Proyecto - Analisis Exploratoria de Datos (EDA) y Visualizacion de base de datos AirBnb 2024 - New York

Metodologia

1. Importar dependencias (library)
2. Cargar base de datos
3. Exploracion inicial
4. Limpieza y transformacion de datos
5. Analis de datos

1. Importar dependencias

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

2. Cargar base de datos

In [ ]:
```
data = pd.read_csv('EDA-dataset.csv', encoding_errors='ignore')
```

3. Exploracion inicial

In [ ]:
```
data.head()
```

Out[ ]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood
0	1.312228e+06	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382	Walter	Brooklyn	Clinton Hill
1	4.527754e+07	Rental unit in New York · ★4.67 · 2 bedrooms ·...	51501835	Jeniffer	Manhattan	Hell's Kitchen
2	9.710000e+17	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354	Joshua	Manhattan	Chelsea
3	3.857863e+06	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271	John And Catherine	Manhattan	Washington Heights
4	4.089661e+07	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963	Stay With Vibe	Manhattan	Murray Hill

5 rows × 22 columns

In [ ]:
```
data.tail()
```

Out[ ]:

	id	name	host_id	host_name	neighbourhood_group	neighbourho
20765	2.473690e+07	Rental unit in New York · ★4.75 · 1 bedroom · ...	186680487	Henry D	Manhattan	Lower East S
20766	2.835711e+06	Rental unit in New York · ★4.46 · 1 bedroom · ...	3237504	Aspen	Manhattan	Greenw Vill
20767	5.182527e+07	Rental unit in New York · ★4.93 · 1 bedroom · ...	304317395	Jeff	Manhattan	Hell's Kitc
20768	7.830000e+17	Rental unit in New York · ★5.0 · 1 bedroom · 1...	163083101	Marissa	Manhattan	Chinato
20769	5.660000e+17	Rental unit in Queens · ★4.89 · 1 bedroom · 1 ...	93827372	Glenroy	Queens	Rosec

5 rows × 22 columns

In [ ]:
```
data.shape
```

Out[ ]:
```
(20770, 22)
```

In [ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20770 entries, 0 to 20769
Data columns (total 22 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              20770 non-null  float64
 1   name                            20770 non-null  object
 2   host_id                         20770 non-null  int64
 3   host_name                       20770 non-null  object
 4   neighbourhood_group             20770 non-null  object
 5   neighbourhood                   20763 non-null  object
 6   latitude                        20763 non-null  float64
 7   longitude                       20763 non-null  float64
 8   room_type                       20763 non-null  object
 9   price                           20736 non-null  float64
 10  minimum_nights                  20763 non-null  float64
 11  number_of_reviews               20763 non-null  float64
 12  last_review                     20763 non-null  object
 13  reviews_per_month               20763 non-null  float64
 14  calculated_host_listings_count  20763 non-null  float64
 15  availability_365                20763 non-null  float64
 16  number_of_reviews_ltm           20763 non-null  float64
 17  license                         20770 non-null  object
 18  rating                          20770 non-null  object
 19  bedrooms                        20770 non-null  object
 20  beds                            20770 non-null  int64
 21  baths                           20770 non-null  object
dtypes: float64(10), int64(2), object(10)
memory usage: 3.5+ MB
```

In [ ]:
```
# resumen estadistico
data.describe()
```

Out[ ]:

	id	host_id	latitude	longitude	price	minimum_r
count	2.077000e+04	2.077000e+04	20763.000000	20763.000000	20736.000000	20763.0
mean	3.033858e+17	1.749049e+08	40.726821	-73.939179	187.714940	28.5
std	3.901221e+17	1.725657e+08	0.060293	0.061403	1023.245124	33.5
min	2.595000e+03	1.678000e+03	40.500314	-74.249840	10.000000	1.0
25%	2.707260e+07	2.041184e+07	40.684159	-73.980755	80.000000	30.0
50%	4.992852e+07	1.086990e+08	40.722890	-73.949597	125.000000	30.0
75%	7.220000e+17	3.143997e+08	40.763106	-73.917475	199.000000	30.0
max	1.050000e+18	5.504035e+08	40.911147	-73.713650	100000.000000	1250.0

4. Preprocesamiento de datos

In [ ]:
```
#Total de datos nulos
data.isnull().sum()
```

```
# eliminando valores nulos
data.dropna(inplace=True)
data.isnull().sum()
```

Out[ ]:

```
id                                0
name                              0
host_id                           0
host_name                         0
neighbourhood_group               0
neighbourhood                     0
latitude                          0
longitude                         0
room_type                         0
price                             0
minimum_nights                    0
number_of_reviews                 0
last_review                       0
reviews_per_month                 0
calculated_host_listings_count    0
availability_365                  0
number_of_reviews_ltm             0
license                           0
rating                            0
bedrooms                          0
beds                              0
baths                             0
dtype: int64
```

In [ ]:

```
# total de valores duplicados
data.duplicated().sum()
# Eliminando duplicados
data.drop_duplicates(inplace=True)
data.duplicated().sum()
```

Out[ ]:

```
0
```

In [ ]:

```
#Tipos de datos
data.dtypes
```

Out[ ]:
```
id                                float64
name                               object
host_id                             int64
host_name                          object
neighbourhood_group                object
neighbourhood                      object
latitude                          float64
longitude                         float64
room_type                          object
price                             float64
minimum_nights                    float64
number_of_reviews                 float64
last_review                        object
reviews_per_month                 float64
calculated_host_listings_count    float64
availability_365                  float64
number_of_reviews_ltm             float64
license                            object
rating                             object
bedrooms                           object
beds                                int64
baths                              object
dtype: object
```

In [ ]:
```
#Cambiar el tipo de dato
data['id'] = data['id'].astype(object)
data.dtypes
data['host_id'] = data['host_id'].astype(object)
data.dtypes
```

Out[ ]:
```
id                                object
name                              object
host_id                           object
host_name                         object
neighbourhood_group               object
neighbourhood                     object
latitude                         float64
longitude                        float64
room_type                         object
price                            float64
minimum_nights                   float64
number_of_reviews                float64
last_review                       object
reviews_per_month                float64
calculated_host_listings_count   float64
availability_365                 float64
number_of_reviews_ltm            float64
license                           object
rating                            object
bedrooms                          object
beds                               int64
baths                             object
dtype: object
```

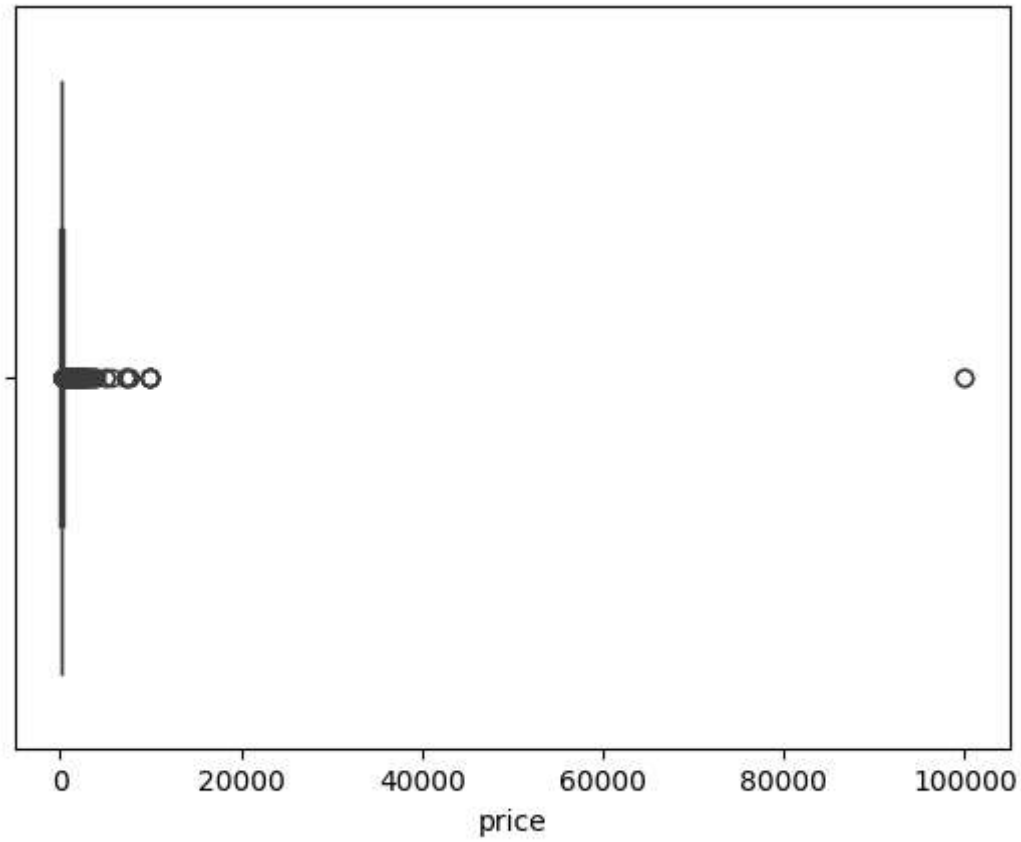
EDA

5. Analisis de datos

Analisis univariado: Tecnica estadistica utilizada para describir y analizar una sola variable a la vez.

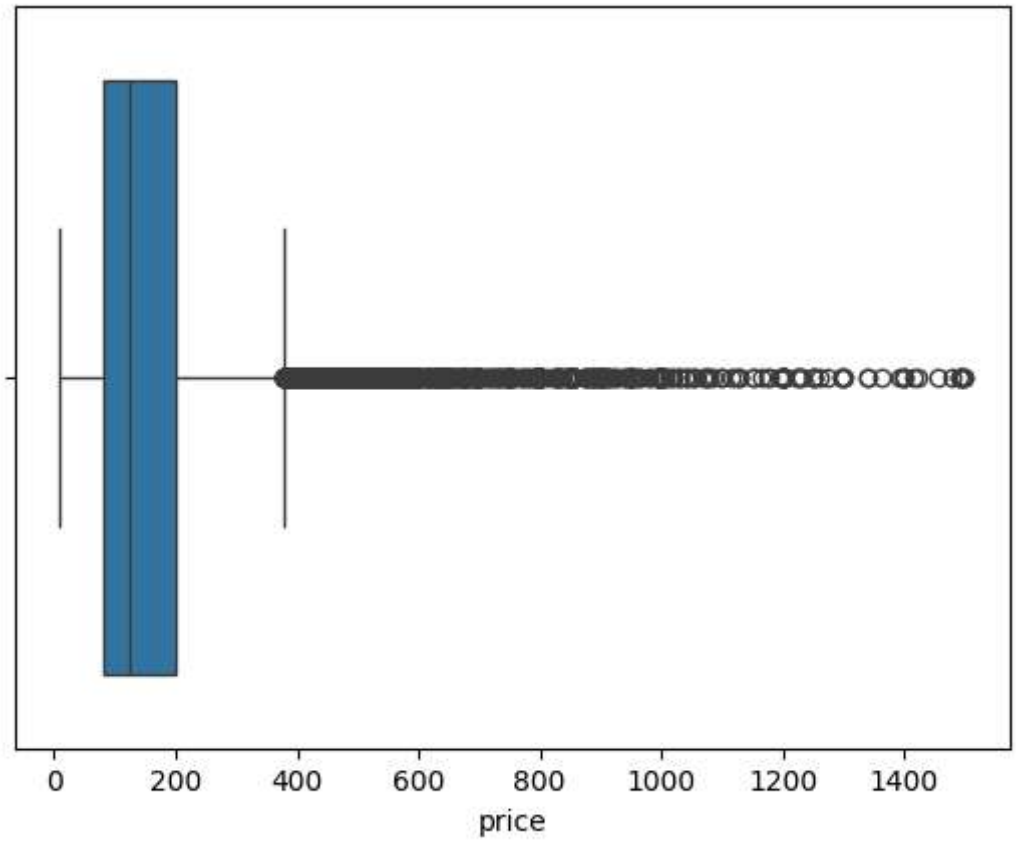
In [ ]:
```
# identificando valores atipicos (Valores fuera del promedio de la muestra)
sns.boxplot(data=data, x='price')
```

Out[ ]:
```
<Axes: xlabel='price'>
```

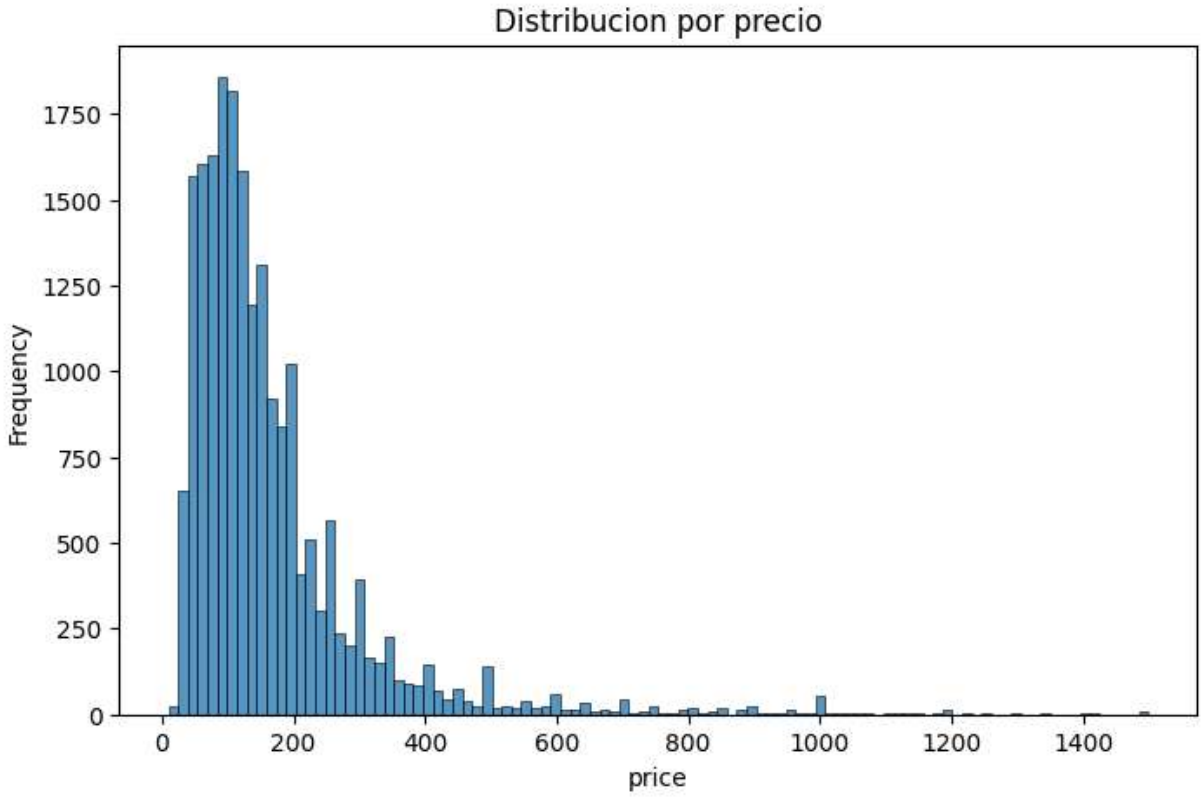


In [ ]:
```
#Filtrando outliers
df = data[data['price'] < 1500]
sns.boxplot(data=df, x='price')
```

Out[ ]:
```
<Axes: xlabel='price'>
```



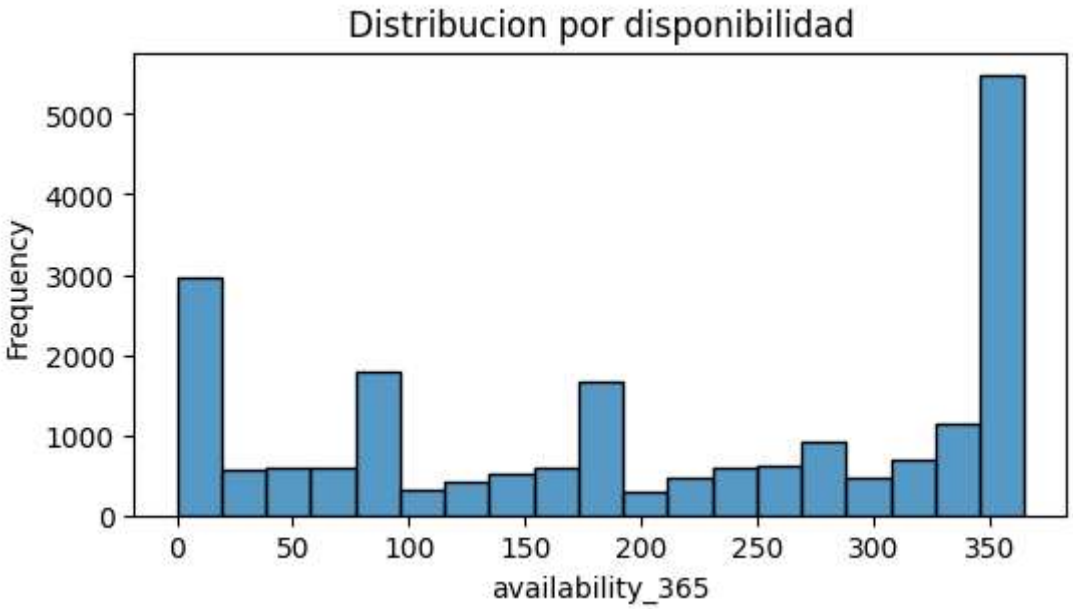
In [ ]:
```
# Distribucion de peso

plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='price', bins=100)
plt.title('Distribucion por precio')
plt.ylabel("Frequency")
plt.show()
```



In [ ]:
```
df.dtypes
```

Out[ ]:
```
id                                 object
name                               object
host_id                            object
host_name                          object
neighbourhood_group                object
neighbourhood                      object
latitude                          float64
longitude                         float64
room_type                          object
price                             float64
minimum_nights                    float64
number_of_reviews                 float64
last_review                        object
reviews_per_month                 float64
calculated_host_listings_count    float64
availability_365                  float64
number_of_reviews_ltm             float64
license                            object
rating                             object
bedrooms                           object
beds                                int64
baths                              object
dtype: object
```

In [ ]:
```
plt.figure(figsize=(6, 3))
sns.histplot(data=df, x='availability_365')
plt.title('Distribucion por disponibilidad')
plt.ylabel("Frequency")
plt.show()
```



Ingenieria de caracteristicas

In [ ]:
```
#Precio promedio por vecindario
df.groupby(by='neighbourhood_group')['price'].mean()
```

Out[ ]:
```
neighbourhood_group
Bronx            107.990506
Brooklyn         155.138317
Manhattan        204.146014
Queens           121.681939
Staten Island    118.780069
Name: price, dtype: float64
```

In [ ]:
```
data['price per bed'] = data['price']/data['beds']
data.head()
```

Out[ ]:

	id	name	host_id	host_name	neighbourhood_group	neighb
0	1312228.0	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382	Walter	Brooklyn	Cli
1	45277537.0	Rental unit in New York · ★4.67 · 2 bedrooms ·...	51501835	Jeniffer	Manhattan	Hell's
2	971000000000000000.0	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354	Joshua	Manhattan	
3	3857863.0	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271	John And Catherine	Manhattan	Wa
4	40896611.0	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963	Stay With Vibe	Manhattan	M

5 rows × 23 columns

In [ ]:
```
# Precio promedio por cama
data.groupby(by='neighbourhood_group')['price per bed'].mean()
```

Out[ ]:
```
neighbourhood_group
Bronx             85.165996
Brooklyn         128.361667
Manhattan        151.931643
Queens            78.507630
Staten Island     67.728101
Name: price per bed, dtype: float64
```

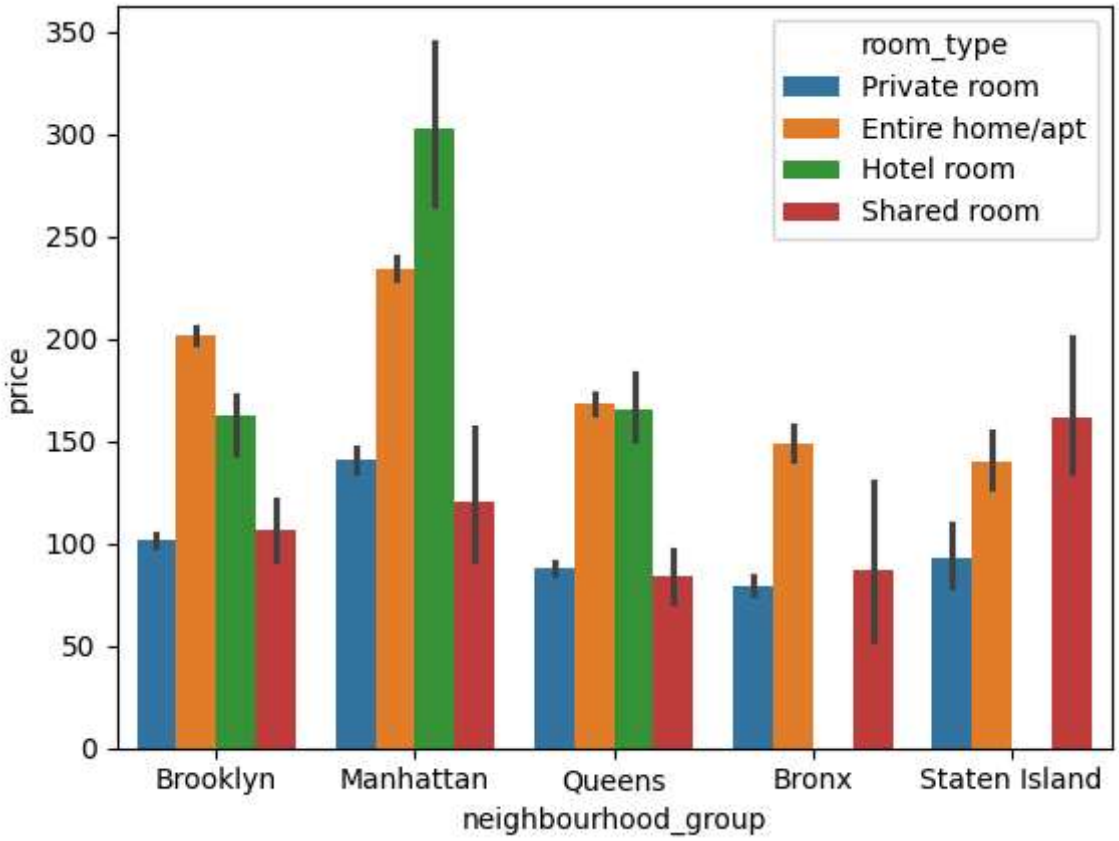
Bi Analysis (Analis bivariado): es una técnica estadística que se utiliza para explorar la relación entre dos variables. Su propósito es entender cómo una variable cambia respecto a otra.

In [ ]:
```
df.columns
```

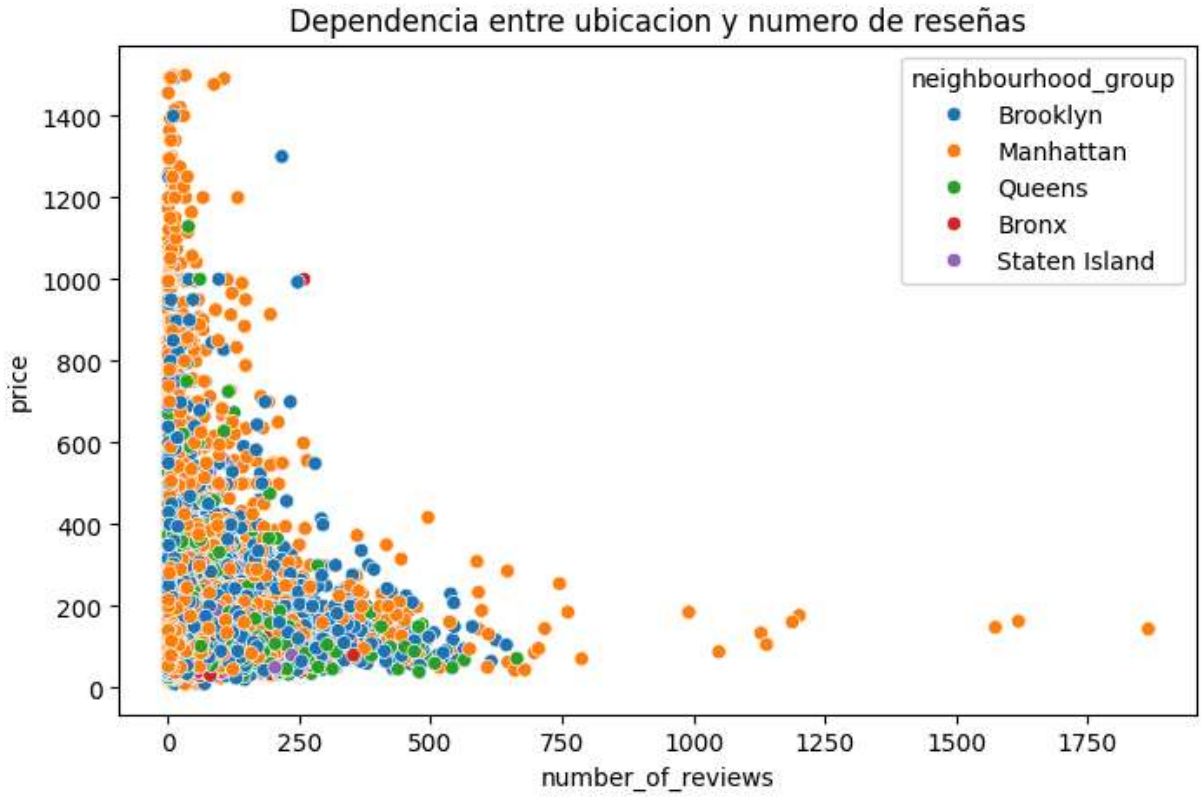
Out[ ]:
```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
       'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'calculated_host_listings_count',
       'availability_365', 'number_of_reviews_ltm', 'license', 'rating',
       'bedrooms', 'beds', 'baths', 'price per bed'],
      dtype='object')
```

In [ ]:
```
# Dependencia de la variable precio al vecindario
sns.barplot(data=df, x='neighbourhood_group', y='price', hue='room_type')
```

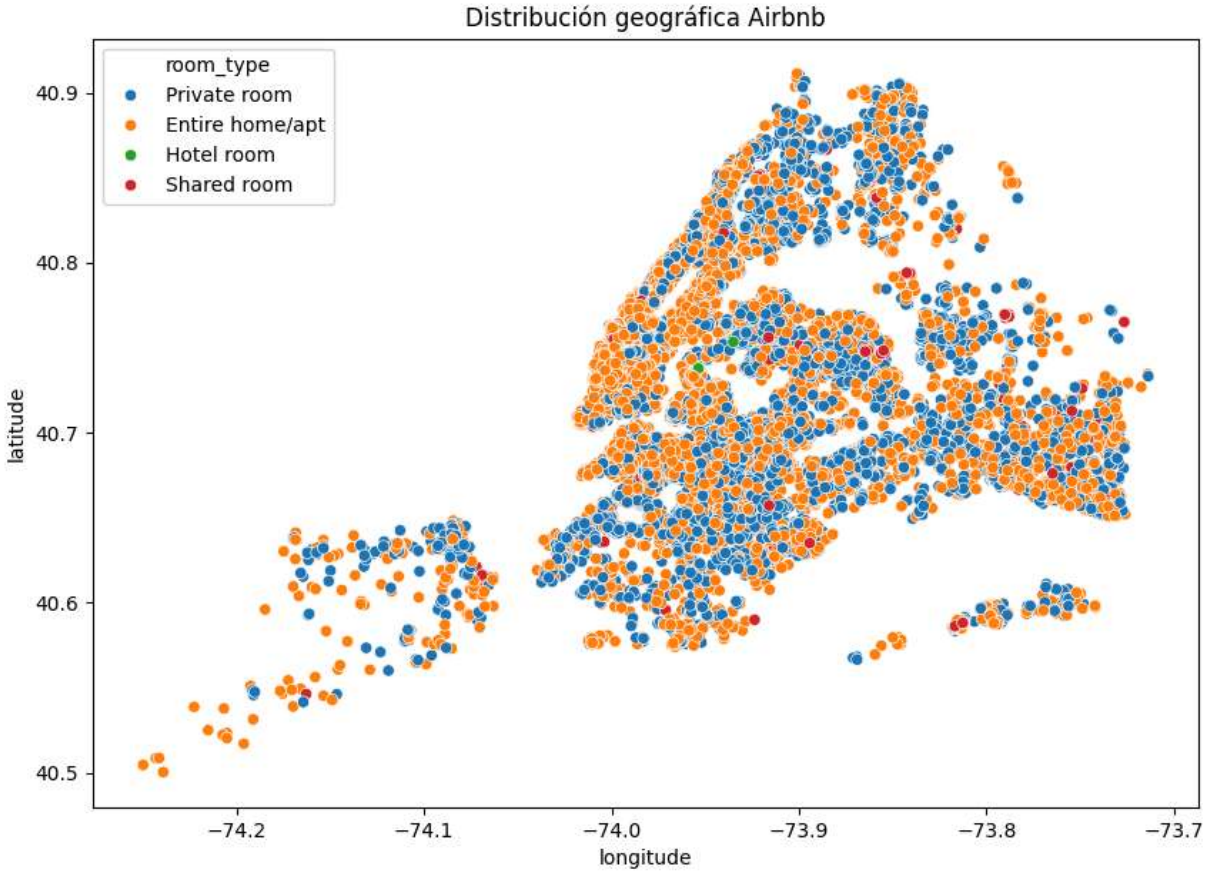
Out[ ]:
```
<Axes: xlabel='neighbourhood_group', ylabel='price'>
```



In [ ]:
```
# Relacion precio con el numero de reseñas
plt.figure(figsize=(8, 5))
plt.title("Dependencia entre ubicacion y numero de reseñas")
sns.scatterplot(data=df, x='number_of_reviews', y='price', hue='neighbourhood_group
plt.show()
```



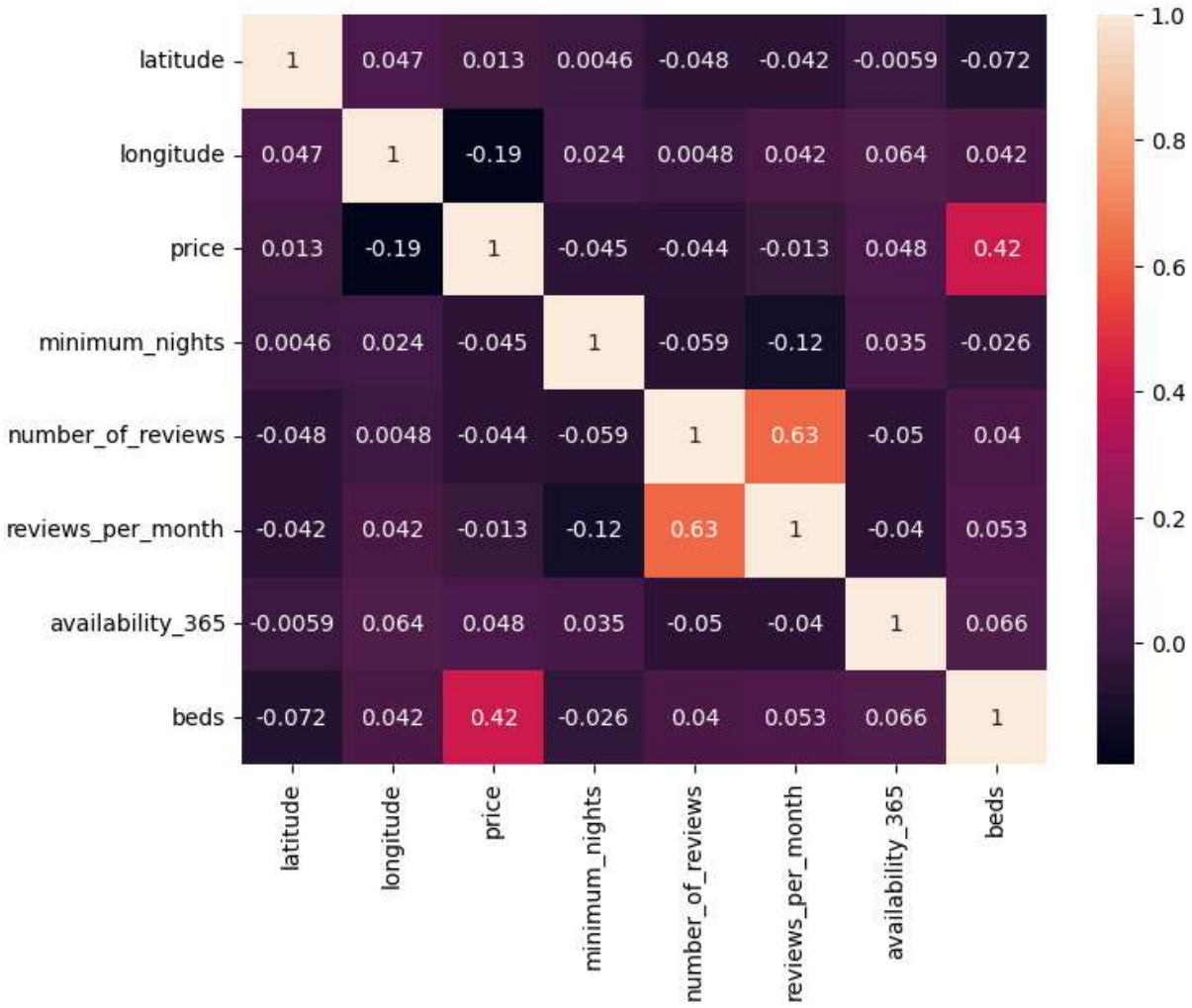
In [ ]:
```
#Distribución geográfica de los anuncios de Airbnb
plt.figure(figsize=(10, 7))
sns.scatterplot(data=df, x='longitude', y='latitude', hue='room_type')
plt.title("Distribución geográfica Airbnb")
plt.show()
```



Nota: Correlacion es la medida estadística que indica si existe una asociación entre dos variables cuantitativas, cuantificando la fuerza y la dirección de dicha relación. Existe una correlación positiva cuando las variables tienden a cambiar en el mismo sentido, una correlación negativa cuando cambian en sentidos opuestos, y una ausencia de correlación cuando no hay una tendencia aparente

In [ ]:
```
# heat map -Correlacion de una variable con otras por columna numerica

corr = df[['latitude', 'longitude', 'price', 'minimum_nights', 'number_of_reviews',
corr

plt.figure(figsize=(8, 6))
sns.heatmap(data=corr, annot=True)
```

Out[ ]: <Axes: >


latitude
longitude
price
minimum_nights
number_of_reviews
reviews_per_month
availability_365
beds
1 0.047 0.013 0.0046 -0.048 -0.042 -0.0059 -0.072
0.047 1 -0.19 0.024 0.0048 0.042 0.064 0.042
0.013 -0.19 1 -0.045 -0.044 -0.013 0.048 0.42
0.0046 0.024 -0.045 1 -0.059 -0.12 0.035 -0.026
-0.048 0.0048 -0.044 -0.059 1 0.63 -0.05 0.04
-0.042 0.042 -0.013 -0.12 0.63 1 -0.04 0.053
-0.0059 0.064 0.048 0.035 -0.05 -0.04 1 0.066
-0.072 0.042 0.42 -0.026 0.04 0.053 0.066 1
1.0
0.8
0.6
0.4
0.2
0.0