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

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!!!

In [ ]:
```
### NOTE:
# Two hashtags (##) means added by supervisor;
# One hashtag (#) means added by me.
```

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 [10]:
```
## Read the csv file
import pandas as pd

Airbnb = pd.read_csv('/Users/richardlinderman/Downloads/Airbnb_Open
```

```
/var/folders/kp/7s802n5534x_q57672tm9tm80000gn/T/ipykernel_87074/8
43023976.py:4: DtypeWarning: Columns (25) have mixed types. Specif
y dtype option on import or set low_memory=False.
  Airbnb = pd.read_csv('/Users/richardlinderman/Downloads/Airbnb_O
pen_Data.csv')
```

In [11]:
```
## Display the first 5 rows
Airbnb.head(5)
```

Out[11]:

	id	NAME	host id	host_identity_verified	host name	neighbourhood group
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brooklyn
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhattan
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhattan
3	1002755	NaN	85098326012	unconfirmed	Garry	Brooklyn
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhattan

5 rows × 26 columns

In [12]:
```
## Display the data types
Airbnb.dtypes
```

Out[12]:
```
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 [13]:
```
Airbnb1 = Airbnb.drop(columns=['host id', 'id', 'country', 'country

Airbnb1
```

Out[13]:

	NAME	host_identity_verified	host name	neighbourhood group	neighbourhood
0	Clean & quiet apt home by the park	unconfirmed	Madaline	Brooklyn	Kensington
1	Skylit Midtown Castle	verified	Jenna	Manhattan	Midtown
2	THE VILLAGE OF HARLEM....NEW YORK !	NaN	Elise	Manhattan	Harlem
3	NaN	unconfirmed	Garry	Brooklyn	Clinton Hill
4	Entire Apt: Spacious Studio/Loft by central park	verified	Lyndon	Manhattan	East Harlem
...	...	...	...	...	...
102594	Spare room in Williamsburg	verified	Krik	Brooklyn	Williamsburg
102595	Best Location near Columbia U	unconfirmed	Mifan	Manhattan	Morningside Heights
102596	Comfy, bright room in Brooklyn	unconfirmed	Megan	Brooklyn	Park Slope
102597	Big Studio-One Stop from Midtown	unconfirmed	Christopher	Queens	Long Island City
102598	585 sf Luxury Studio	unconfirmed	Rebecca	Manhattan	Upper West Side

102599 rows × 22 columns

In [14]:
```
# host id and id were dropped because both, despite being numbers a
```

In [ ]:
```

```

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 [15]:
```
## Check for missing values in the dataframe and display the count
Airbnb1.isnull()

Airbnb2 = Airbnb1.isnull()
Airbnb3 = Airbnb2.sum()

Airbnb3.sort_values(ascending=True)
```

Out[15]:
```
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 [16]:
```
Airbnb1['lat'] = Airbnb1['lat'].fillna(Airbnb1['lat'].mode()[0])
Airbnb1['long'] = Airbnb1['long'].fillna(Airbnb1['long'].mode()[0])
Airbnb1['neighbourhood'] = Airbnb1['neighbourhood'].fillna(Airbnb1[
Airbnb1['neighbourhood group'] = Airbnb1['neighbourhood group'].fil
Airbnb1['cancellation_policy'] = Airbnb1['cancellation_policy'].fil
Airbnb1['instant_bookable'] = Airbnb1['instant_bookable'].fillna(Ai
Airbnb1['number of reviews'] = Airbnb1['number of reviews'].fillna(
Airbnb1['Construction year'] = Airbnb1['Construction year'].fillna(
Airbnb1['price'] = Airbnb1['price'].fillna(Airbnb1['price'].mode()[
Airbnb1['NAME'] = Airbnb1['NAME'].fillna(Airbnb1['NAME'].mode()[0])
Airbnb1['service fee'] = Airbnb1['service fee'].fillna(Airbnb1['ser
Airbnb1['host_identity_verified'] = Airbnb1['host_identity_verified
Airbnb1['calculated host listings count'] = Airbnb1['calculated hos
Airbnb1['review rate number'] = Airbnb1['review rate number'].filln
Airbnb1['host name'] = Airbnb1['host name'].fillna(Airbnb1['host na
Airbnb1['minimum nights'] = Airbnb1['minimum nights'].fillna(Airbnb
Airbnb1['availability 365'] = Airbnb1['availability 365'].fillna(Ai
Airbnb1['reviews per month'] = Airbnb1['reviews per month'].fillna(
Airbnb1['last review'] = Airbnb1['last review'].fillna(Airbnb1['las
Airbnb1['house_rules'] = Airbnb1['house_rules'].fillna(Airbnb1['hou
Airbnb1['license'] = Airbnb1['license'].fillna(Airbnb1['license'].m
```

In [17]:
```
## Check whether there are any duplicate values in the dataframe an
Airbnb1.duplicated()
```

Out[17]:
```
0         False
1         False
2         False
3         False
4         False
          ...
102594     True
102595     True
102596     True
102597     True
102598     True
Length: 102599, dtype: bool
```

In [18]:
```
Airbnb4 = Airbnb1.drop_duplicates()
Airbnb4
```

Out[18]:

	NAME	host_identity_verified	host name	neighbourhood group	neighbourhood	
0	Clean & quiet apt home by the park	unconfirmed	Madaline	Brooklyn	Kensington	40

1	Skylit Midtown Castle	verified	Jenna	Manhattan	Midtown	40	
2	THE VILLAGE OF HARLEM....NEW YORK !	unconfirmed	Elise	Manhattan	Harlem	40	
3	Home away from home	unconfirmed	Garry	Brooklyn	Clinton Hill	40	
4	Entire Apt: Spacious Studio/Loft by central park	verified	Lyndon	Manhattan	East Harlem	40	
...	...	...	...	...	...	...	
102053	Cozy bright room near Prospect Park	unconfirmed	Mariam	Brooklyn	Flatbush	40	
102054	Private Bedroom with Amazing Rooftop View	verified	Trey	Brooklyn	Bushwick	40	
102055	Pretty Brooklyn One-Bedroom for 2 to 4 people	verified	Michael	Brooklyn	Bedford-Stuyvesant	40	
102056	Room & private bathroom in historic Harlem	unconfirmed	Shireen	Manhattan	Harlem	40	
102057	Rosalee Stewart	verified	Stanley	Manhattan	Harlem	40	

99137 rows × 22 columns

In [19]:
```
## Display the total number of records in the dataframe after remov
len(Airbnb4.index)
```

Out[19]: 99137

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 [20]:
```
## Rename the column.
Airbnb4.rename(columns={"availability 365": "days_booked"})
```

Out[20]:

	NAME	host_identity_verified	host name	neighbourhood group	neighbourhood	
0	Clean & quiet apt home by the park	unconfirmed	Madaline	Brooklyn	Kensington	40
1	Skylit Midtown Castle	verified	Jenna	Manhattan	Midtown	40
2	THE VILLAGE OF HARLEM....NEW YORK !	unconfirmed	Elise	Manhattan	Harlem	40
3	Home away from home	unconfirmed	Garry	Brooklyn	Clinton Hill	40
4	Entire Apt: Spacious Studio/Loft by central park	verified	Lyndon	Manhattan	East Harlem	40
...	...	...	...	...	...	
102053	Cozy bright room near Prospect Park	unconfirmed	Mariam	Brooklyn	Flatbush	40
102054	Private Bedroom with Amazing Rooftop View	verified	Trey	Brooklyn	Bushwick	40

102055	Pretty Brooklyn One-Bedroom for 2 to 4 people		verified	Michael	Brooklyn	Bedford-Stuyvesant	40
102056	Room & private bathroom in historic Harlem		unconfirmed	Shireen	Manhattan	Harlem	40
102057	Rosalee Stewart		verified	Stanley	Manhattan	Harlem	40

99137 rows × 22 columns

In [21]:
```
## Convert all column names to lowercase and replace the spaces wit
Airbnb4.columns = Airbnb4.columns.str.lower()
Airbnb4.columns = Airbnb4.columns.str.replace(" ", "_")
Airbnb4
```

Out[21]:

	name	host_identity_verified	host_name	neighbourhood_group	neighbourl
0	Clean & quiet apt home by the park	unconfirmed	Madaline	Brooklyn	Kensir
1	Skylit Midtown Castle	verified	Jenna	Manhattan	Mid
2	THE VILLAGE OF HARLEM....NEW YORK !	unconfirmed	Elise	Manhattan	Ha
3	Home away from home	unconfirmed	Garry	Brooklyn	Clinto
4	Entire Apt: Spacious Studio/Loft by central park	verified	Lyndon	Manhattan	East Ha
...	...	...	...	...	
102053	Cozy bright room near Prospect Park	unconfirmed	Mariam	Brooklyn	Flat
102054	Private Bedroom with Amazing Rooftop View	verified	Trey	Brooklyn	Bush

102055	Pretty Brooklyn One-Bedroom for 2 to 4 people	verified	Michael	Brooklyn	Bec Stuyv
102056	Room & private bathroom in historic Harlem	unconfirmed	Shireen	Manhattan	Ha
102057	Rosalee Stewart	verified	Stanley	Manhattan	Ha

99137 rows × 22 columns

In [22]:
```
## Remove the dollar sign and comma from the columns. If necessary,
Airbnb4['price'] = Airbnb4['price'].str.replace('$', '')
Airbnb4['price'] = Airbnb4['price'].str.replace(',', '')
Airbnb4['service_fee'] = Airbnb4['service_fee'].str.replace('$', ''
Airbnb4['service_fee'] = Airbnb4['service_fee'].str.replace(',', ''
```

```
/var/folders/kp/7s802n5534x_q57672tm9tm80000gn/T/ipykernel_87074/1
377424481.py:2: FutureWarning: The default value of regex will cha
nge from True to False in a future version. In addition, single ch
aracter regular expressions will *not* be treated as literal strin
gs when regex=True.
  Airbnb4['price'] = Airbnb4['price'].str.replace('$', '')
/var/folders/kp/7s802n5534x_q57672tm9tm80000gn/T/ipykernel_87074/1
377424481.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy)
  Airbnb4['price'] = Airbnb4['price'].str.replace('$', '')
/var/folders/kp/7s802n5534x_q57672tm9tm80000gn/T/ipykernel_87074/1
377424481.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy)
  Airbnb4['price'] = Airbnb4['price'].str.replace(',', '')
/var/folders/kp/7s802n5534x_q57672tm9tm80000gn/T/ipykernel_87074/1
377424481.py:4: FutureWarning: The default value of regex will cha
nge from True to False in a future version. In addition, single ch
aracter regular expressions will *not* be treated as literal strin
```

```
gs when regex=True.
  Airbnb4['service_fee'] = Airbnb4['service_fee'].str.replace('$',
'')
/var/folders/kp/7s802n5534x_q57672tm9tm80000gn/T/ipykernel_87074/1
377424481.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy)
  Airbnb4['service_fee'] = Airbnb4['service_fee'].str.replace('$',
'')
/var/folders/kp/7s802n5534x_q57672tm9tm80000gn/T/ipykernel_87074/1
377424481.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy)
  Airbnb4['service_fee'] = Airbnb4['service_fee'].str.replace(',',
'')
```

In [23]:
```
def convert_column_to_float(df, column_name):
    df[column_name] = df[column_name].astype(float)
    return df
```

In [24]:
```
Airbnb4 = convert_column_to_float(Airbnb4, 'price')
```

```
/var/folders/kp/7s802n5534x_q57672tm9tm80000gn/T/ipykernel_87074/3
929838235.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy)
  df[column_name] = df[column_name].astype(float)
```

In [25]:
```
Airbnb4 = convert_column_to_float(Airbnb4, 'service_fee')
```

```
/var/folders/kp/7s802n5534x_q57672tm9tm80000gn/T/ipykernel_87074/3
929838235.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy)
  df[column_name] = df[column_name].astype(float)
```

In [26]:
```
Airbnb4.dtypes
```

Out[26]:
```
name                              object
host_identity_verified            object
host_name                         object
neighbourhood_group               object
neighbourhood                     object
lat                              float64
long                             float64
instant_bookable                    bool
cancellation_policy               object
room_type                         object
construction_year                float64
price                            float64
service_fee                      float64
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 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 [27]:
```
## List the count of various room types avaliable with Airbnb
Airbnb4['room_type'].nunique()
```

Out[27]: 4

In [28]:
```
## Which room type adheres to more strict cancellation policy
Airbnb5 = Airbnb4[['room_type','cancellation_policy']]
Airbnb5['Total'] = 1
Airbnb5.groupby(['room_type','cancellation_policy']).sum()
```

```
/var/folders/kp/7s802n5534x_q57672tm9tm80000gn/T/ipykernel_87074/2
135813906.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/in
dexing.html#returning-a-view-versus-a-copy)
  Airbnb5['Total'] = 1
```

Out[28]:

		Total
room_type	**cancellation_policy**	
Entire home/apt	**flexible**	17360
	moderate	17389
	strict	17239
Hotel room	**flexible**	44
	moderate	37
	strict	34
Private room	**flexible**	14829
	moderate	15120
	strict	14936
Shared room	**flexible**	714
	moderate	717
	strict	718

In [29]:
```
# As seen above, Private rooms are the most strict on cancellations
```

In [30]:
```
## List the prices by neighborhood group and also mention which is t
Airbnb6 = Airbnb4[['neighbourhood_group','price']]
Airbnb6
```

Out[30]:

	neighbourhood_group	price
0	Brooklyn	966.0
1	Manhattan	142.0
2	Manhattan	620.0
3	Brooklyn	368.0
4	Manhattan	204.0
...	...	...
102053	Brooklyn	696.0
102054	Brooklyn	909.0
102055	Brooklyn	387.0
102056	Manhattan	848.0
102057	Manhattan	1128.0

99137 rows × 2 columns

In [66]:
```
Airbnb7 = Airbnb6.groupby(['neighbourhood_group']).mean()
Airbnb7
```

Out[66]:

	price
neighbourhood_group	
Bronx	625.271511
Brooklyn	625.471627
Manhattan	621.649486
Queens	628.642929
Staten Island	625.060870
brookln	580.000000
manhatan	460.000000

In [78]:
```
def highlight_row(row):
    return ['background-color: yellow' if row['price'] == Airbnb7['
```

In [79]:
```
highlighted_Airbnb = Airbnb7.style.apply(highlight_row, axis=1)
```

In [80]:
```
Airbnb6['price'].max()
```

Out[80]: 1200.0

In [81]:
```
highlighted_Airbnb
```

Out[81]:

neighbourhood_group	price
Bronx	625.271511
Brooklyn	625.471627
Manhattan	621.649486
Queens	628.642929
Staten Island	625.060870
brookln	580.000000
manhatan	460.000000

In [ ]:
```
# Here, we can see that the most expensive neighbourhood to rent fr
```

Task 5a: Data Visualization (Any Tool)

- Create a horizontal bar chart to display the top 10 most expensive neighborhoods in the dataset.
  - Create another chart with the 10 cheapest neighborhoods in the dataset.
- Create a box and whisker chart that showcases the price distribution of all listings split by room 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.

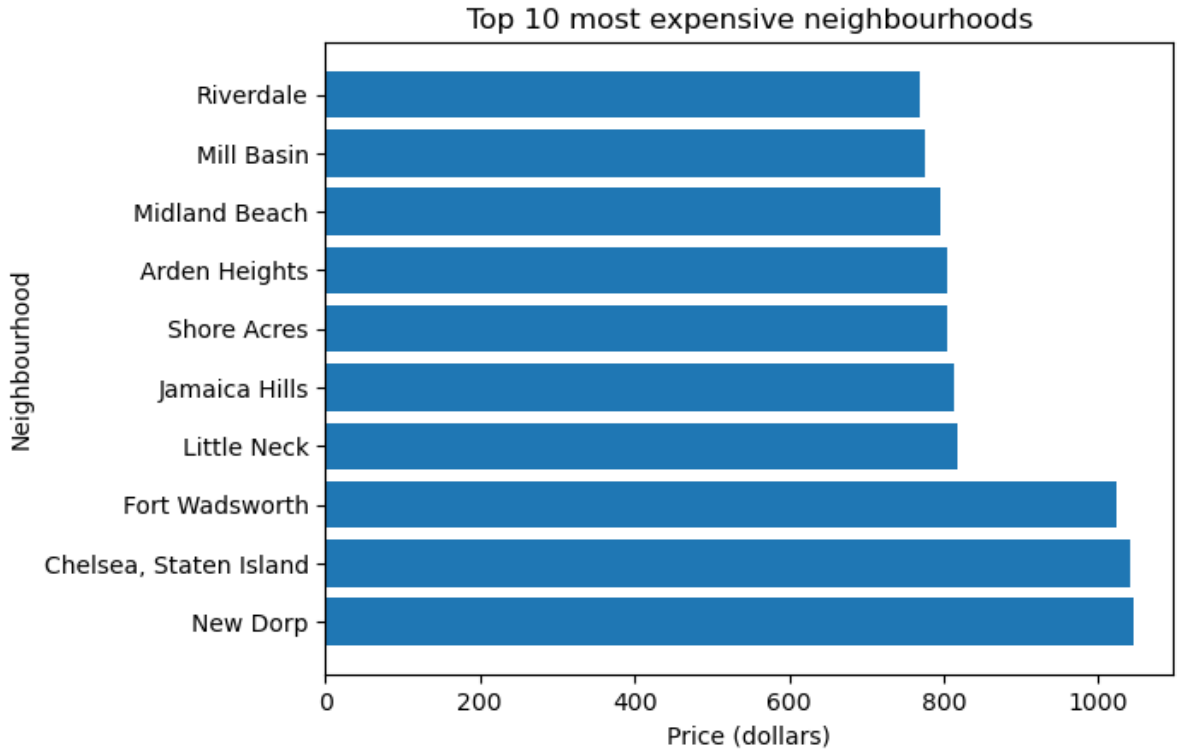
In [87]:
```
Airbnb8 = Airbnb4[['neighbourhood','price']]
Airbnb8 = Airbnb8.groupby(['neighbourhood']).mean()
Airbnb8 = Airbnb8.sort_values(by='price', ascending=False)
Airbnb8 = Airbnb8.head(10)
Airbnb8 = Airbnb8.reset_index()
Airbnb8
```

Out[87]:

	neighbourhood	price
0	New Dorp	1045.333333
1	Chelsea, Staten Island	1042.000000
2	Fort Wadsworth	1024.000000
3	Little Neck	817.750000
4	Jamaica Hills	812.904762
5	Shore Acres	805.142857
6	Arden Heights	804.888889
7	Midland Beach	796.176471
8	Mill Basin	775.142857
9	Riverdale	768.736842

In [88]:
```
import matplotlib.pyplot as plt

plt.barh(Airbnb8['neighbourhood'], Airbnb8['price'])
plt.title('Top 10 most expensive neighbourhoods')
plt.ylabel('Neighbourhood')
plt.xlabel('Price (dollars)')
plt.show()
```



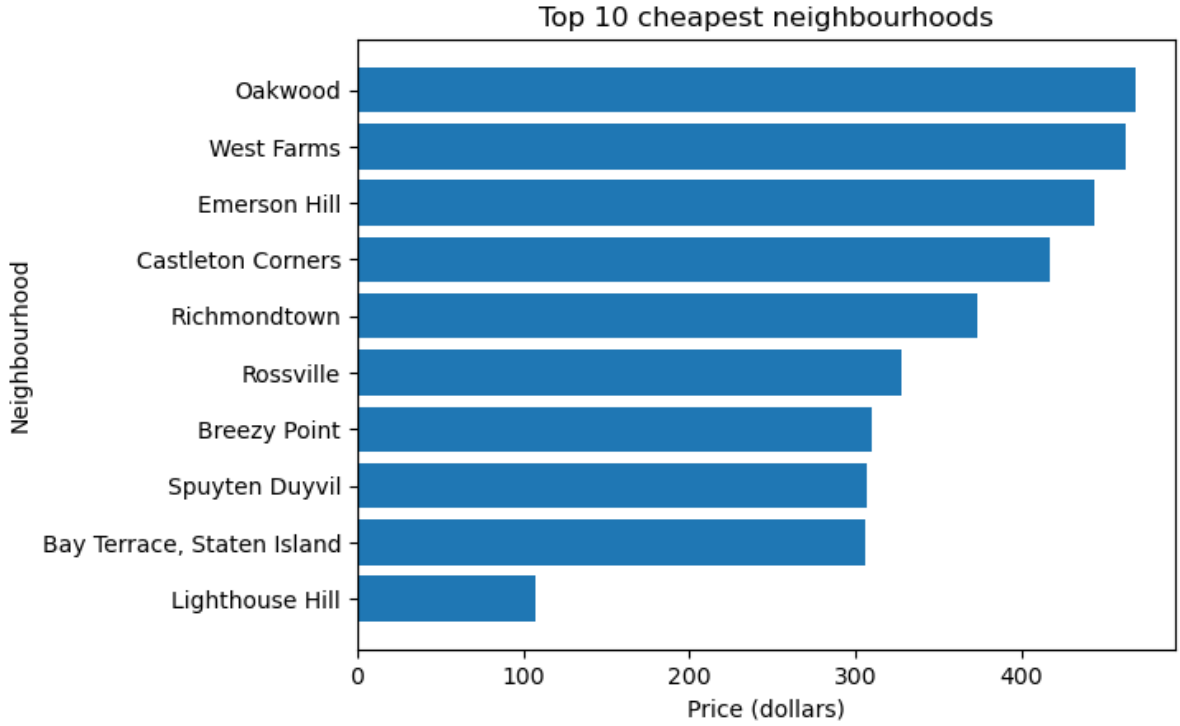
In [89]:
```
# The most expensive neighbourhood on average is New Dorp at just o
```

In [90]:
```
Airbnb9 = Airbnb4[['neighbourhood','price']]
Airbnb9 = Airbnb9.groupby(['neighbourhood']).mean()
Airbnb9 = Airbnb9.sort_values(by='price', ascending=True)
Airbnb9 = Airbnb9.head(10)
Airbnb9 = Airbnb9.reset_index()
Airbnb9
```

Out[90]:

	neighbourhood	price
0	Lighthouse Hill	107.666667
1	Bay Terrace, Staten Island	306.000000
2	Spuyten Duyvil	307.000000
3	Breezy Point	309.888889
4	Rossville	327.500000
5	Richmondtown	373.400000
6	Castleton Corners	417.230769
7	Emerson Hill	443.800000
8	West Farms	463.166667
9	Oakwood	469.307692

In [91]:
```
plt.barh(Airbnb9['neighbourhood'], Airbnb9['price'])
plt.title('Top 10 cheapest neighbourhoods')
plt.ylabel('Neighbourhood')
plt.xlabel('Price (dollars)')
plt.show()
```



In [ ]:
```
# Meanwhile, the cheapest neighbourhood on average is Lighthouse Hi
```

In [36]:
```
Airbnb10 = Airbnb4[['room_type','price']]
Airbnb10
```

Out[36]:

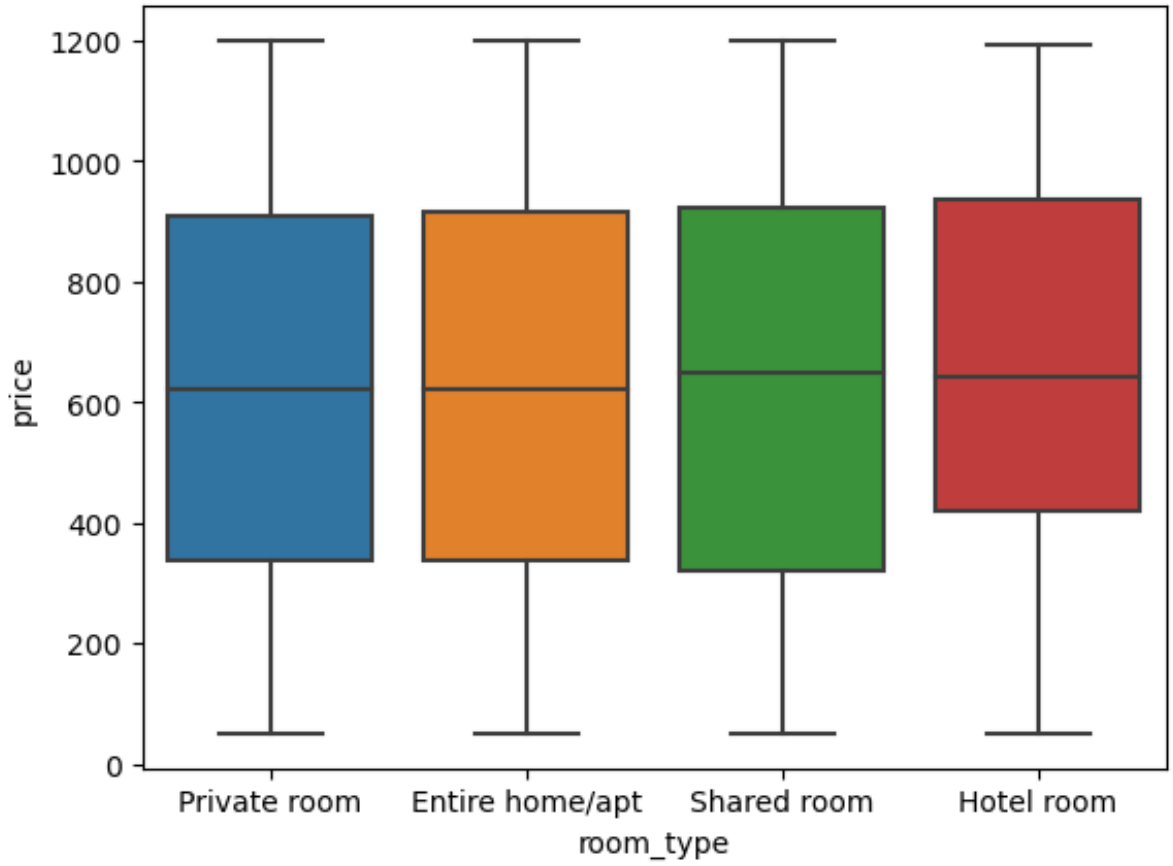
	room_type	price
0	Private room	966.0
1	Entire home/apt	142.0
2	Private room	620.0
3	Entire home/apt	368.0
4	Entire home/apt	204.0
...	...	...
102053	Private room	696.0
102054	Private room	909.0
102055	Entire home/apt	387.0
102056	Private room	848.0
102057	Entire home/apt	1128.0

99137 rows × 2 columns

In [37]:
```
import seaborn as sb

sb.boxplot( x = 'room_type',y = 'price', data = Airbnb10 )
```

Out[37]: <Axes: xlabel='room_type', ylabel='price'>



In [ ]:
```
# While the different room types look roughly the same in spread, t
# higher price than private rooms or apartments based on their medi
# cheap than other room types.
```

Task 5b: Data Visualization (Any Tool)

- Create a scatter plot to illustrate the relationshi between the cleaning fee and the room price and write down the kind of correlation, if any, that you see.
- Create a line chart to showcase the total amount of listings available per year.

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 [41]:
```
Airbnb11 = Airbnb4[['service_fee','price']]
Airbnb11 = Airbnb11.dropna()
Airbnb11
```

Out[41]:

	service_fee	price
0	193.0	966.0
1	28.0	142.0
2	124.0	620.0
3	74.0	368.0
4	41.0	204.0
...	...	...
102053	41.0	696.0
102054	41.0	909.0
102055	41.0	387.0
102056	41.0	848.0
102057	41.0	1128.0

99137 rows × 2 columns

In [42]:
```
plt.scatter(Airbnb11['service_fee'], Airbnb11['price'])
plt.title('Service fee versus price')
plt.ylabel('Service fee')
plt.xlabel('Price (dollars)')
plt.show()
```

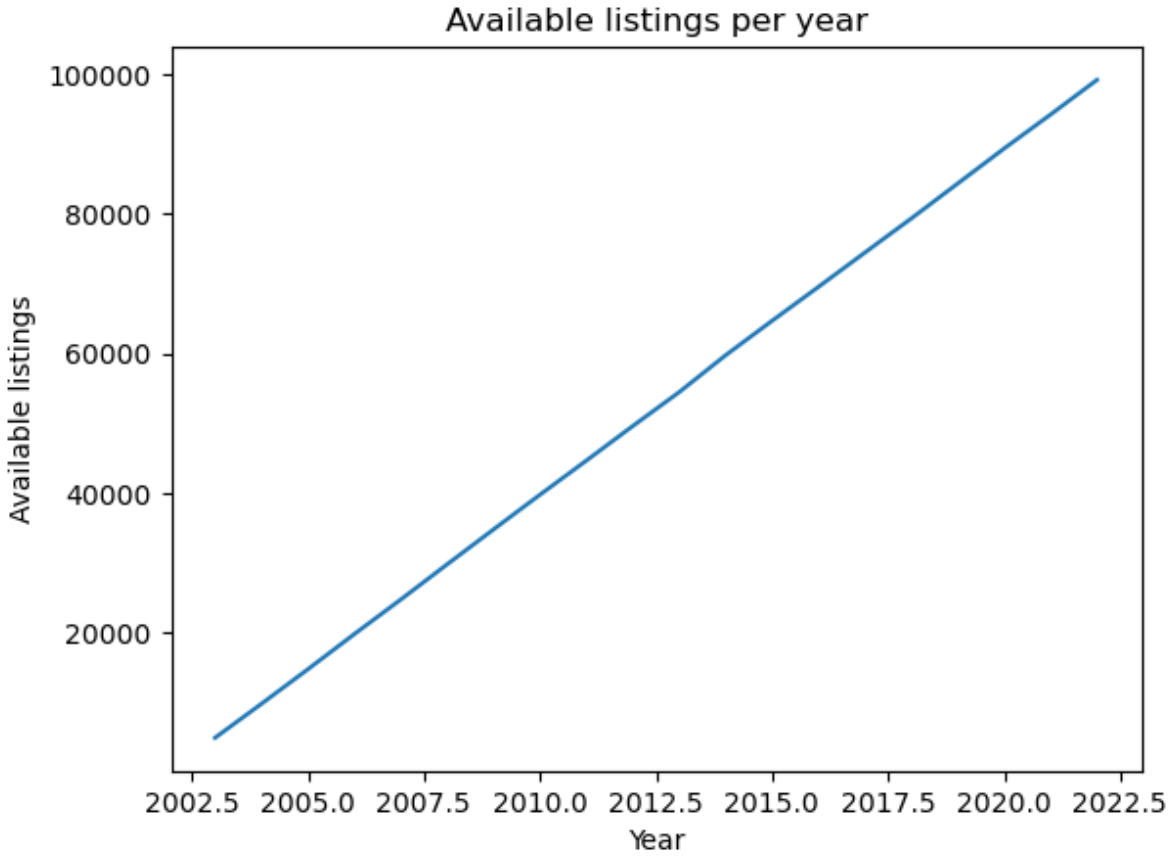
In [43]:
```
# Ignoring the modes, which represent vertical and horizontal lines
# strong positive correlation between service fee and price.
```

In [44]:
```
Airbnb12 = Airbnb4['construction_year']
Airbnb12 = Airbnb12.value_counts()
Airbnb12 = Airbnb12.reset_index()
Airbnb12 = Airbnb12.sort_values(by='index', ascending=True)
Airbnb12 = Airbnb12.reset_index()
Airbnb12 = Airbnb12.drop('level_0', axis=1)
Airbnb12['cum_year'] = Airbnb12['construction_year'].cumsum()
Airbnb12
```

Out[44]:

	index	construction_year	cum_year
0	2003.0	4966	4966
1	2004.0	4864	9830
2	2005.0	4949	14779
3	2006.0	5029	19808
4	2007.0	4941	24749
5	2008.0	5046	29795
6	2009.0	4999	34794
7	2010.0	4966	39760
8	2011.0	4887	44647
9	2012.0	4945	49592
10	2013.0	4855	54447
11	2014.0	5277	59724
12	2015.0	4927	64651
13	2016.0	4858	69509
14	2017.0	4911	74420
15	2018.0	4883	79303
16	2019.0	5004	84307
17	2020.0	5005	89312
18	2021.0	4866	94178
19	2022.0	4959	99137

In [45]:
```
plt.plot(Airbnb12['index'], Airbnb12['cum_year'])
plt.title('Available listings per year')
plt.xlabel('Year')
plt.ylabel('Available listings')
plt.show()
```



In [46]:
```
# As we can see above, the amount of available listings went up at
```

Task 5c: Data Visualization (Any Tool)

- Create a data visualization of your choosing using one of the review columns in isolation or in combination with another column.
- Create a visualization to compare at least two different variables between super hosts and regular hosts.

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.

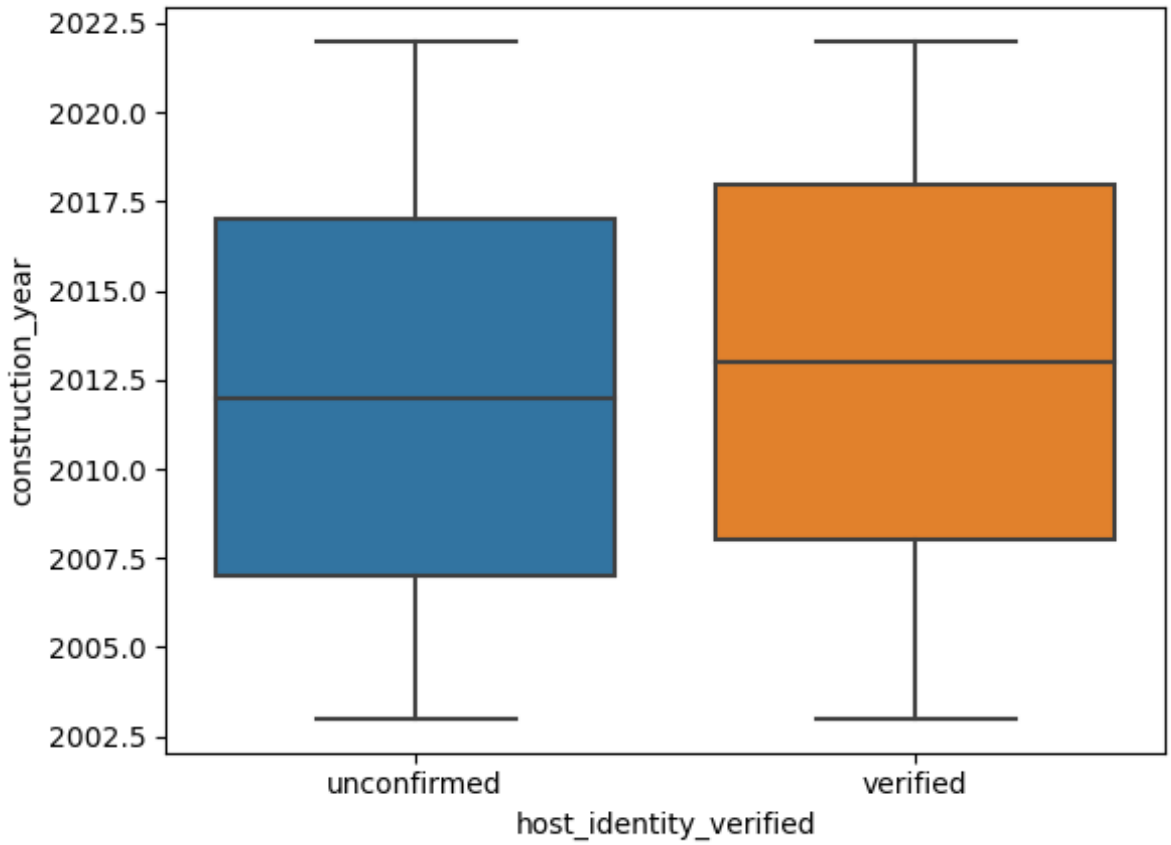
In [47]:
```
Airbnb13 = Airbnb4[['price','reviews_per_month']]
Airbnb13 = Airbnb13.dropna()
Airbnb13

plt.scatter(Airbnb13['price'], Airbnb13['reviews_per_month'])
plt.title('Nights spent at residence, depending on price')
plt.xlabel('Minimum number of nights spent at residence')
plt.ylabel('Price ($)')
plt.show()
```

In [48]:
```
Airbnb14 = Airbnb4[['construction_year','host_identity_verified']]
Airbnb14 = Airbnb14.dropna()

sb.boxplot( x = 'host_identity_verified',y = 'construction_year', d
```

Out[48]:
```
<Axes: xlabel='host_identity_verified', ylabel='construction_year'
>
```



In [49]:
```
# As we can see above, though the median year of construction was r
# verified and those unconfirmed, for the interquartile range it se
# established more recently than unverified hotels.
```

In [ ]:
```

```