

Welcome to the Airbnb Mini Practice Project

Throughout this unit on Python Data Analytics, you've been introduced to the following powerful libraries:

- Matplotlib
- Seaborn
- Pandas

Each of these libraries will enhance your data analysis capabilities.

We've created this challenging exercise to reinforce your understanding of how these libraries work.

Please note, there is a particular emphasis on the Pandas Library. This is the most critical Python library for data analytics. You'll see many similarities between Pandas and Pivot Tables!

The most important thing you can do to build confidence with Python is to practice programming, all the time. This way you will build muscle memory. Don't simply copy the code you've written previously. Write it again and again so you store it in your memory.

As this is a practice exercise, we've included a copy of what the outputs *should* look like for the majority of the questions to give you some guidance.

Time to get started!

Import the airbnb_2.csv file.

Once you do this, you can start your analysis.

Don't forget to import the libraries you need to read .csv files!

Step 1: Import Libraries

Put your code in the box below.

```
In [1]: import matplotlib.pyplot as plt  
       import seaborn as sns
```

```
import pandas as pd
```

Step 2: Import the Airbnb Dataset

Now that you have the Pandas Libraries imported, it's time to import the Airbnb dataset.

i) Import the Airbnb dataset.

ii) Use `.info()` function to better understand the variables inside your dataset.

Put your code in the box below

```
In [2]: airbnb = pd.read_csv('airbnb_2.csv')
airbnb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
 ---  --  
 0   id               48895 non-null   int64  
 1   name              48879 non-null   object  
 2   host_id            48895 non-null   int64  
 3   host_name           48874 non-null   object  
 4   neighbourhood_group 48895 non-null   object  
 5   neighbourhood        48895 non-null   object  
 6   latitude             48895 non-null   float64 
 7   longitude            48895 non-null   float64 
 8   room_type            48895 non-null   object  
 9   price                48895 non-null   int64  
 10  minimum_nights       48895 non-null   int64  
 11  number_of_reviews     48895 non-null   int64  
 12  last_review           38843 non-null   object  
 13  reviews_per_month      38843 non-null   float64 
 14  calculated_host_listings_count 48895 non-null   int64  
 15  availability_365       48895 non-null   int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

```
In [ ]:
```

Step 3: Exploring your data with Pandas

The rest of these questions will have you focus on using the following Pandas Skills:

- Subsetting a Pandas DataFrame using [] and boolean operators
- Summing up records with value_counts()
- Creating calculated fields
- Group By in Pandas
- Creating Bar Plots with Matplotlib

i) Please count how many Airbnb listings are in each of the 5 Neighbourhood Groups (Manhattan, Brooklyn, Queens, Bronx, Staten Island), then identify which Neighbourhood Groups have the greatest number of Airbnb listings.

Put your code in the box below

```
In [3]: airbnb['neighbourhood_group'].value_counts()
```

```
Out[3]: Manhattan      21661
Brooklyn       20104
Queens         5666
Bronx          1091
Staten Island   373
Name: neighbourhood_group, dtype: int64
```

We want to focus our attention on the 3 most popular Neighbourhood Groups, by listing volume.

ii) Calculate the percentage of Airbnb listings that each Neighbourhood Group contains.

See this resource for more details .

Put your code in the box below.

```
In [4]: airbnb['neighbourhood_group'].value_counts(normalize=True)
```

```
Out[4]: Manhattan      0.443011
Brooklyn       0.411167
Queens         0.115881
Bronx          0.022313
Staten Island   0.007629
Name: neighbourhood_group, dtype: float64
```

iii) Create a new calculated field called Revenue and place this into the Airbnb DataFrame. This is to be calculated by using the Price Column x Number_Of_Reviews Columns

Put your code in the box below

```
In [6]: airbnb['Revenue']=airbnb['price']*airbnb['number_of_reviews']
print(airbnb.columns)
```

```
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', 'Revenue'],
      dtype='object')
```

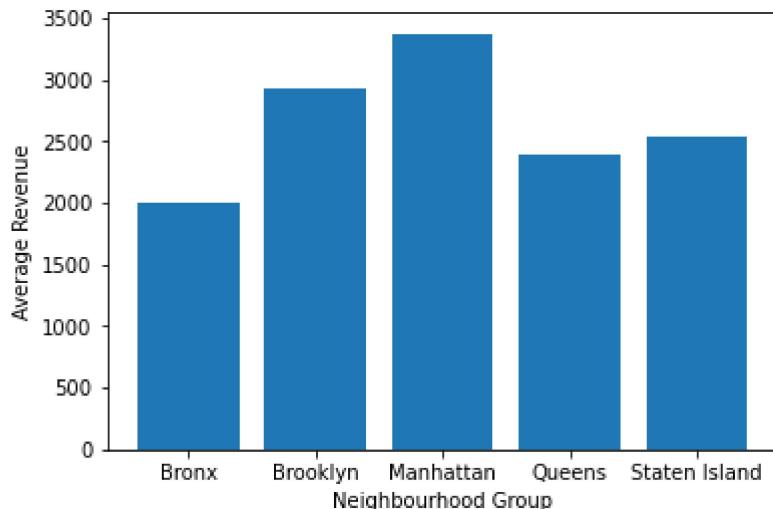
iv) Create a Bar Plot that shows which Neighbourhood Group has the highest average revenues. In order to best calculate this, you'd want to consider how you can use the .groupby() syntax to assist you!

If you're stuck, we recommend you go back to this [DataCamp link](#). Specifically, Chapter 4 which covers how GROUP BY is used in Pandas.

Put your code in the box below.

```
In [8]: bar_airbnb=airbnb.groupby('neighbourhood_group')['Revenue'].mean()
fig, ax = plt.subplots()
print(bar_airbnb)
ax.bar(bar_airbnb.index,bar_airbnb)
ax.set_xlabel('Neighbourhood Group')
ax.set_ylabel('Average Revenue')
plt.show()
```

```
neighbourhood_group
Bronx           1999.439963
Brooklyn        2927.567400
Manhattan       3375.960574
Queens          2400.151077
Staten Island   2541.764075
Name: Revenue, dtype: float64
```



Challenge Questions

V) Filter the Airbnb DataFrame to include only the Neighbourhood Groups Manhattan, Brooklyn, and Queens.

Then, identify the top 3 Revenue Generating Neighborhoods within each of the 3 Neighbourhood_Groups. This should give us 9 Overall Rows: 3 of the top generating neighbourhoods within each of the 3 Neighbourhood_Groups

This is a tricky question that will test your group-by skills.

We recommend you consider the following:

```
condition1 = someDataFrame[ 'someColumn']=='someCondition'  
condition2 = someDataFrame[ 'someColumn']=='someCondition'
```

Step One - Filter the DataFrame using the Conditions

```
filtered_dataframe = someDataFrame[condition1 OR condition 2]
```

```
#Hint: You might want to look up what the OR symbol in Python is represented as in operator form (i.e. AND  
(& )
```

Step Two - Group the Data by Neighbourhood_Group and Neighbourhood. Don't forget you're looking to SUM up the Revenues.

The remaining steps we recommend you think very carefully about.

You might want to make use of the `.reset_index(inplace=True)` function to help reset the indexes in your Grouped Up Dataframe...!

Put your code in the box below.

```
In [12]: Manhattan=airbnb['neighbourhood_group']=='Manhattan'  
Brooklyn=airbnb['neighbourhood_group']=='Brooklyn'  
Queens=airbnb['neighbourhood_group']=='Queens'  
filtered_df=airbnb[Manhattan | Brooklyn | Queens]  
filtered_byneighbourhood=filtered_df.groupby(['neighbourhood_group','neighbourhood'])['Revenue'].sum()  
a=filtered_byneighbourhood.sort_values(ascending=False).reset_index()  
a.groupby('neighbourhood_group').head(3)
```

```
Out[12]:
```

	neighbourhood_group	neighbourhood	Revenue
0	Brooklyn	Williamsburg	12389011
1	Brooklyn	Bedford-Stuyvesant	12352457
2	Manhattan	Harlem	8598692
3	Manhattan	Hell's Kitchen	8238991
4	Manhattan	East Village	7574535
8	Brooklyn	Bushwick	4762224
17	Queens	Astoria	1880840
28	Queens	Long Island City	1374945
33	Queens	Flushing	1140450

VI) Filter the Airbnb Dataframe to include only the top 3 Neighbroos within each neighbourhood_group.

After doing this, identify the top average revenue-generating room-type for each of the nine neighbourhoods and plot this in a Bar Chart.

This is a tricky question that will test your group-by skills. Think back to the previous question and how you approached this; you can approach this in a similar manner.

We recommend you consider the following:

```
condition1 = someDataFrame[ 'someColumn' ]=='someCondition'  
condition2 = someDataFrame[ 'someColumn' ]=='someCondition'
```

Step One - Filter the Dataframe using the Conditions

```
filtered_dataframe = someDataFrame[condition1 OR condition 2]
```

#Hint: You might want to look up what the OR symbol in Python is represented as in operator form (i.e. AND (&))

Step Two - Group the Data by Neighbourhood_Group and Neighbourhood. Don't forget you're looking to SUM up the Revenues.

The remaining steps we recommend you think very carefully about.

You might want to make use of the .reset_index(inplace=True) function to help reset the indexes in your Grouped Up Dataframe...!

Put your code in the box below.

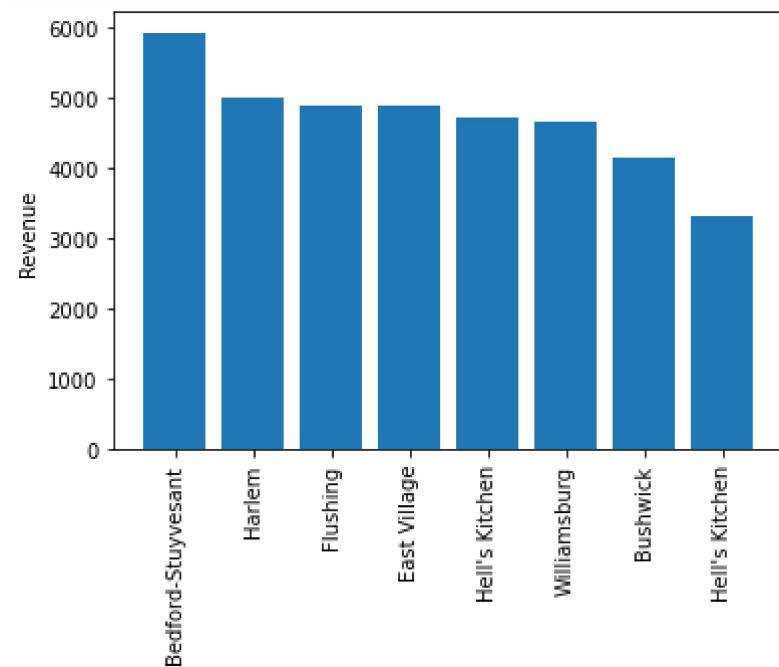
```
In [25]: condition1=airbnb[ 'neighbourhood' ]=='Williamsburg'  
condition2=airbnb[ 'neighbourhood' ]=='Bedford-Stuyvesant'  
condition3=airbnb[ 'neighbourhood' ]=='Harlem'  
condition4=airbnb[ 'neighbourhood' ]=="Hell's Kitchen"  
condition5=airbnb[ 'neighbourhood' ]=='East Village'  
condition6=airbnb[ 'neighbourhood' ]=='Bushwick'  
condition7=airbnb[ 'neighbourhood' ]=='Astoria'  
condition8=airbnb[ 'neighbourhood' ]=='Long Island City'  
condition9=airbnb[ 'neighbourhood' ]=='Flushing'  
step_1=airbnb[condition1|condition2|condition3|condition4|condition5|condition6|condition7|condition8|condition9]  
step_2=step_1.groupby(['neighbourhood', 'room_type'])['Revenue'].mean().sort_values(ascending=False).head(9).reset_index
```

```
In [26]: print(step_2)  
fig, ax = plt.subplots()  
ax.bar(step_2['neighbourhood'],step_2['Revenue'])  
ax.set_xticklabels(step_2['neighbourhood'], rotation=90)  
ax.set_ylabel('Revenue')  
plt.show()
```

```
neighbourhood      room_type      Revenue
0  Bedford-Stuyvesant  Entire home/apt  5923.995600
1          Harlem    Entire home/apt  4991.899614
2        Flushing    Entire home/apt  4879.349057
3   East Village    Entire home/apt  4871.841880
4   Hell's Kitchen  Private room  4713.443452
5  Williamsburg    Entire home/apt  4659.608673
6     Bushwick    Entire home/apt  4128.241888
7   Hell's Kitchen  Entire home/apt  4116.112890
8  Long Island City  Entire home/apt  3303.504167
```

```
C:\Users\anjus\AppData\Local\Temp\ipykernel_16972\2538763581.py:4: UserWarning: FixedFormatter should only be used together with FixedLocator
```

```
ax.set_xticklabels(step_2['neighbourhood'], rotation=90)
```



```
In [ ]:
```