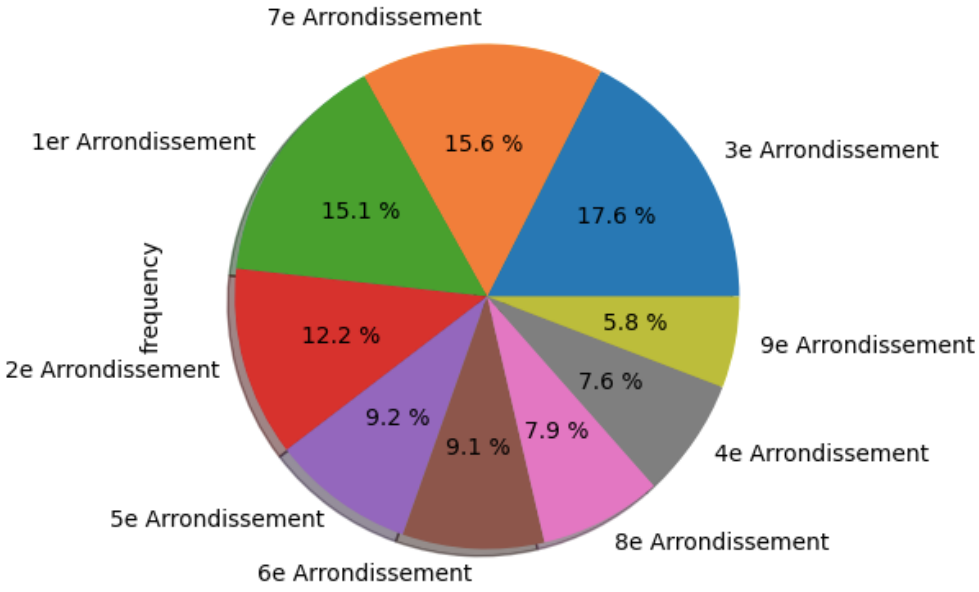
2. Host response time: Con la gráfica de barras, podemos obtener una comparación de cuánto tardan los hosts que rentan en Lyon en responder. La mayoría del tiempo de respuesta de los hosts es dentro de una hora, y el menor es en pocos días o más. Esto significa que tienen un corto rango de tiempo de respuesta y eso los hace eficientes (la frecuencia es por el número de datos registrados).



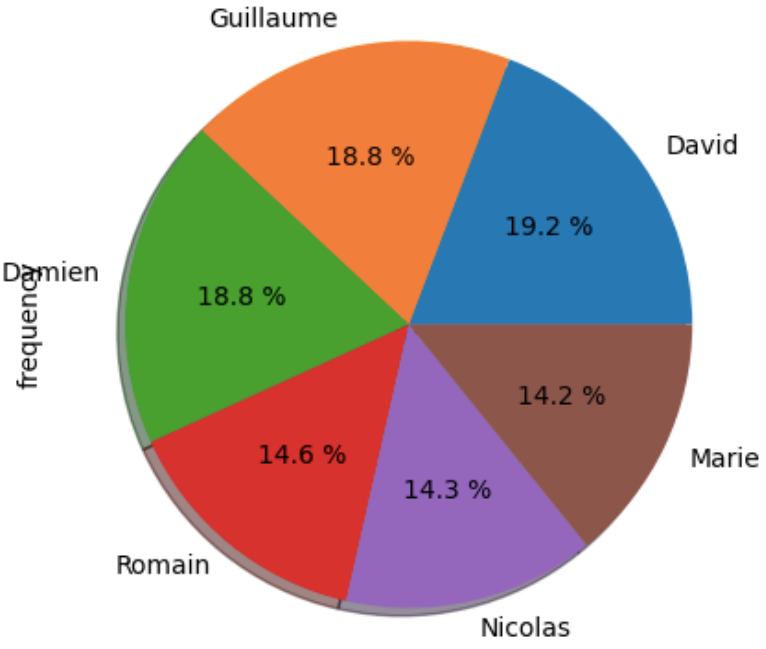
3. Host neighbourhood: Con la gráfica de pastel, podemos observar el porcentaje de los registros (frecuencia) que se encuentran en las diferentes colonias de la ciudad de Lyon. Podemos observar que la 2nd y 8th son las más concurridas. Esto porque son parte del centro y de las más habitadas poblacionalmente. Sin embargo, esta variable la ingresó el host manualmente. Más adelante veremos otra.



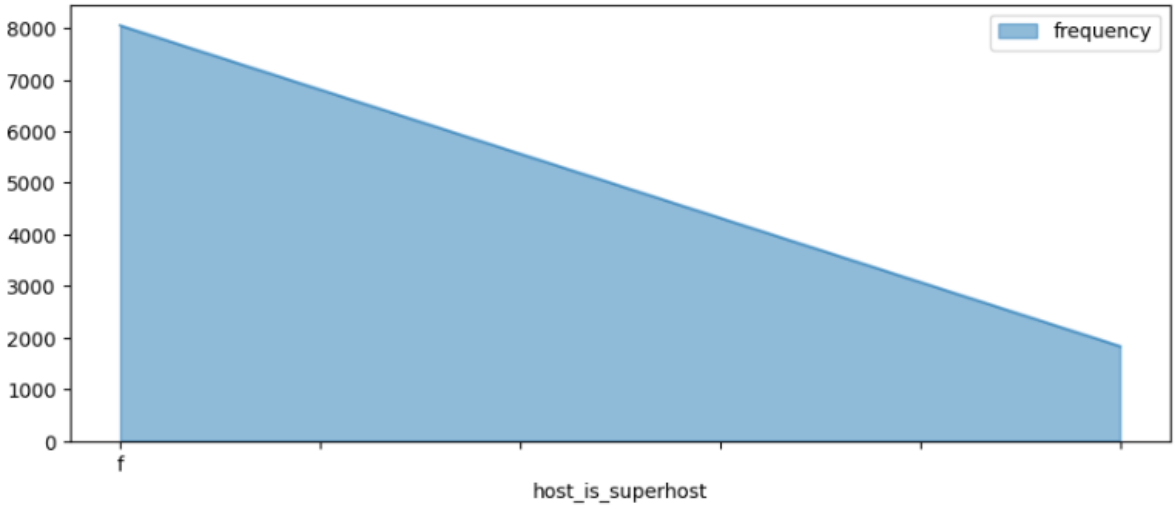
4. Negihbourhood cleansed: Esta variable a diferencia de la otra, ya fue modificada por Airbnb. Es decir, que realmente las direcciones con más registros son la 3ra y la 7ma. El cambio se puede deber a equivocaciones de parte del host.



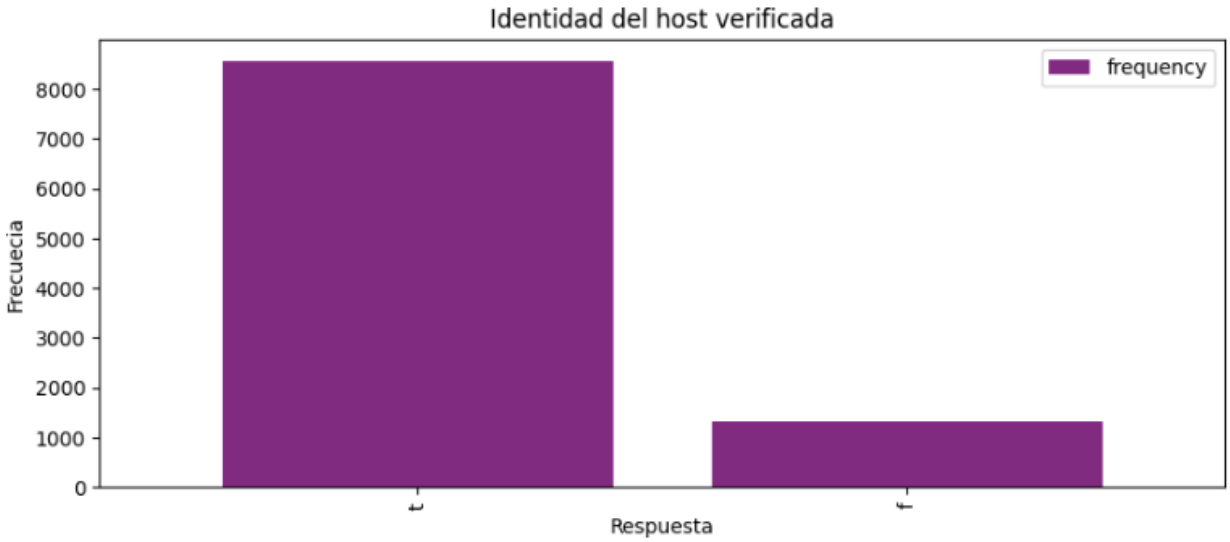
5. Host name: Con la gráfica de pastel, podemos comparar y ver la proporción de los nombres más comunes de los hosts en Lyon. Podemos observar que la mayoría de los host se llaman David, y que igual la mayoría son hombres. Esto es muy interesante porque podemos analizar si hay algún tipo de correlación entre ser host y el género. Si es que los hombres suelen rentar más a través de Airbnb que mujeres.



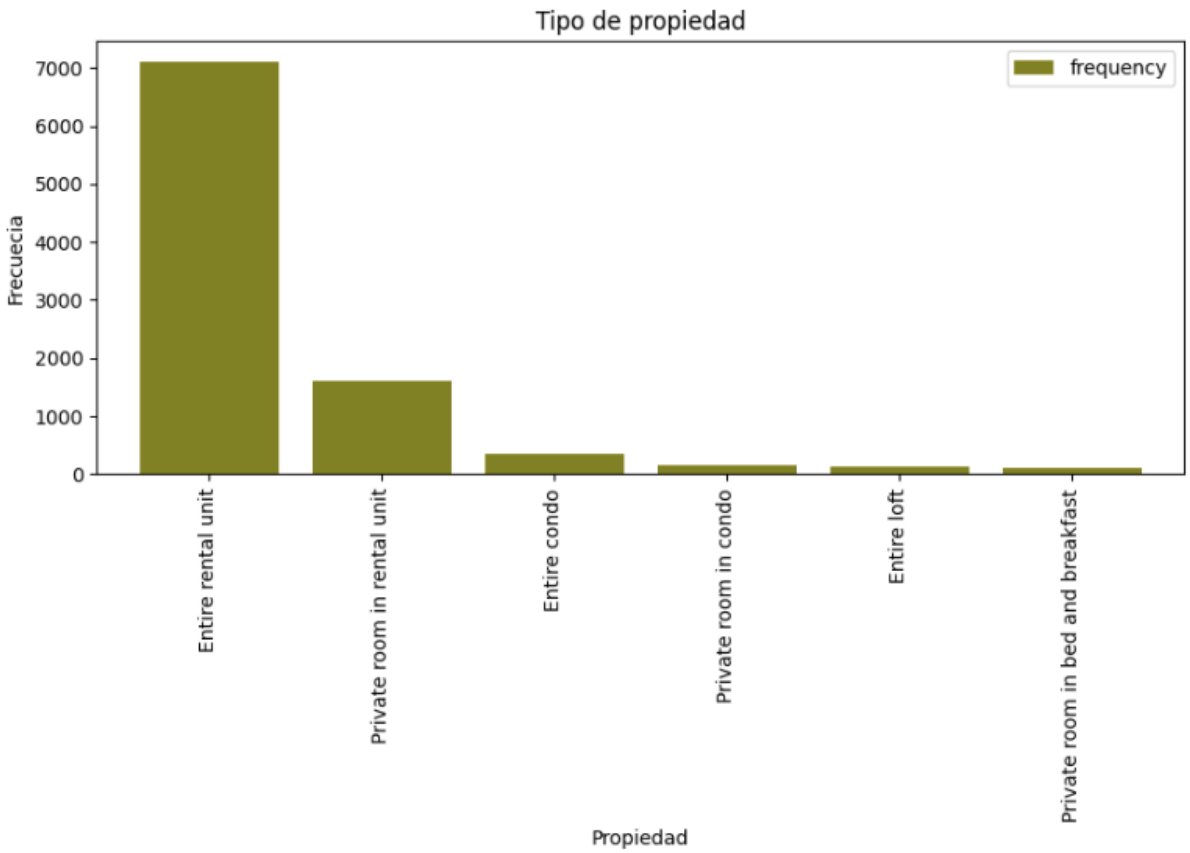
6. Host is super host: Con la gráfica de área, podemos observar que la mayoría de los hosts que rentan en Lyon, no son superhosts. Esto puede ser un indicador de que no es muy usual ser superhost en Lyon y que no es estrictamente necesario para poder rentar o generar confianza a los clientes.



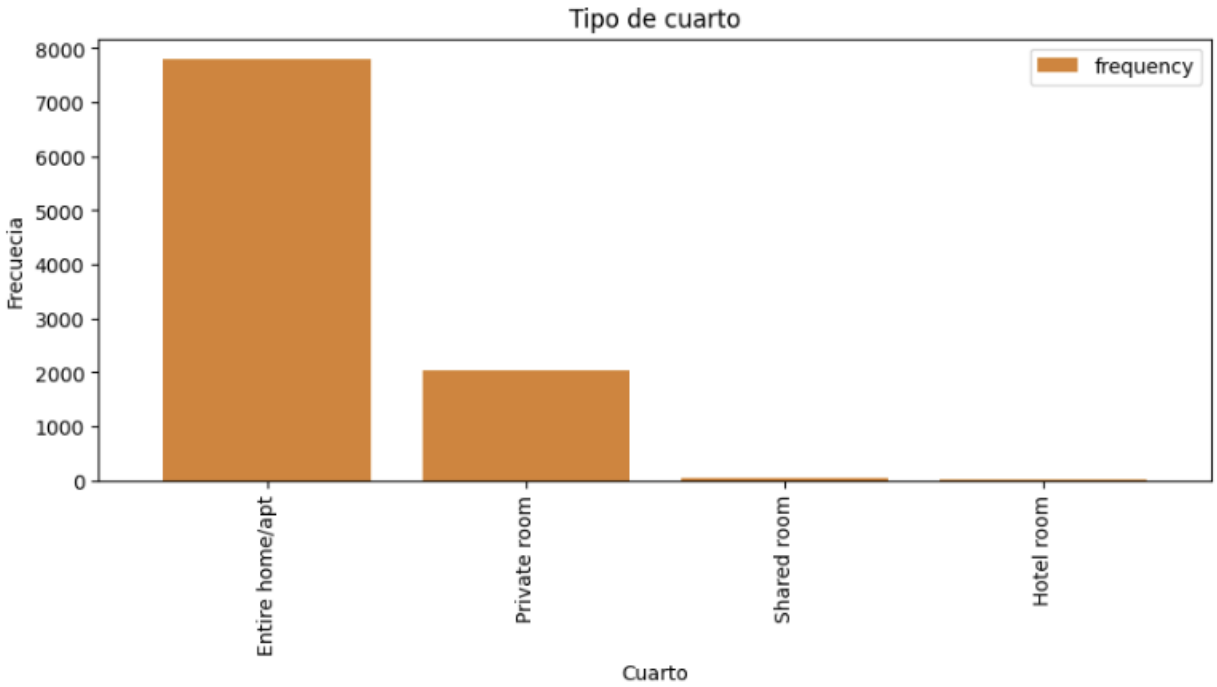
7. Host identity verified: Complementando con la variable anterior, aunque la mayoría no sean superhost, la mayoría si tienen su identidad verificada. Esto es un indicador que tener la identidad verificada si es primordial para poder rentar un alojamiento.



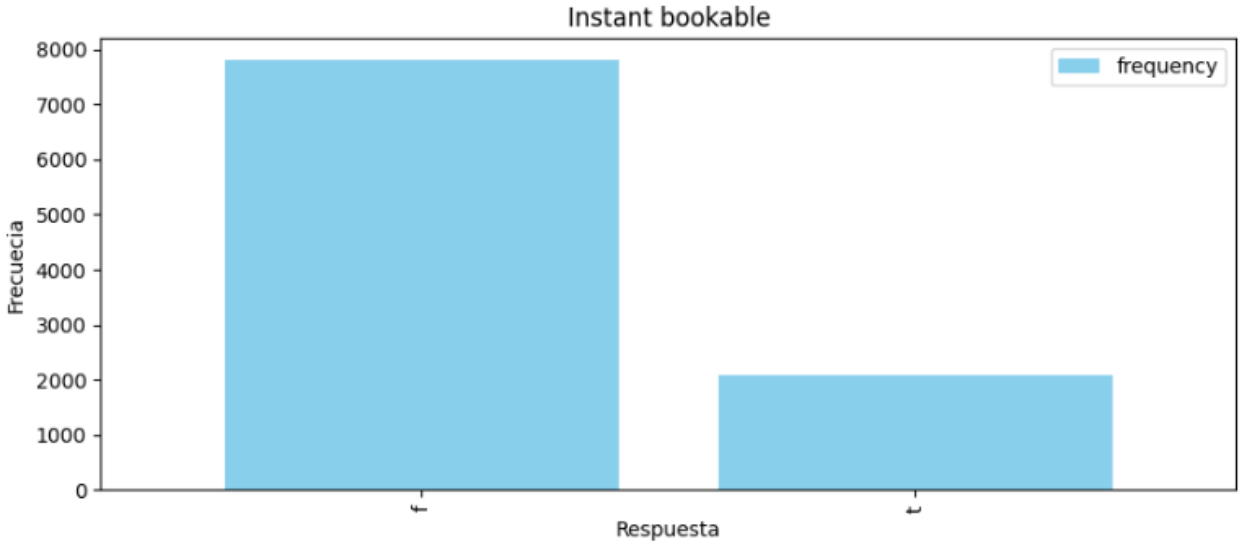
8. Property type: En el gráfico de barras podemos observar que la mayoría de las propiedades en renta son unidades enteras (casas), y que de ahí sigue la renta de cuartos. Esto podría indicar una gran afluencia de familias o grupo de personas (amigos) que van a visitar Lyon.



9. Room type: Esta variable se complementa con la anterior, ya que podemos observar que el tipo de lugar que se renta más es la casa entera, seguido por un cuarto privado.



10. Instant bookable: Con esta gráfica, podemos observar que la mayoría de registros tienen un falso, esto quiere decir que la mayoría de las propiedades no tienen disponibilidad y por eso no se puede reservar al instante. A lo mejor se debe a la fecha en la que se capturaron los datos, como en temporada alta, de vacaciones, etc.



DETERMINACIÓN DE CLASES PARA DATOS AGRUPADOS

Primer paso

Sacar el máximo y el mínimo de la variable numérica escogida, para poder usarlo más adelante en determinar nuestros límites.

```
1  #Obtenemos el limite superior y el límite inferior de la columna objetivo
2  Max=Francia['host_listings_count'].max()
3  Min=Francia['host_listings_count'].min()
4  Limites= [Min, Max]
5  Limites
```

✓ 0.1s

```
[1.0, 3.0]
```

Segundo paso

Calcular el rango R, restando el máximo con el mínimo.

```
1  #Calculamos el rango R
2  R = Max - Min
3  R
```

✓ 0.0s

```
2.0
```

Tercer paso

Calcular el número de intervalos ni con n que es la población que establecimos desde el principio (siendo 9890 registros en total), aplicando la fórmula / regla de Sturges.

```
#Calculamos el número de intervalos de clase 'ni', aplicando la regla de Struges.
ni = 1+3.32*np.log10(n)
ni
```

✓ 0.0s

```
14.264051688102635
```

Cuarto paso

Calcular el ancho del intervalo “i”, restando el rango R con el número de intervalos.

```
#Calculamos el ancho del intervalo 'i'
i = R / ni
i
```

✓ 0.0s

```
0.14021261586342684
```

Quinto paso

Usar la función de numpy de linspace, poniendo como parámetro una décima antes de nuestro límite inferior, una décima después de nuestro límite superior, y el redondeo del número de intervalos.

```
#Categorización de variables
#Declaramos 2 intervalos (menores de 100) y (mayores de 100)
#Ajustamos los limites para que todos los valores sean incluidos en los intervalos
intervalos=np.linspace(0.9, 3.1, 15)
intervalos
```

✓ 0.0s

```
array([0.9       , 1.05714286, 1.21428571, 1.37142857, 1.52857143,
       1.68571429, 1.84285714, 2.        , 2.15714286, 2.31428571,
       2.47142857, 2.62857143, 2.78571429, 2.94285714, 3.1       ])
```

Sexto paso

Crear el nombre de las categorías de acuerdo a lo que sea más útil, yo en mi caso me base en los intervalos. Posteriormente, sustituir los valores por las categorías con cut.

```
#Creamos las categorías
categorias = ['0.9 - 1.06', '1.06 - 1.21', '1.21 - 1.37', '1.37 - 1.53',
 '1.53 - 1.69', '1.69 - 1.84', '1.84 - 2.0', '2.0 - 2.16',
 '2.16 - 2.31', '2.31 - 2.47', '2.47 - 2.63', '2.63 - 2.79',
 '2.79 - 2.94', '2.94 - 3.1']
```

✓ 0.0s Python

```
#Finalmente creamos las categorías en la columna numérica
Francia['host_listings_count'] = pd.cut(x=Francia['host_listings_count'],bins = intervalos, labels =cat
Francia['host_listings_count']
```

1. Host listings count: La variable fue categorizada para representar rangos de cantidad de anuncios por anfitrión. Las nuevas etiquetas permiten diferenciar anfitriones

ocasionales de profesionales. Se observa que la mayoría tiene solo 1 anuncio, lo que indica que predominan anfitriones individuales.

```
#Creamos las categorías
categorias = ['0.9 - 1.06', '1.06 - 1.21', '1.21 - 1.37', '1.37 - 1.53',
 '1.53 - 1.69', '1.69 - 1.84', '1.84 - 2.0', '2.0 - 2.16',
 '2.16 - 2.31', '2.31 - 2.47', '2.47 - 2.63', '2.63 - 2.79',
 '2.79 - 2.94', '2.94 - 3.1']
```

✓ 0.0s Python

```
#Finalmente creamos las categorías en la columna numérica
Francia['host_listings_count'] = pd.cut(x=Francia['host_listings_count'],bins = intervalos, labels =cat
Francia['host_listings_count']
```

2. Accommodates: Se convirtió en variable categórica para agrupar la capacidad de personas por alojamiento. Esto facilita clasificar propiedades según si son para parejas, familias o grupos. Se observa que la mayoría de alojamientos admiten 2 o 4 personas, lo que sugiere enfoque turístico estándar.

```
#Creamos las categorías
categorias = ['0.9 - 1.34', '1.34 - 1.79', '1.79 - 2.23', '2.23 - 2.67',
 '2.67 - 3.11', '3.11 - 3.56', '3.56 - 4.0', '4.0 - 4.44',
 '4.44 - 4.89', '4.89 - 5.33', '5.33 - 5.77', '5.77 - 6.21',
 '6.21 - 6.66', '6.66 - 7.1']
```

Python

```
#Finalmente creamos las categorías en la columna numérica
Francia['accommodates'] = pd.cut(x=Francia['accommodates'],bins = intervalos, labels =categorias)
Francia['accommodates']
```

Python

```
9863    2.67 - 3.11
9864     4.0 - 4.44
9865    1.79 - 2.23
```

3. Bathrooms: La variable fue agrupada en una sola categoría porque todos los registros tienen 1 baño. Se mantuvo la transformación para mantener consistencia con otras variables categorizadas. Esto indica que la oferta es homogénea en términos de servicios sanitarios.

```
#Creamos las categorías
categorias = ['0.9 - 0.914', '0.914 - 0.929', '0.929 - 0.943', '0.943 - 0.957',
 '0.957 - 0.971', '0.971 - 0.986', '0.986 - 1.0', '1.0 - 1.014',
 '1.014 - 1.029', '1.029 - 1.043', '1.043 - 1.057', '1.057 - 1.071',
 '1.071 - 1.086', '1.086 - 1.1']

#Finalmente creamos las categorías en la columna numérica
Francia['bathrooms'] = pd.cut(x=Francia['bathrooms'],bins = intervalos, labels =categorias)
Francia['bathrooms']
```

P

```
156    0.986 - 1.0
157    0.986 - 1.0
158    0.986 - 1.0
```

4. Beds: La variable se transformó en clases para indicar la cantidad de camas disponibles. Ahora es más claro distinguir entre alojamientos para 1, 2 o más personas. La mayoría tiene 1 cama, lo que refuerza la idea de alojamientos pensados para estancias cortas o parejas.

```python
#Creamos las categorías
categorias = ['Sin cama', 'Cama individual mínima', '1 cama individual', 'Cama individual cómoda',
              'Cama doble pequeña', '1 cama doble', 'Cama doble amplia', '2 camas individuales',
              'Litera o 2 camas', '3 camas', '4 camas', '5 camas', '6 camas', 'Más de 6 camas']
```

Python

```python
#Finalmente creamos las categorías en la columna numérica
Francia['beds'] = pd.cut(x=Francia['beds'],bins = intervalos, labels =categorias)
Francia['beds']
```

Python

```
63                4 camas
64       Cama doble amplia
65       Cama doble amplia
66          Más de 6 camas
67                4 camas
```

5. Price: El precio fue categorizado en rangos que permiten comparar distintos niveles de alojamiento. Esto facilita distinguir entre precios bajos, medios y altos. Se observa una gran concentración en precios de 70 a 85 euros, lo que indica una tendencia de mercado hacia tarifas accesibles.

```python
#Creamos las categorías
categorias = ['Categoria1(59-62 euros)','Categoría2(62-65 euros)','Categoría3(65-68 euros)','Categoría4(
              'Categoría5(71-74 euros)','Categoría6(74-77 euros)','Categoría7(77-80 euros)','Categoría8(
              'Categoria9(82-85 euros)','Categoria10(85-88 euros)','Categoria11(88-91 euros)','Categoria
              'Categoria13(94-97 euros)','Categoria14(97-100 euros)']
```

Python

```python
#Finalmente creamos las categorías en la columna numérica
Francia['price'] = pd.cut(x=Francia['price'],bins = intervalos, labels =categorias)
Francia['price']
```

Python

```
863    Categoría5(71-74 euros)
864    Categoría2(62-65 euros)
865    Categoría7(77-80 euros)
866    Categoría7(77-80 euros)
```

6. Minimum nights average ntm: Se categorizó esta variable para distinguir el tipo de estancia esperada (corta, media o larga). Las etiquetas creadas ayudan a detectar si los anfitriones priorizan reservas rápidas o prolongadas. La mayoría permite estancias cortas (1 noche), lo que refleja una alta flexibilidad en la oferta.

```python
#Creamos las categorías
categorias = ['Categoría1(1-4 noches)','Categoría2(4-7 noches)','Categoría3(7-10 noches)','Categoría4(1
              'Categoría5(13-16 noches)','Categoría6(16-19 noches)','Categoría7(19-23 noches)','Categor
              'Categoria9(26-29 noches)','Categoria10(29-32 noches)','Categoria11(32-35 noches)','Categ
              'Categoria13(38-41 noches)','Categoria14(41-45 noches)']
```

Python

```python
#Finalmente creamos las categorías en la columna numérica
Francia['minimum_nights_avg_ntm'] = pd.cut(x=Francia['minimum_nights_avg_ntm'],bins = intervalos, label
Francia['minimum_nights_avg_ntm']
```

Python

```
63      Categoría1(1-4 noches)
64      Categoría1(1-4 noches)
65      Categoría1(1-4 noches)
```

7. Availability 365: Agrupar la disponibilidad permite identificar la frecuencia con la que los alojamientos están activos. Ahora es más fácil distinguir entre propiedades temporales y permanentes. Se observa que muchos anuncios están disponibles los 365 días, lo que sugiere un uso comercial continuo.

```python
#Creamos las categorías
categorias = ['Categoría1(0-26 días)','Categoría2(26-52 días)','Categoría3(52-78 días)','Categoría4(78-1
              'Categoría5(104-130 días)','Categoría6(130-156 días)','Categoría7(156-183 días)','Categorí
              'Categoria9(209-235 días)','Categoria10(235-261 días)','Categoria11(261-287 días)','Categc
              'Categoria13(313-339 días)','Categoria14(339-366 días)']
```

Python

```python
#Finalmente creamos las categorías en la columna numérica
Francia['availability_365'] = pd.cut(x=Francia['availability_365'],bins = intervalos, labels =categorias
Francia['availability_365']
```

Python

```
63      Categoría1(0-26 días)
64      Categoría1(0-26 días)
```

8. Number of reviews: Se agruparon las reseñas para conocer la popularidad del alojamiento. Esto permite diferenciar propiedades nuevas de otras muy activas en la plataforma. La mayoría tiene pocas reseñas, lo que puede deberse a una alta rotación de nuevos anfitriones.

```python
#Creamos las categorías
categorias = ['Categoría1(0-5 reviews)','Categoría2(5-12 reviews)','Categoría3(12-19 reviews)','Categorí
              'Categoría5(25-32 reviews)','Categoría6(32-39 reviews)','Categoría7(39-46 reviews)','Categ
              'Categoria9(52-59 reviews)','Categoria10(59-66 reviews)','Categoria11(66-72 reviews)','Cat
              'Categoria13(79-86 reviews)','Categoria14(86-93 reviews)']
```

Python

```python
#Finalmente creamos las categorías en la columna numérica
Francia['number_of_reviews'] = pd.cut(x=Francia['number_of_reviews'],bins = intervalos, labels =categori
Francia['number_of_reviews']
```

Python

```
9863       Categoría1(0-5 reviews)
9864       Categoría1(0-5 reviews)
9865       Categoría1(0-5 reviews)
```

9. Review scores ratings: Se categorizó el puntaje de reseñas para facilitar la interpretación de la calidad del servicio. Las etiquetas permiten identificar fácilmente si el alojamiento es altamente valorado o no. La mayoría tiene calificaciones muy altas, especialmente 5.0, lo que indica buena satisfacción general.

```python
#Creamos las categorías
categorias = ['4.1 - 4.17: Muy baja','4.17 - 4.24: Baja','4.24 - 4.31: Regular baja','4.31 - 4.39: Regul
 '4.39 - 4.46: Aceptable','4.46 - 4.53: Buena','4.53 - 4.6: Buena alta','4.6 - 4.67: Muy buena','4.67 -
 '4.74 - 4.81: Excelente','4.81 - 4.89: Excelente alta','4.89 - 4.96: Top rating','4.96 - 5.03: Perfecta
 '5.03 - 5.1: Impecable']
```

Python

```python
#Finalmente creamos las categorías en la columna numérica
Francia['review_scores_rating'] = pd.cut(x=Francia['review_scores_rating'],bins = intervalos, labels =ca
Francia['review_scores_rating']
```

Python

```
      4.74 - 4.81: Excelente
      4.74 - 4.81: Excelente
      4.74 - 4.81: Excelente
      4.74 - 4.81: Excelente
```

10. Review per month: Se agruparon las frecuencias de reseñas mensuales para estimar el nivel de actividad. Las nuevas etiquetas ayudan a distinguir propiedades con poca rotación de huéspedes de las más populares. Se observó que la mayoría tiene entre 0.01 y 0.2 reseñas por mes, lo que indica una actividad baja o estacional.

```python
#Creamos las categorías
categorias = ['Sin reseñas', 'Muy baja', 'Baja', 'Baja-media', 'Media',
              'Media-alta', 'Alta', 'Alta constante', 'Muy alta', 'Muy alta +',
              'Intensa', 'Frecuencia alta', 'Frecuencia muy alta', 'Top host']
```

Python

```python
#Finalmente creamos las categorías en la columna numérica
Francia['reviews_per_month'] = pd.cut(x=Francia['reviews_per_month'],bins = intervalos, labels =categori
Francia['reviews_per_month']
```

Python

```
9863                Baja
9864                Baja
```