Capstone Project Creation

IBM SkillsBuild Europe Delivery - Data Analytics

Pre-requisite

- Understanding of Python, Power BI or Tableau
- Understanding of Data Cleaning
- Understanding Data Visualization

Level of Exercise: Intermediate

Duration: approximately 3 hours

Data Analytics of Airbnb Data:

Objective:

In this exericise, you will be performing Data Analytics on an Open Dataset dataset coming from Airbnb. Some of the tasks include

- Data Cleaning.
- Data Transformation
- Data Visualization.

Overview of Airbnb Data:

People's main criteria when visiting new places are reasonable accommodation and food. Airbnb (Air-Bed-Breakfast) is an online marketplace created to meet this need of people by renting out their homes for a short term. They offer this facility at a relatively lower price than hotels. Further people worldwide prefer the homely and economical service offered by them. They offer services across various geographical locations

Dataset Source

YOu can get the dataset for this assessment using the following link: https://www.kaggle.com/datasets/arianazmoudeh/airbnbopendata (https://www.kaggle.com/datasets/arianazmoudeh/airbnbopendata)

This dataset contains information such as the neighborhood offering these services, room type, price,avaliabilty, reviews, service fee, cancellation policy and rules to use the house. This analysis will help airbnb in improving its services.

So all the best for your Data Analytics Journey on Airbnb data!!!

Task 1: Data Loading (Python)

1. Read the csv file and load it into a pandas dataframe.
2. Display the first five rows of your dataframe.
3. Display the data types of the columns.

In [3]:
```
## Read the csv file
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots

df = pd.read_csv('Airbnb_Open_Data.csv',low_memory=False)
```

In [5]:
```
## Display the first 5 rows
df.head()
```

Out[5]:

	id	NAME	host id	host_identity_verified	host name	neighbourhood group	neighbourhood	lat	long	country	...	service fee	minin ni
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brooklyn	Kensington	40.64749	-73.97237	United States	...	$193	
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhattan	Midtown	40.75362	-73.98377	United States	...	$28	
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhattan	Harlem	40.80902	-73.94190	United States	...	$124	
3	1002755	NaN	85098326012	unconfirmed	Garry	Brooklyn	Clinton Hill	40.68514	-73.95976	United States	...	$74	
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhattan	East Harlem	40.79851	-73.94399	United States	...	$41	

5 rows × 26 columns

In [6]:
```
## Display the data types
df.dtypes
```

Out[6]:
```
id                                int64
NAME                             object
host id                           int64
host_identity_verified           object
host name                        object
neighbourhood group              object
neighbourhood                    object
lat                             float64
long                            float64
country                          object
country code                     object
instant_bookable                 object
cancellation_policy              object
room type                        object
Construction year               float64
price                            object
service fee                      object
minimum nights                  float64
number of reviews               float64
last review                      object
reviews per month               float64
review rate number              float64
calculated host listings count  float64
availability 365                float64
house_rules                      object
license                          object
dtype: object
```

Task 2a: Data Cleaning (Any Tool)

1. Drop some of the unwanted columns. These include `host id`, `id`, `country` and `country code` from the dataset.
2. State the reason for not including these columns for your Data Analytics.

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots before and after the elimination of the columns.

In [7]:
```
df.columns
```

Out[7]:
```
Index(['id', 'NAME', 'host id', 'host_identity_verified', 'host name',
       'neighbourhood group', 'neighbourhood', 'lat', 'long', 'country',
       'country code', 'instant_bookable', 'cancellation_policy', 'room type',
       'Construction year', 'price', 'service fee', 'minimum nights',
       'number of reviews', 'last review', 'reviews per month',
       'review rate number', 'calculated host listings count',
       'availability 365', 'house_rules', 'license'],
      dtype='object')
```

In [8]:
```
df.drop(columns=['id', 'host id','country', 'country code', ], axis=1, inplace=True)
# Reason for dropping  `host id`, `id`, `country` and `country code` columns:
#`id` and `host id` are random ids which doesnt add any value to the dataset, While 'country` and `country code` are having c
```

In [9]:
```
df.dtypes
```

Out[9]:
```
NAME                             object
host_identity_verified           object
host name                        object
neighbourhood group              object
neighbourhood                    object
lat                             float64
long                            float64
instant_bookable                 object
cancellation_policy              object
room type                        object
Construction year               float64
price                            object
service fee                      object
minimum nights                  float64
number of reviews               float64
last review                      object
reviews per month               float64
review rate number              float64
calculated host listings count  float64
availability 365                float64
house_rules                      object
license                          object
dtype: object
```

Task 2b: Data Cleaning (Python)

- Check for missing values in the dataframe and display the count in ascending order. **If the values are missing, impute the values as per the datatype of the columns.**
- Check whether there are any duplicate values in the dataframe and, if present, remove them.
- Display the total number of records in the dataframe before and after removing the duplicates.

In [10]:
```
## Check for missing values in the dataframe and display the count in ascending order.
df.isnull().sum().sort_values()
```

Out[10]:
```
room type                           0
lat                                 8
long                                8
neighbourhood                      16
neighbourhood group                29
cancellation_policy                76
instant_bookable                  105
number of reviews                 183
Construction year                 214
price                             247
NAME                              250
service fee                       273
host_identity_verified            289
calculated host listings count    319
review rate number                326
host name                         406
minimum nights                    409
availability 365                  448
reviews per month               15879
last review                     15893
house_rules                     52131
license                        102597
dtype: int64
```

In [12]:
```
## Check whether there are any duplicate values in the dataframe and if present remove them.
df.shape
```

Out[12]:
```
(102599, 22)
```

In [13]:
```
df.duplicated().sum()
```

Out[13]:
```
3436
```

In [14]:
```
df.drop_duplicates(inplace=True)
```

In [15]:
```
## Display the total number of records in the dataframe after removing the duplicates.
df.shape
```

Out[15]:
```
(99163, 22)
```

Task 3: Data Transformation (Any Tool)

- Rename the column `availability 365` to `days_booked`
- Convert all column names to lowercase and replace the spaces in the column names with an underscore "_".

- Remove the dollar sign and comma from the columns `price` and `service_fee` . If necessary, convert these two columns to the appropriate data type.

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots of your work.

In [17]:
```
## Rename the column.
df.rename(columns={'availability 365':'days_booked'}, inplace=True)
df.head(2)
```

Out[17]:

ong	instant_bookable	cancellation_policy	room type	...	service fee	minimum nights	number of reviews	last review	reviews per month	review rate number	calculated host listings count	days_booked	house_rules	lic
237	False	strict	Private room	...	$193	10.0	9.0	10/19/2021	0.21	4.0	6.0	286.0	Clean up and treat the home the way you'd like...	
377	False	moderate	Entire home/apt	...	$28	30.0	45.0	5/21/2022	0.38	4.0	2.0	228.0	Pet friendly but please confirm with me if the...	

In [18]:
```
## Convert all column names to lowercase and replace the spaces with an underscore "_"
df.columns = [col.lower().replace(' ','_') for col in df.columns]
df.columns
```

Out[18]:
```
Index(['name', 'host_identity_verified', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'lat', 'long', 'instant_bookable',
       'cancellation_policy', 'room_type', 'construction_year', 'price',
       'service_fee', 'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'review_rate_number',
       'calculated_host_listings_count', 'days_booked', 'house_rules',
       'license'],
      dtype='object')
```

In [20]:
```
## Remove the dollar sign and comma from the columns. If necessary, convert these two columns to the appropriate data type.
df[['price','service_fee']].head()
```

Out[20]:

	price	service_fee
0	$966	$193
1	$142	$28
2	$620	$124
3	$368	$74
4	$204	$41

In [28]:
```
def remove_dollar_comma_sign(value):
    if pd.isna(value):
        return np.NaN
    else:
        return float(value.replace( "$","").replace(",",""))
```

In [30]:
```
df['price'] = df['price'].apply(lambda x: remove_dollar_comma_sign(x))
```

In [31]:
```
df['service_fee'] = df['service_fee'].apply(lambda x: remove_dollar_comma_sign(x))
```

In [29]:
```
df[['price', 'service_fee']].head()
```

Out[29]:

	price	service_fee
0	966	193
1	142	28
2	620	124
3	368	74
4	204	41

Task 4: Exploratory Data Analysis (Any Tool)

- List the count of various room types avaliable in the dataset.
- Which room type has the most strict cancellation policy?
- List the average price per neighborhood group, and highlight the most expensive neighborhood to rent from.

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots of your work.

In [32]:
```
## List the count of various room types avaliable with Airbnb
df['room_type'].unique()
```

Out[32]:
```
array(['Private room', 'Entire home/apt', 'Shared room', 'Hotel room'],
      dtype=object)
```

In [33]:
```
df['room_type'].value_counts()
```

Out[33]:
```
room_type
Entire home/apt    52003
Private room       44895
Shared room         2150
Hotel room           115
Name: count, dtype: int64
```

In [34]:
```
df['cancellation_policy'].unique()
```

Out[34]:
```
array(['strict', 'moderate', 'flexible', nan], dtype=object)
```

In [36]:
```
## Which room type adheres to more strict cancellation policy
df_group_prep = df[df['cancellation_policy']=='strict']

df_group_prep.shape
```

Out[36]:
```
(32930, 22)
```

In [37]:
```
## List the prices by neighborhood group and also mention which is the most expensive neighborhood group for rentals
grp_avg = df['price'].groupby(df['neighbourhood_group']).mean().sort_values(ascending=False).reset_index()
grp_avg
```

Out[37]:

	neighbourhood_group	price
0	Queens	629.712735
1	Bronx	626.614412
2	Staten Island	626.431843
3	Brooklyn	626.428192
4	Manhattan	622.683781
5	brookln	580.000000
6	manhatan	460.000000

In [ ]:
```
#Most expensive neighborhood: Queens
```

Task 5a: Data Visualization (Any Tool)

* Create a horizontal bar chart to display the top 10 most expensive neighborhoods in the dataset
* List the neighborhoods which offer short term rentals within 10 days. Illustrate with a bar graph
* List the prices with respect to room type using a bar graph and also state your inferences.
* Create a pie chart that shows distribution of booked days for each neighborhood group

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots of your work.

In [38]:
```
grp2 = df['price'].groupby(df['neighbourhood']).sum().sort_values(ascending=False)
grp2.head(10)
```

Out[38]:
```
neighbourhood
Bedford-Stuyvesant    4782134.0
Williamsburg          4659604.0
Harlem                3316270.0
Bushwick              3035466.0
Hell's Kitchen        2394057.0
Upper West Side       2305160.0
Upper East Side       2175764.0
East Village          2077759.0
Midtown               1984887.0
Crown Heights         1941184.0
Name: price, dtype: float64
```

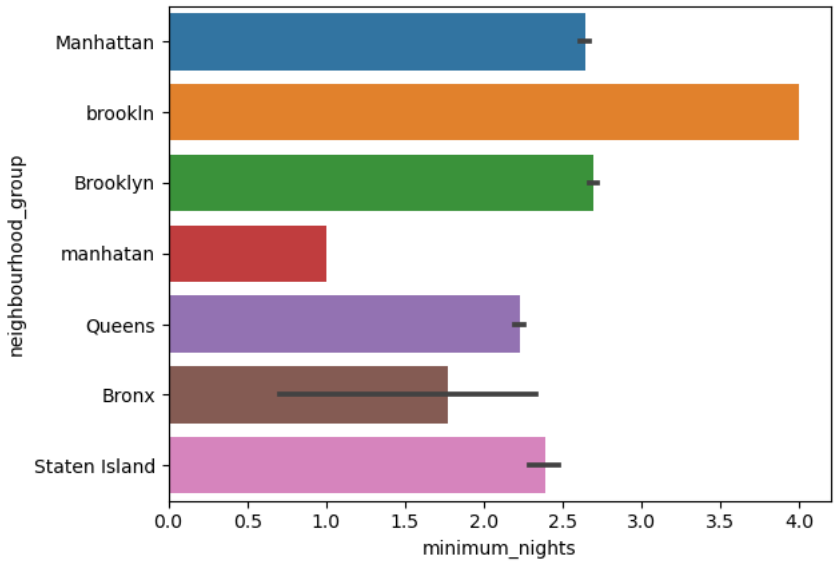
In [39]:
```
grp2.head(10).plot(kind='barh',color={'blue'})
plt.xlabel('Price')
plt.ylabel('Neighbourhood')
plt.title('Top 10 Price vs Neighbourhood')
plt.show
```

Out[39]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

In [40]:
```
# List the neighborhoods which offer short term rentals within 10 days. Illustrate with a bar graph
df_filter_min_nights = df[df['minimum_nights']<10]
df_filter_min_nights['neighbourhood_group'].value_counts()

sns.barplot(x='minimum_nights',
          y='neighbourhood_group',
          data=df_filter_min_nights, orient='h')
```

Out[40]:
```
<Axes: xlabel='minimum_nights', ylabel='neighbourhood_group'>
```



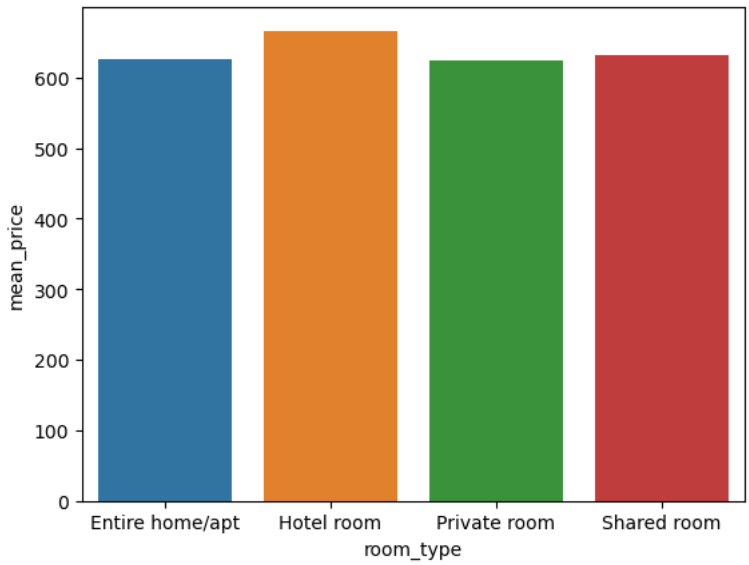
In [42]:
```
# List the prices with respect to room type using a bar graph and also state your inferences.
df1 = df.groupby(['room_type']).agg(mean_price=('price','mean'))
df1 = df1.reset_index()
df1.head()
```

Out[42]:

	room_type	mean_price
0	Entire home/apt	625.263948
1	Hotel room	666.391304
2	Private room	624.818326
3	Shared room	632.439309

In [43]:
```
sns.barplot(x='room_type',
            y='mean_price',
            data=df1)
```

Out[43]: <Axes: xlabel='room_type', ylabel='mean_price'>



In [ ]:
```
#conclusion: Hotel room are more expensive than Airbnb room, and also to entire home/apt
```

In [44]:
```
##Create a pie chart that shows distribution of booked days for each neighborhood group
grp3=df['days_booked'].groupby(df['neighbourhood_group']).mean().sort_values().reset_index()
grp3
```

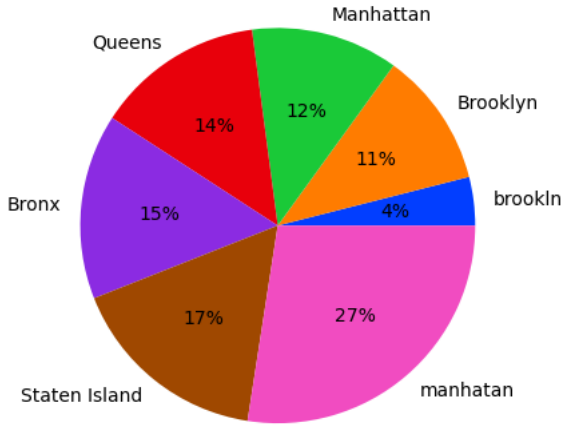
Out[44]:

	neighbourhood_group	days_booked
0	brookln	47.000000
1	Brooklyn	130.906573
2	Manhattan	142.843801
3	Queens	162.745440
4	Bronx	179.502308
5	Staten Island	196.097933
6	manhatan	325.000000

In [45]:
```
palette_color = sns.color_palette('bright')

plt.pie(grp3['days_booked'], labels=grp3['neighbourhood_group'], colors=palette_color, autopct='%.0f%%')
```

Out[45]:
```
([<matplotlib.patches.Wedge at 0x1e5e465f890>,
  <matplotlib.patches.Wedge at 0x1e5e46ec510>,
  <matplotlib.patches.Wedge at 0x1e5e46eda50>,
  <matplotlib.patches.Wedge at 0x1e5e46ed310>,
  <matplotlib.patches.Wedge at 0x1e5e46f8a50>,
  <matplotlib.patches.Wedge at 0x1e5e46fa310>,
  <matplotlib.patches.Wedge at 0x1e5e46fb850>],
 [Text(1.0914587496364725, 0.13681300319044332, 'brookln'),
  Text(0.9099062061951819, 0.618118674630925, 'Brooklyn'),
  Text(0.2697789315732275, 1.0664048612413617, 'Manhattan'),
  Text(-0.5870934200172105, 0.9302264864926688, 'Queens'),
  Text(-1.0945304203431792, 0.10955892909016555, 'Bronx'),
  Text(-0.6865574944118081, -0.8594409850984419, 'Staten Island'),
  Text(0.7157819394971657, -0.8352581727164817, 'manhatan')],
 [Text(0.5953411361653486, 0.07462527446751452, '4%'),
  Text(0.4963124761064628, 0.3371556407077772, '11%'),
  Text(0.1471521444944877, 0.5816753788589245, '12%'),
  Text(-0.3202327745548421, 0.5073962653596374, '14%'),
  Text(-0.5970165929144613, 0.059759415867363025, '15%'),
  Text(-0.3744859060428044, -0.46878599187187736, '17%'),
  Text(0.3904265124529994, -0.4555953669362627, '27%')])
```



Task 5b: Data Visualization (Any Tool)

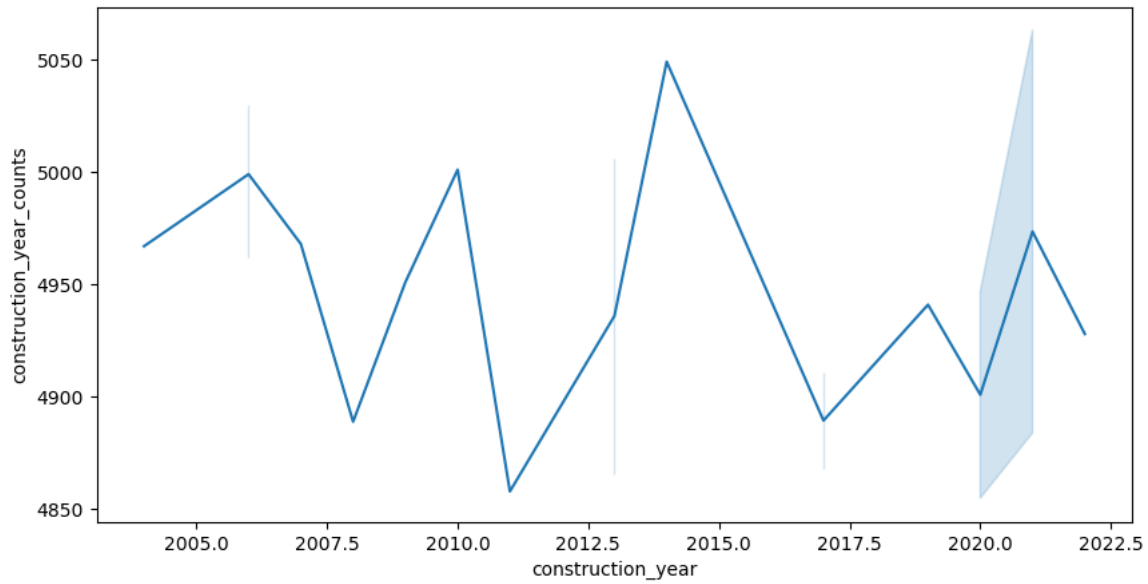
- Does service price and room price have an impact on each other. Illustrate this relationship with a scatter plot and state your inferences
- Using a line graph show in which year the maximum construction of rooms took place.

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots of your work.

In [46]:
```
plt.figure(figsize=(15,10))
plt.title('Relationship bewtween price and service fee', size=25, color='red')
sns.scatterplot(x=df['price'], y=df['service_fee'], hue=df.room_type, s=30)
```

Out[46]: <Axes: title={'center': 'Relationship bewtween price and service fee'}, xlabel='price', ylabel='service_fee'>

In [48]:
```
plt.figure(figsize=(10,5))
df['construction_year_counts']=df['construction_year'].value_counts()
sns.lineplot(x='construction_year', y='construction_year_counts', data=df)
plt.show()
```



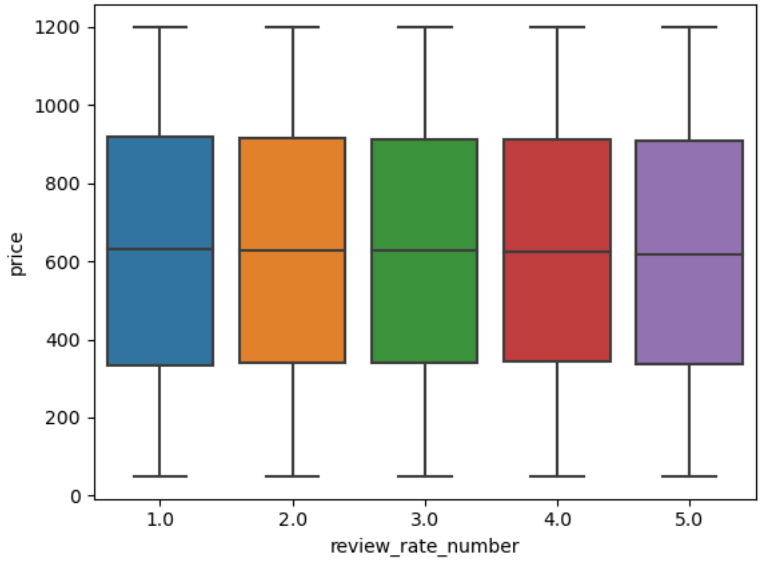
Task 5c: Data Visualization (Any Tool)

- With the help of box plots illustrate the following
- Effect of Review Rate number on price
- Effect of host identity verified on price

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots of your work.

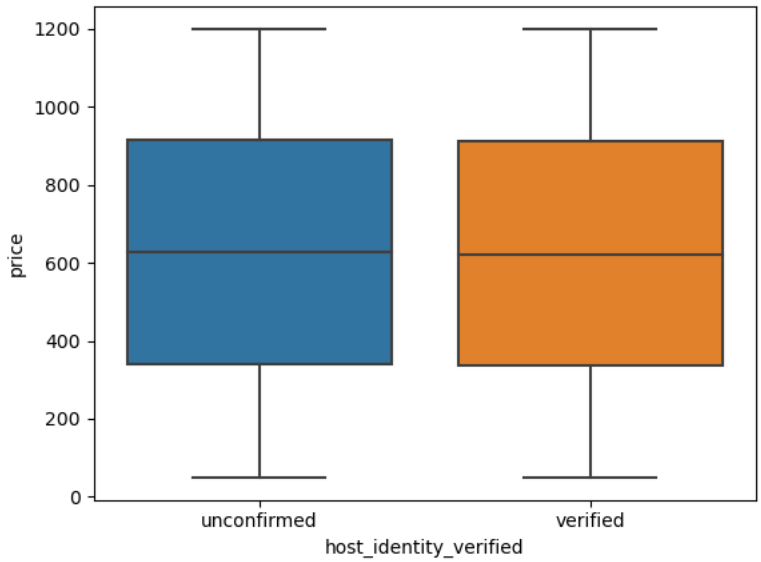
In [49]:
```
sns.boxplot(x='review_rate_number', y='price', data=df)
```

Out[49]: <Axes: xlabel='review_rate_number', ylabel='price'>



In [50]:
```
sns.boxplot(x='host_identity_verified', y='price', data=df)
```

Out[50]: <Axes: xlabel='host_identity_verified', ylabel='price'>



In [ ]: