Course Title:

Data Science (数据科学)

(Semester: Fall 2021)

Dr. Oluwarotimi W. SAMUEL

Research Center for Neural Engineering
Shenzhen Institutes of Advanced Technology
Chinese Academy of Sciences

Contact: (Email: samuel@siat.ac.cn & timitex92@gmail.com)

Phone: +86-15814491870

(2021.10.28)

Data Visualization

❑ **Outline for today’s lecture**

- ✓ Data Visualization
- ✓ Tools for Visualizing Data
- ✓ Case Study Using Online Dataset

Data Visualization

- **Objective:** This lecture will focus on Data Visualization with practical application using a real-life data.
- **Expectation:** At the end of this lecture, students are expected to understand the procedure for visualizing data in the context of a data science project.

❑ Data Visualization Definition:

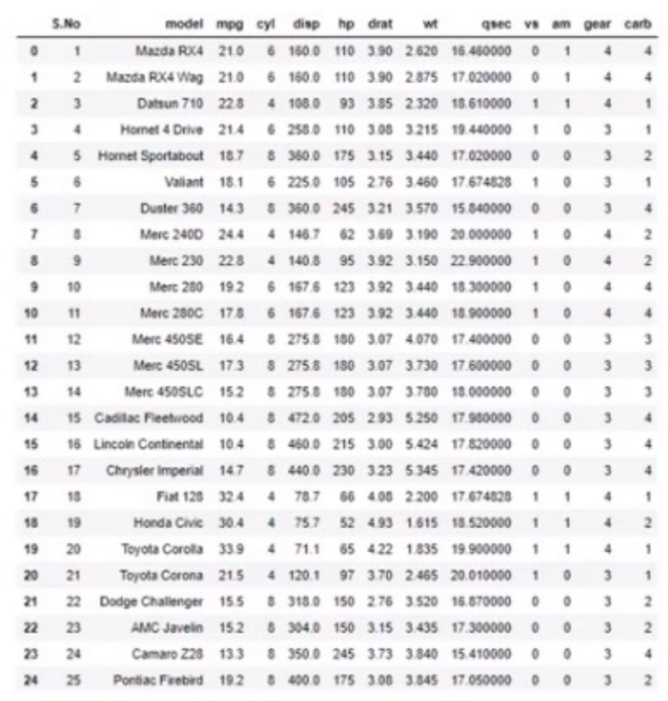
- ✓ Data visualization is an important step in DS process that involves the use of graphical/pictorial representation to present information/data.

- ✓ This allows us to better observe trends/patterns in our data set.

- ✓ Taking a look at a large set of numbers in a tabular form may limit understanding of inherent trends.

❑ Data Description via Visualization:

✓ Hence, visualization of data using graphs, charts, and other forms of graphical representation is essential.

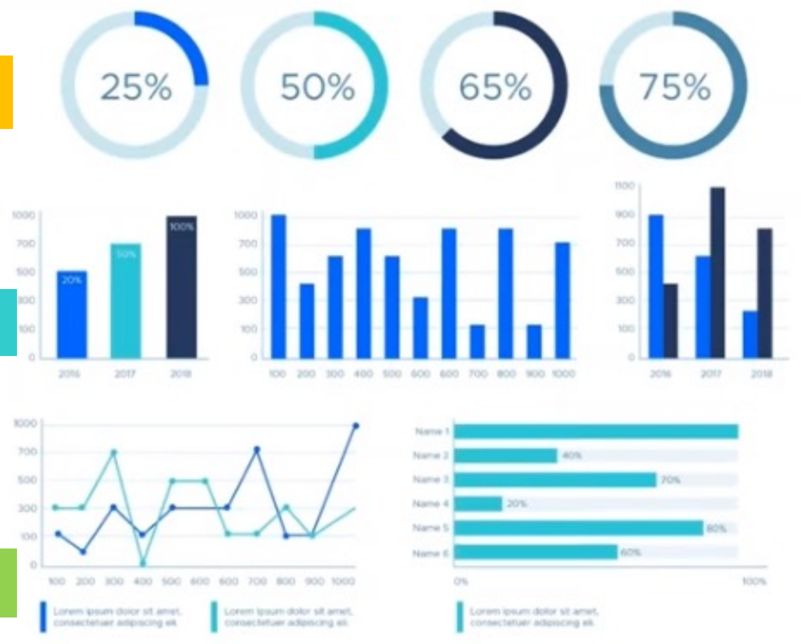


	S.No	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.460000	0	1	4	4
1	2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.020000	0	1	4	4
2	3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.610000	1	1	4	1
3	4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.440000	1	0	3	1
4	5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.020000	0	0	3	2
5	6	Valiant	18.1	6	225.0	105	2.76	3.460	17.674828	1	0	3	1
6	7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.840000	0	0	3	4
7	8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.000000	1	0	4	2
8	9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.900000	1	0	4	2
9	10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.300000	1	0	4	4
10	11	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.900000	1	0	4	4
11	12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.400000	0	0	3	3
12	13	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.600000	0	0	3	3
13	14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.000000	0	0	3	3
14	15	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.980000	0	0	3	4
15	16	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.820000	0	0	3	4
16	17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.420000	0	0	3	4
17	18	Fiat 128	32.4	4	78.7	66	4.08	2.200	17.674828	1	1	4	1
18	19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.520000	1	1	4	2
19	20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.900000	1	1	4	1
20	21	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.010000	1	0	3	1
21	22	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.870000	0	0	3	2
22	23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.300000	0	0	3	2
23	24	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.410000	0	0	3	4
24	25	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.050000	0	0	3	2

Pie chart

Bar plots

Line plots



❑ Importance of Data Visualization:

1. To view changes over time easily via a visual aid rather that plain data
2. To discover correlations among two or more variables seamlessly
3. To simplify complex information into user-friendly formats
4. To tell a better story with a bunch of pictures over time

❑ Visualization:

- ✓ Note that, Data Visualization is an essential tool for Data Science.
- ✓ Visualization brings Data to life.
- ✓ Often, a good visualization can convey trends and anomalies in the data more efficiently than a written description (text/table)

❑ Data Visualization:

Visualization can be a great way to convey your predictions and conclusions to others.

❑ Tools for Data Visualization:

We need to use some computational tools to create Data Visualization.

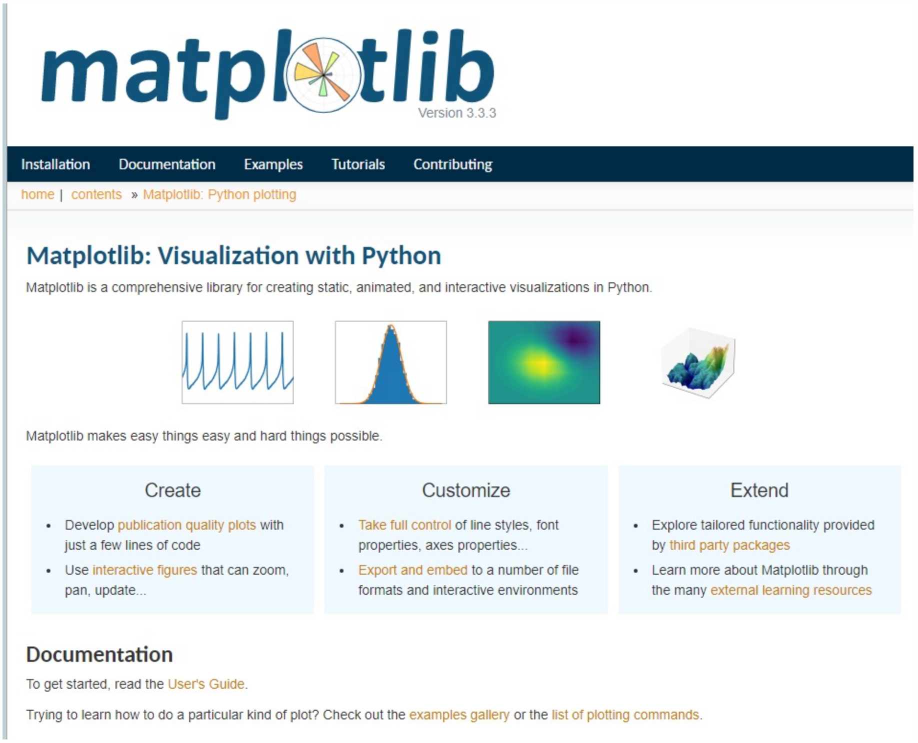
Two useful visualization tools that I will introduce to you are:

- ✓ Python's matplotlib
- ✓ Seaborn libraries

❑ Two Useful Data Visualization Tools:

Matplotlib is a library in Python that allows us to create a 2D plot of our data…

Seaborn is a library in Python that based on Matplotlib that allows us to create a Multi-Dimensional plots and more advanced visualization of our data



matplotlib Version 3.3.3

Installation Documentation Examples Tutorials Contributing

home | contents » Matplotlib: Python plotting

Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Matplotlib makes easy things easy and hard things possible.

Create
- Develop publication quality plots with just a few lines of code
- Use interactive figures that can zoom, pan, update...

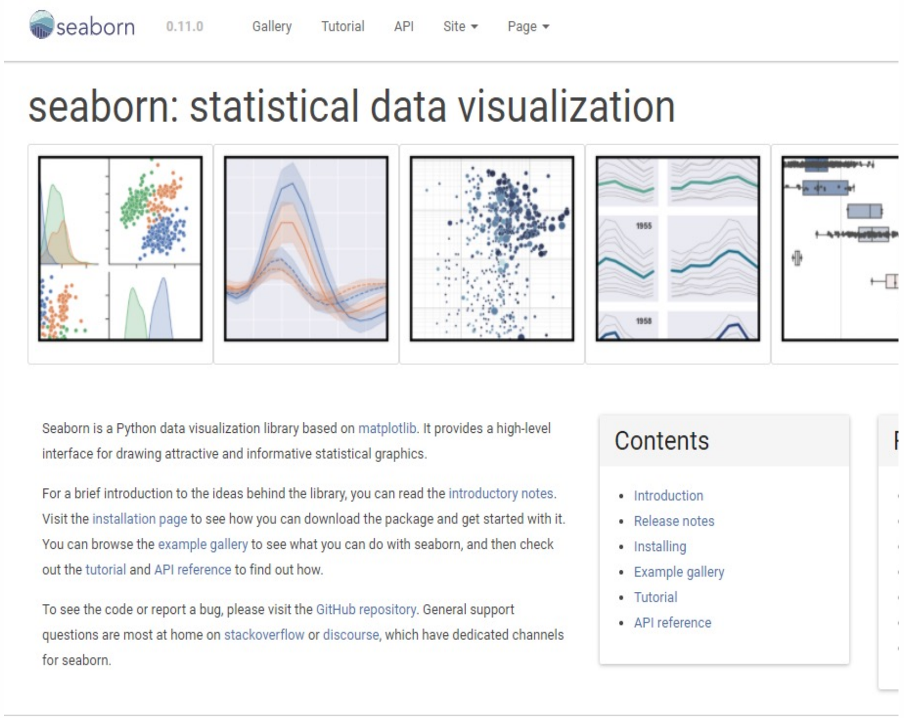
Customize
- Take full control of line styles, font properties, axes properties...
- Export and embed to a number of file formats and interactive environments

Extend
- Explore tailored functionality provided by third party packages
- Learn more about Matplotlib through the many external learning resources

Documentation

To get started, read the User's Guide.

Trying to learn how to do a particular kind of plot? Check out the examples gallery or the list of plotting commands.



seaborn 0.11.0 Gallery Tutorial API Site Page

seaborn: statistical data visualization

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

For a brief introduction to the ideas behind the library, you can read the introductory notes. Visit the installation page to see how you can download the package and get started with it. You can browse the example gallery to see what you can do with seaborn, and then check out the tutorial and API reference to find out how.

To see the code or report a bug, please visit the GitHub repository. General support questions are most at home on stackoverflow or discourse, which have dedicated channels for seaborn.

Contents
- Introduction
- Release notes
- Installing
- Example gallery
- Tutorial
- API reference

https://matplotlib.org/index.html

https://seaborn.pydata.org/

❑ Useful Data Visualization Tools:

✓ Usually, different types of charts are used to visualize different types of data (depending their types)

✓ Qualitative Data (also know as categorical data)

- Subtypes are **nominal** and **ordinal** data

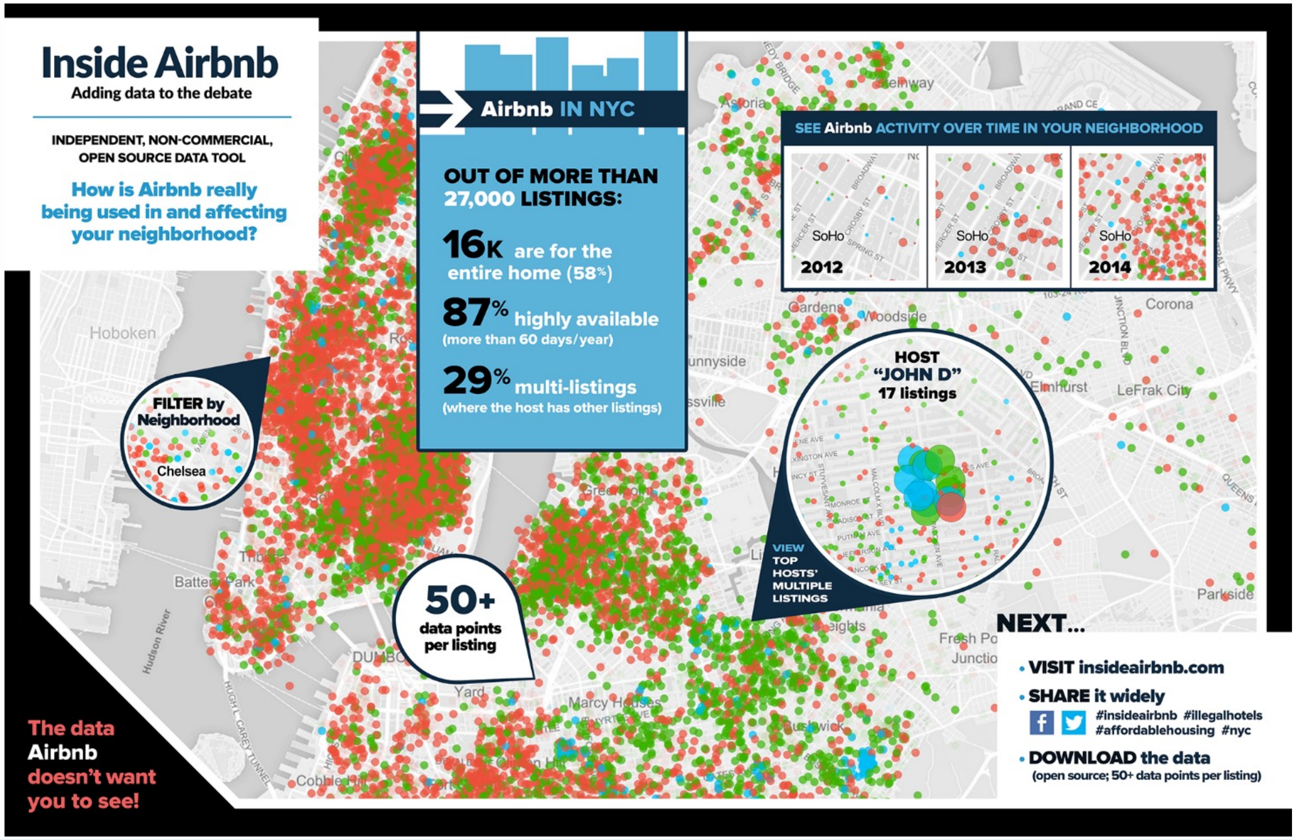
Nominal data has no inherent order

Ordinal data falls into ordered categories

❑ Case Study:

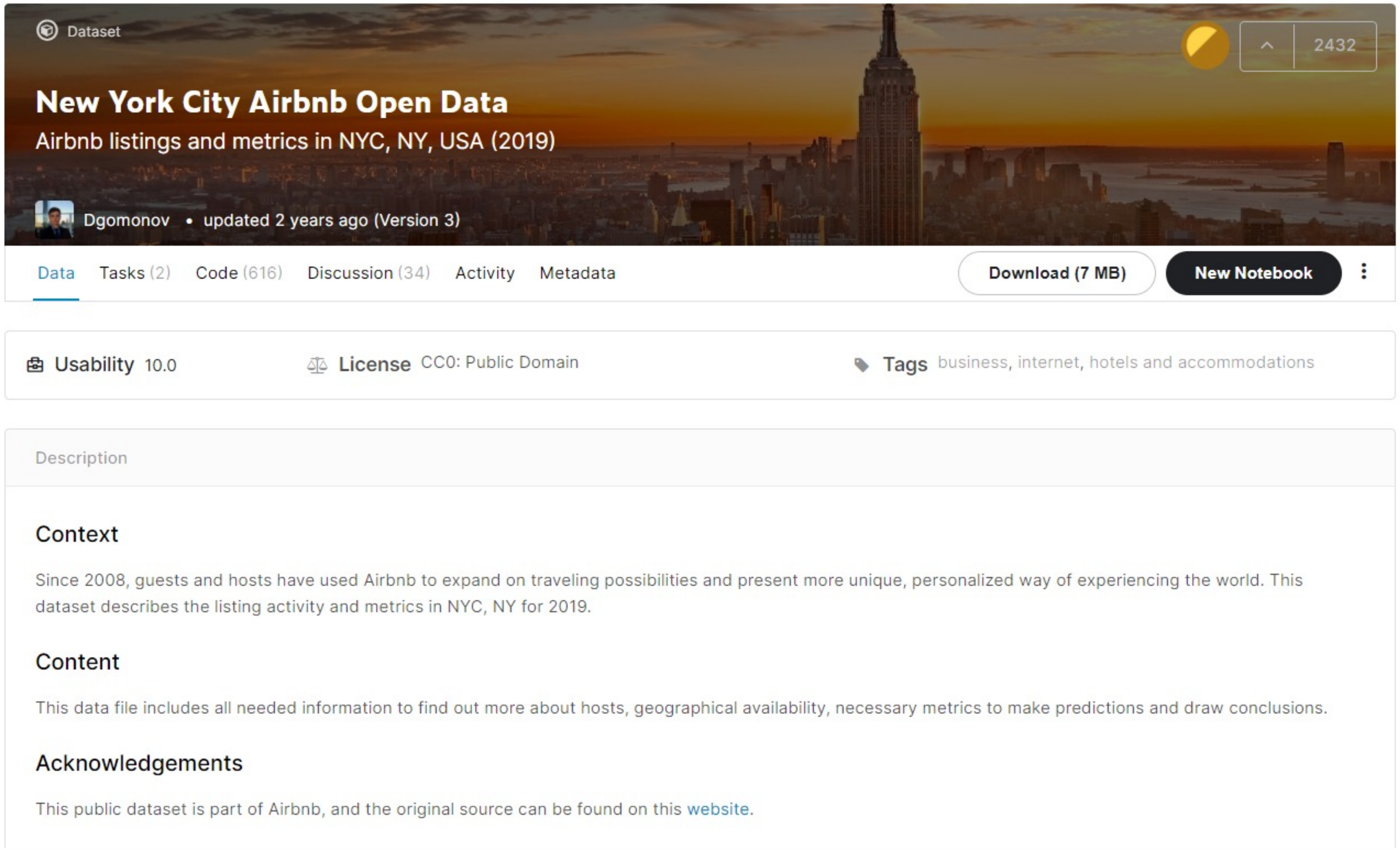
Airbnb--Provide data that quantifies the impact of short-term rentals on housing and residential communities; and also provides a platform to support advocacy for policies to protect our cities from the impacts of short-term rentals.

http://insideairbnb.com/about.html


Inside Airbnb
Adding data to the debate
INDEPENDENT, NON-COMMERCIAL, OPEN SOURCE DATA TOOL
How is Airbnb really being used in and affecting your neighborhood?
Airbnb IN NYC
OUT OF MORE THAN 27,000 LISTINGS:
16K are for the entire home (58%)
87% highly available (more than 60 days/year)
29% multi-listings (where the host has other listings)
SEE Airbnb ACTIVITY OVER TIME IN YOUR NEIGHBORHOOD
SoHo 2012
SoHo 2013
SoHo 2014
FILTER by Neighborhood
Chelsea
HOST "JOHN D" 17 listings
VIEW TOP HOSTS' MULTIPLE LISTINGS
50+ data points per listing
The data Airbnb doesn't want you to see!
NEXT...
VISIT insideairbnb.com
SHARE it widely
#insideairbnb #illegalhotels #affordablehousing #nyc
DOWNLOAD the data (open source; 50+ data points per listing)

Data Visualization

❑ Case Study:



New York City Airbnb Open Data

Airbnb listings and metrics in NYC, NY, USA (2019)

Dgomonov • updated 2 years ago (Version 3)

Usability 10.0 | License CC0: Public Domain | Tags business, internet, hotels and accommodations

Description

Context

Since 2008, guests and hosts have used Airbnb to expand on traveling possibilities and present more unique, personalized way of experiencing the world. This dataset describes the listing activity and metrics in NYC, NY for 2019.

Content

This data file includes all needed information to find out more about hosts, geographical availability, necessary metrics to make predictions and draw conclusions.

Acknowledgements

This public dataset is part of Airbnb, and the original source can be found on this website.

https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data

❑ Case Study:

Data Description

Since 2008, guests and hosts have used **Airbnb** to expand on traveling possibilities and present more unique, personalized way of experiencing the world. This dataset describes the listing activity and metrics in NYC, NY for 2019.

❑ Case Study:

Data Description

This data file includes all needed information to find out more about **hosts**, **geographical availability**, **prices**, necessary metrics to make **predictions** and **draw conclusions**.

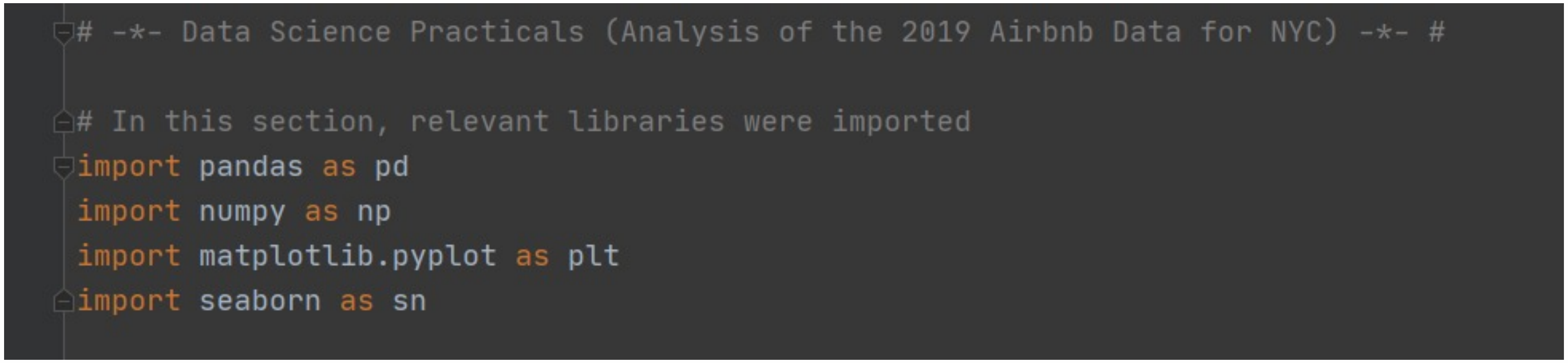
Case Study:
Inspiration

- ✓ What can we learn about different hosts and areas?
- ✓ What can we learn from predictions? (ex: locations, prices, reviews, etc.)
- ✓ Which hosts are the busiest and why?
- ✓ Is there any noticeable difference of traffic among different areas and what could be the reason for it?

Performing Data Visualization

❑ **Case Study:**

First, we need to import the following needed libraries (matplolib and seaborn) to work with the Airbnb data.



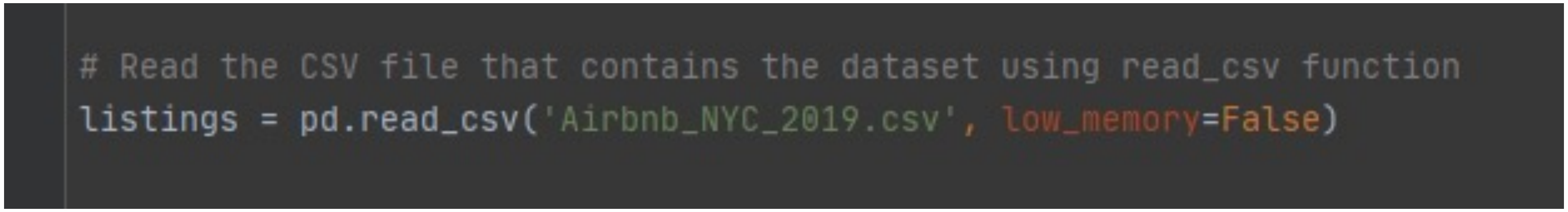
```
# -*- Data Science Practicals (Analysis of the 2019 Airbnb Data for NYC) -*- #

# In this section, relevant libraries were imported
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
```

Performing Data Visualization

❑ **Case Study:**

Next, we read the csv file that contains the data using the "read_csv()" function provided by the Pandas library.

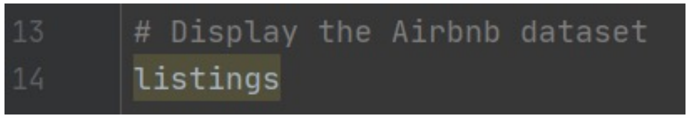


```
# Read the CSV file that contains the dataset using read_csv function
listings = pd.read_csv('Airbnb_NYC_2019.csv', low_memory=False)
```

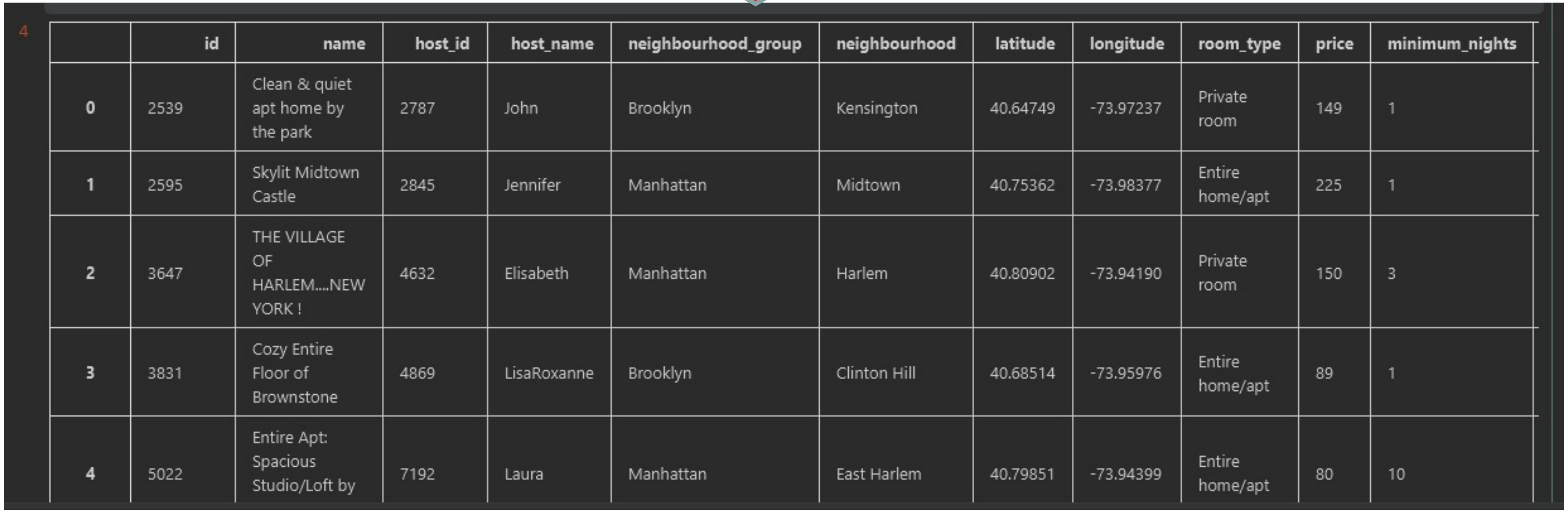
Performing Data Visualization

❑ Case Study:

Below is what the Airbnb data for NYC looks like using the following command for the display.



```
# Display the Airbnb dataset
listings
```



	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1
4	5022	Entire Apt: Spacious Studio/Loft by	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10

Performing Data Visualization

❑ **Case Study:**

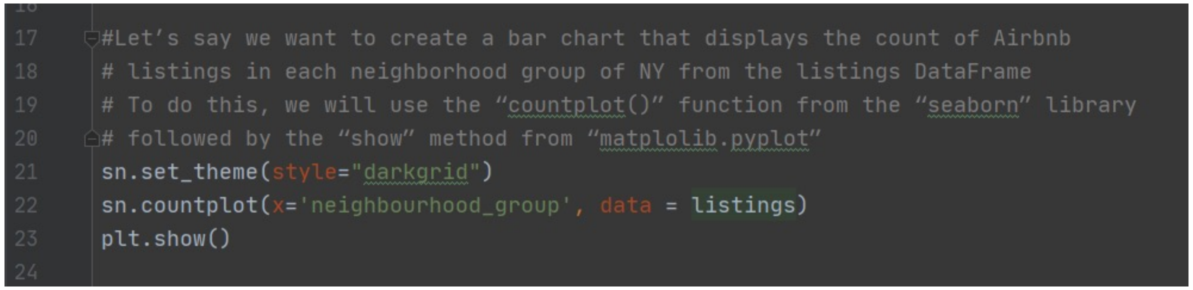
Let's say we want to create a **bar chart** that displays the count of Airbnb listings in each neighborhood group of NYC from the listings DataFrame

To do this, we will use the "countplot()" method from the "**seaborn**" library followed by the "**show**" method from "***matplolib.pyplot***"

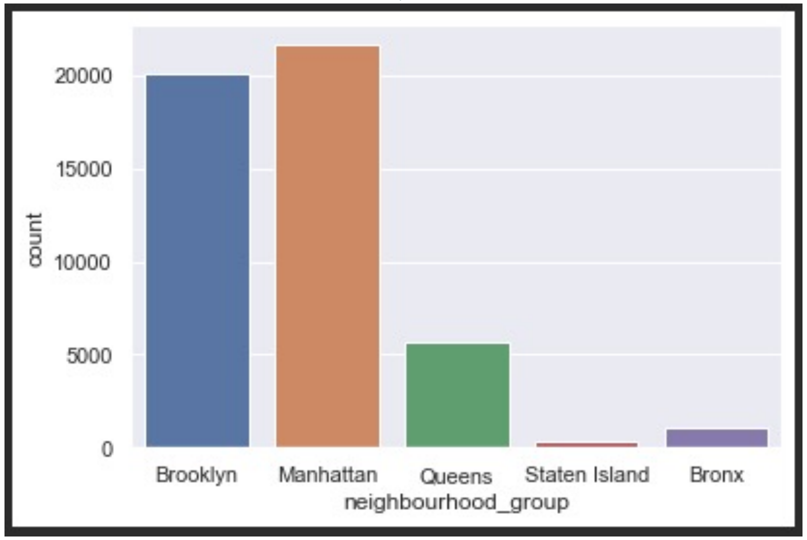
Data Visualization

❑ Case Study:

The result is shown as follows (count of neighborhood group):



```
#Let's say we want to create a bar chart that displays the count of Airbnb
# listings in each neighborhood group of NY from the listings DataFrame
# To do this, we will use the "countplot()" function from the "seaborn" library
# followed by the "show" method from "matplolib.pyplot"
sn.set_theme(style="darkgrid")
sn.countplot(x='neighbourhood_group', data = listings)
plt.show()
```



Performing Data Visualization

❑ **Case Study:**

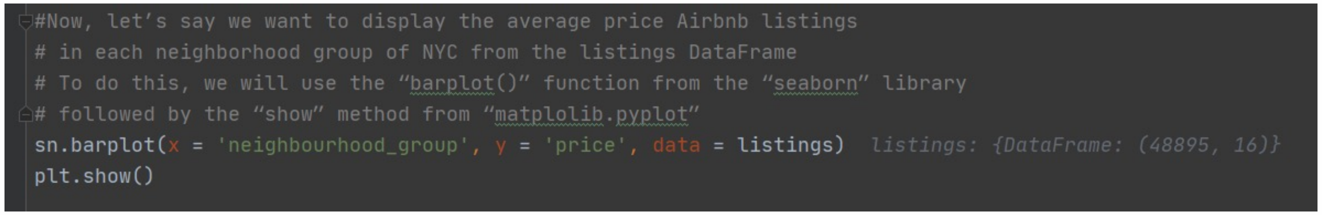
Now, let's say we want to display the **average number of** in each neighborhood group of NYC from the listings DataFrame

To do this, we will use the "**barplot()**" function from the "**seaborn**" library followed by the "**show**" method from "matplolib.pyplot"

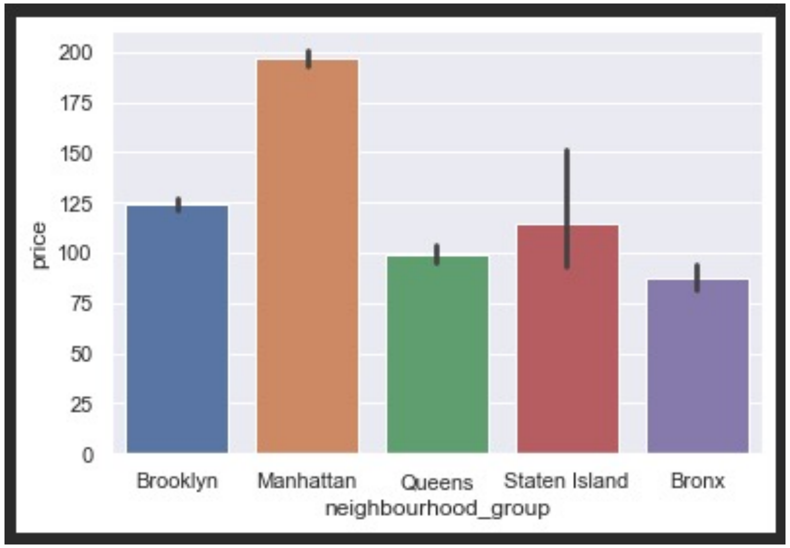
Performing Data Visualization

❑ Case Study:

Below is the results:



```
#Now, let’s say we want to display the average price Airbnb listings
# in each neighborhood group of NYC from the listings DataFrame
# To do this, we will use the “barplot()” function from the “seaborn” library
# followed by the “show” method from “matplolib.pyplot”
sn.barplot(x = 'neighbourhood_group', y = 'price', data = listings)  listings: {DataFrame: (48895, 16)}
plt.show()
```



Performing Data Visualization

❑ **Case Study:**

The **black lines (confidence intervals)** at the middle of each bar denotes the confidence interval that were generated by the bar plot method.

Though **confidence intervals** are an important concept, but it is not within the scope of this lecture, so we will proceed to remove it from the plot as follows.

We can achieve this by setting the "ci" parameter to False.

Performing Data Visualization

- **Case Study:**

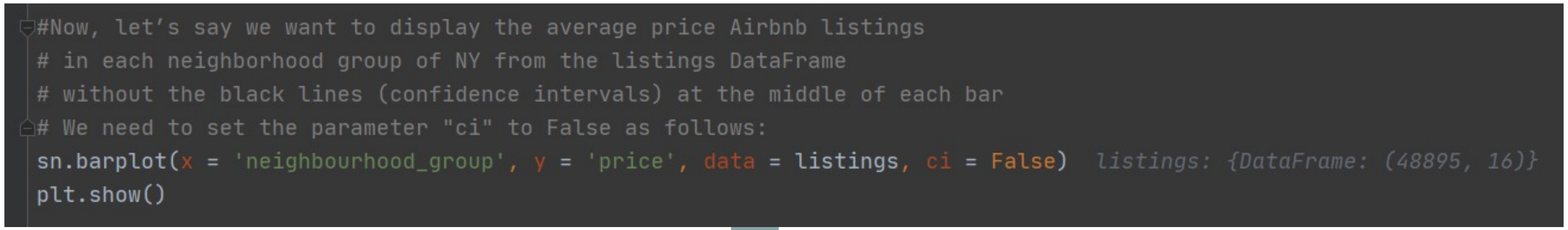
Let's say we want to reproduce the barplot above without the black lines (**confidence intervals: ci**).

Though confidence intervals are an important concept, but it is not within the scope of this lecture, so we will proceed to remove it from the plot as follows.

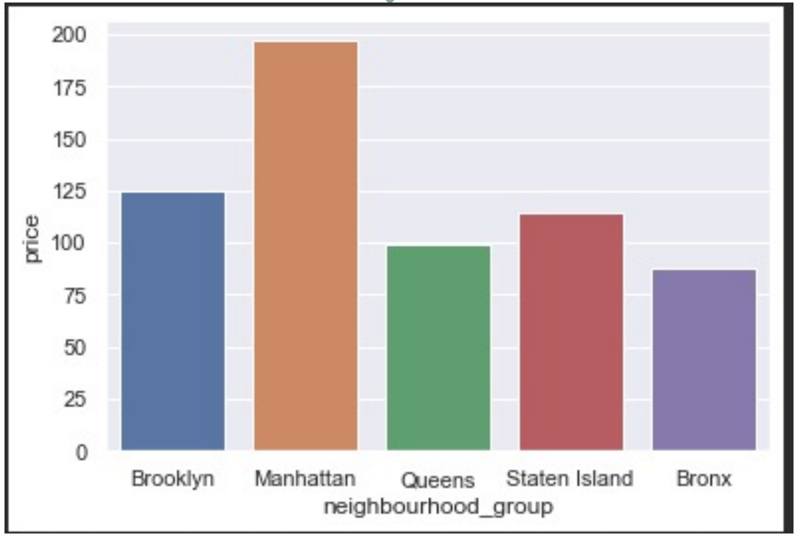
Performing Data Visualization

❑ Case Study:

Below is the results:



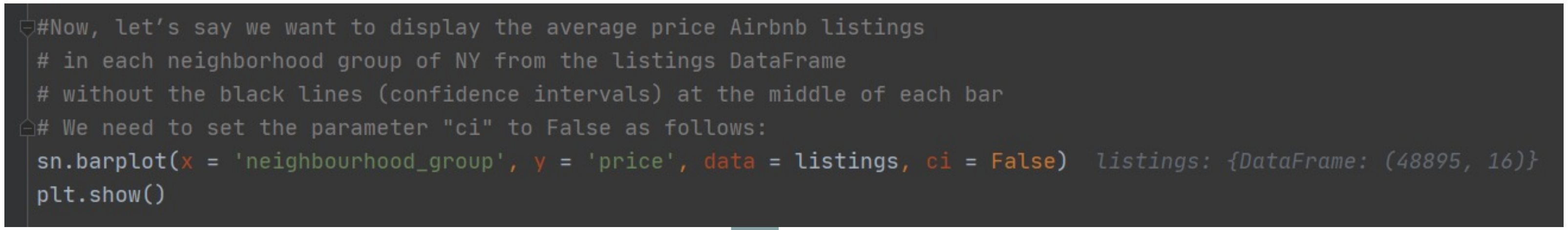
```
#Now, let’s say we want to display the average price Airbnb listings
# in each neighborhood group of NY from the listings DataFrame
# without the black lines (confidence intervals) at the middle of each bar
# We need to set the parameter "ci" to False as follows:
sn.barplot(x = 'neighbourhood_group', y = 'price', data = listings, ci = False)  listings: {DataFrame: (48895, 16)}
plt.show()
```



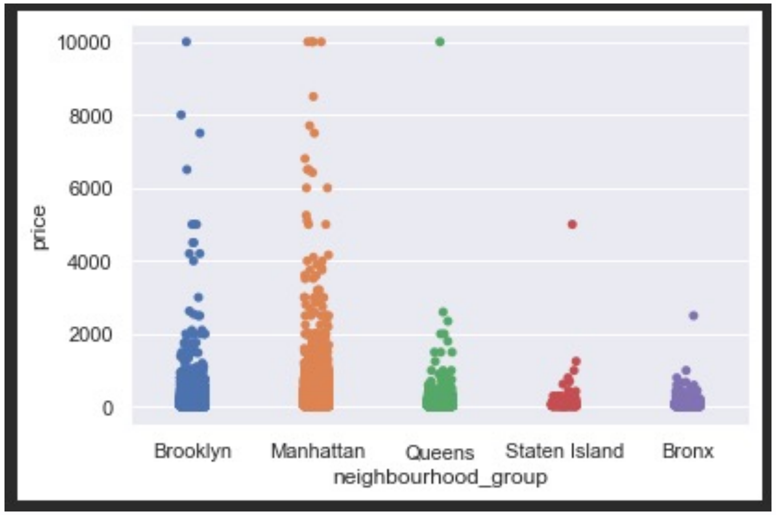
Performing Data Visualization

❑ Case Study:

Below is the results:



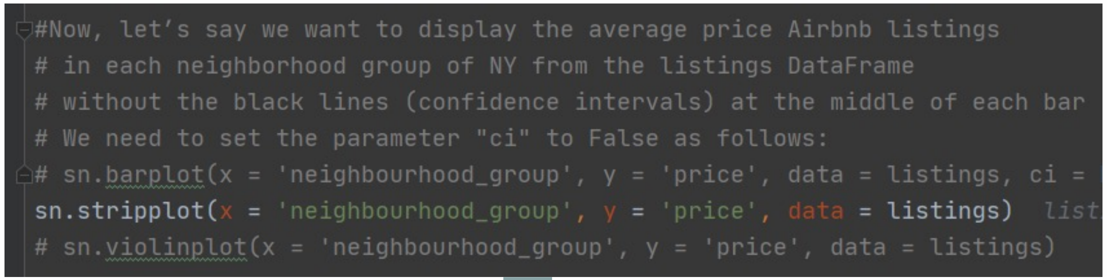
```
#Now, let’s say we want to display the average price Airbnb listings
# in each neighborhood group of NY from the listings DataFrame
# without the black lines (confidence intervals) at the middle of each bar
# We need to set the parameter "ci" to False as follows:
sn.barplot(x = 'neighbourhood_group', y = 'price', data = listings, ci = False)  listings: {DataFrame: (48895, 16)}
plt.show()
```



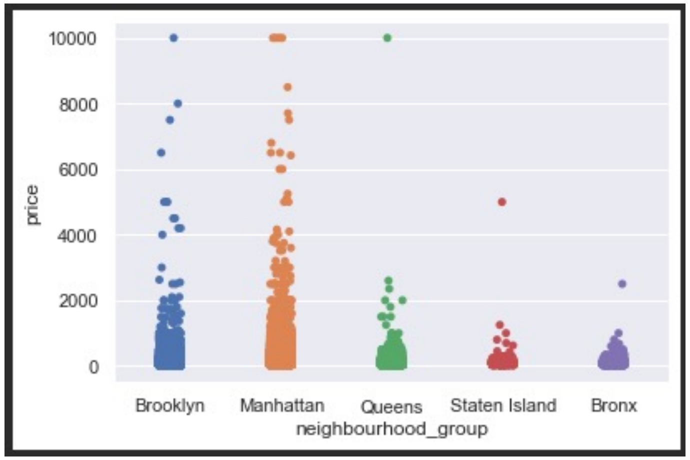
Performing Data Visualization

❑ Case Study:

Below is the results using “stripplot()” function



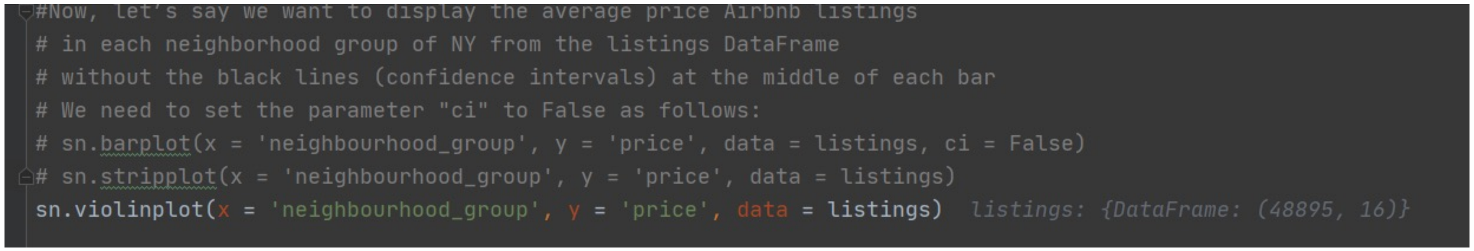
```
#Now, let’s say we want to display the average price Airbnb listings
# in each neighborhood group of NY from the listings DataFrame
# without the black lines (confidence intervals) at the middle of each bar
# We need to set the parameter "ci" to False as follows:
# sn.barplot(x = 'neighbourhood_group', y = 'price', data = listings, ci =
sn.stripplot(x = 'neighbourhood_group', y = 'price', data = listings)  list
# sn.violinplot(x = 'neighbourhood_group', y = 'price', data = listings)
```



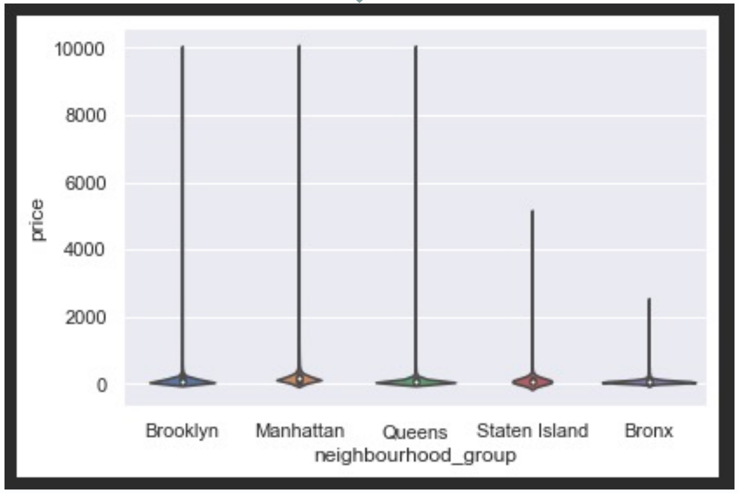
Performing Data Visualization

❑ **Case Study:**

Below is the results using "violinplot()" function



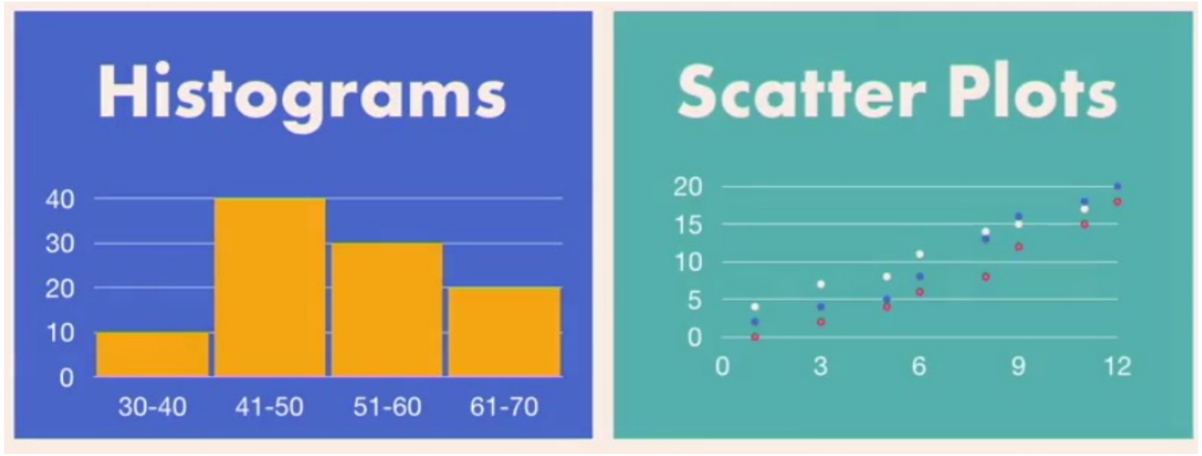
```
#Now, let's say we want to display the average price Airbnb listings
# in each neighborhood group of NY from the listings DataFrame
# without the black lines (confidence intervals) at the middle of each bar
# We need to set the parameter "ci" to False as follows:
# sn.barplot(x = 'neighbourhood_group', y = 'price', data = listings, ci = False)
# sn.stripplot(x = 'neighbourhood_group', y = 'price', data = listings)
sn.violinplot(x = 'neighbourhood_group', y = 'price', data = listings)  listings: {DataFrame: (48895, 16)}
```



Performing Data Visualization

❑ **Case Study:**

Next, we shall look at some common charts for visualizing Quantitative Data such as the **Histogram** and **Scatter Plots.**



NB: Quantitative data is another term for numerical data

Performing Data Visualization

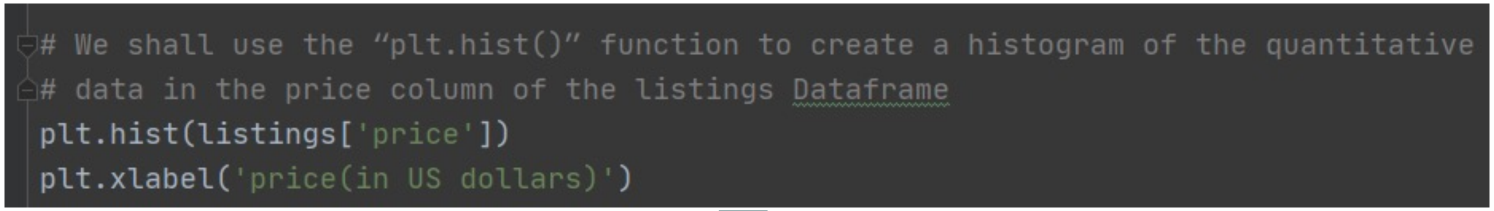
❑ **Case Study:**

Next, we will use the “plt.hist()” function to create a histogram of the quantitative data in the price column of the listings DataFrame

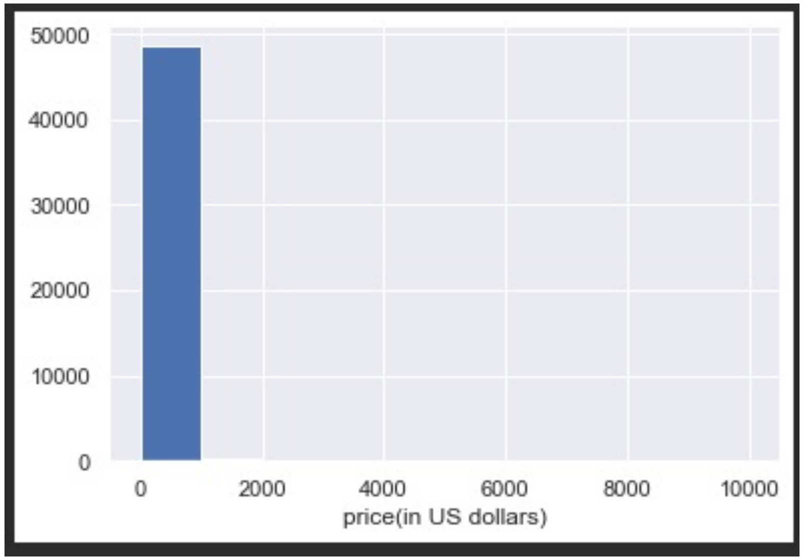
Performing Data Visualization

❑ **Case Study:**

Below is the results:



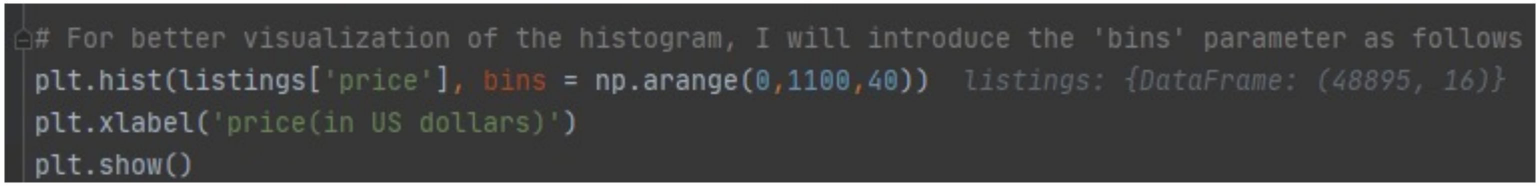
```
# We shall use the “plt.hist()” function to create a histogram of the quantitative
# data in the price column of the listings Dataframe
plt.hist(listings['price'])
plt.xlabel('price(in US dollars)')
```



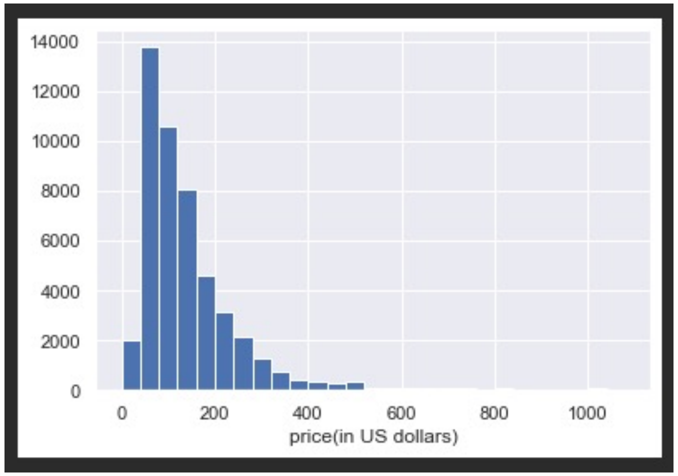
Performing Data Visualization

❑ Case Study:

To better see where the data lies in the histogram, I will introduce the “bins” parameters:



```
# For better visualization of the histogram, I will introduce the 'bins' parameter as follows
plt.hist(listings['price'], bins = np.arange(0,1100,40))  listings: {DataFrame: (48895, 16)}
plt.xlabel('price(in US dollars)')
plt.show()
```



Now it is easier to see where the data lies with respect to the bins.

A lot of the Airbnb price listings lies between 50 and 500 USD

Performing Data Visualization

❑ **Case Study:**

Next, let's talk about the **Scatter Plot**. They are often used to compare two set of quantitative data.

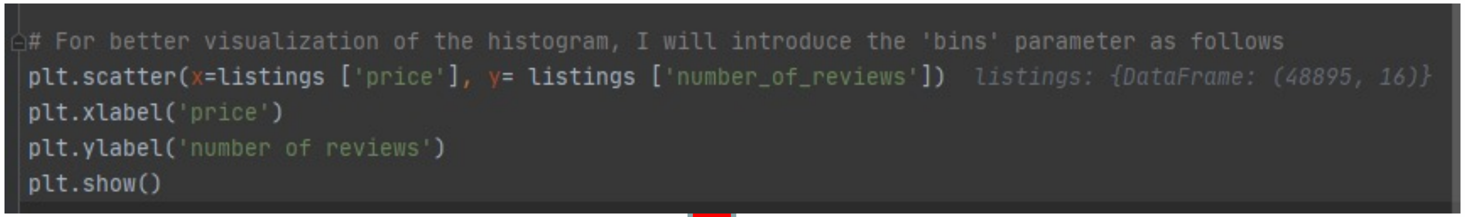
Assuming we want to compare the **prices** of Airbnb listings and the numbers of **reviews** for the listings.

We can achieve this as follows:

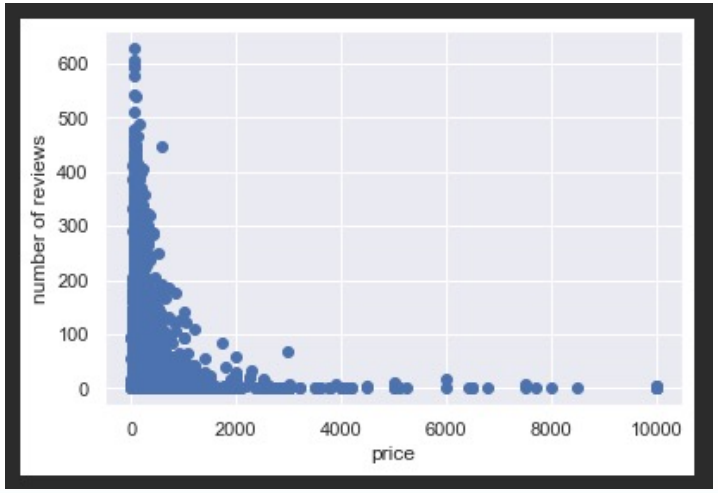
Performing Data Visualization

❑ Case Study:

To do this, we will use the ‘**scatter()**’ function from the matplotlib.pyplot.



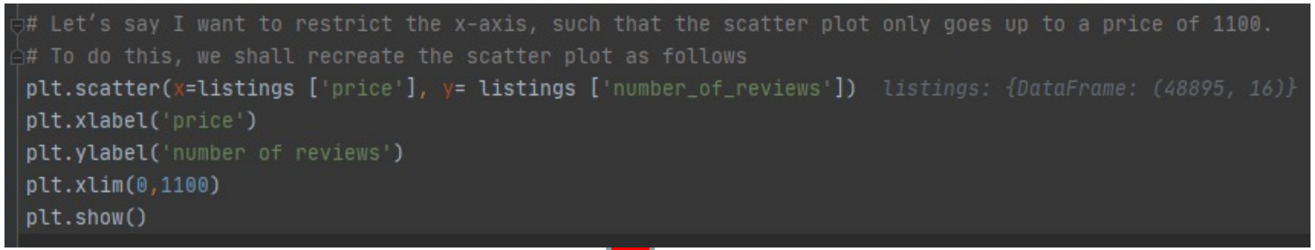
```
# For better visualization of the histogram, I will introduce the 'bins' parameter as follows
plt.scatter(x=listings ['price'], y= listings ['number_of_reviews'])  listings: {DataFrame: (48895, 16)}
plt.xlabel('price')
plt.ylabel('number of reviews')
plt.show()
```



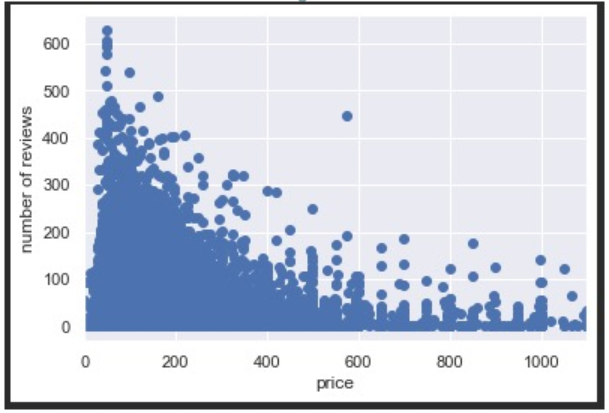
Performing Data Visualization

❑ Case Study:

Let's say we want to restrict the **x-axis**, such that the scatter plot only goes up to a price of **1100**. To do this, we shall recreate the plot as follows:



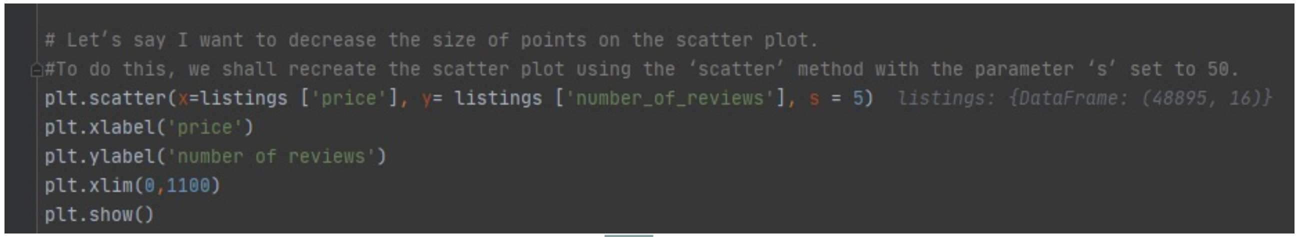
```
# Let's say I want to restrict the x-axis, such that the scatter plot only goes up to a price of 1100.
# To do this, we shall recreate the scatter plot as follows
plt.scatter(x=listings ['price'], y= listings ['number_of_reviews'])  listings: {DataFrame: (48895, 16)}
plt.xlabel('price')
plt.ylabel('number of reviews')
plt.xlim(0,1100)
plt.show()
```



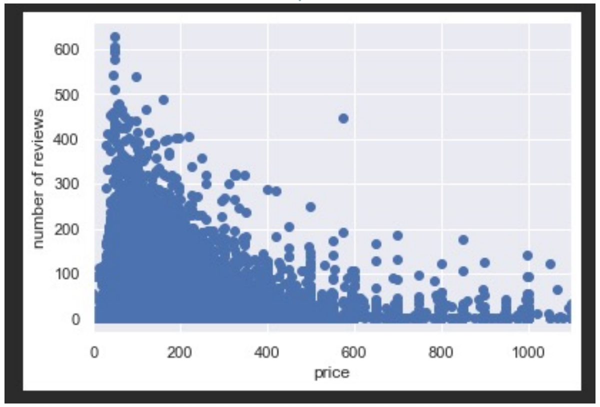
Performing Data Visualization

❑ **Case Study:**

Let's say I want to decrease the size of points on the scatter plot To do this, we shall recreate the scatter plot using the 'scatter' method with the parameter 's' set to 5.

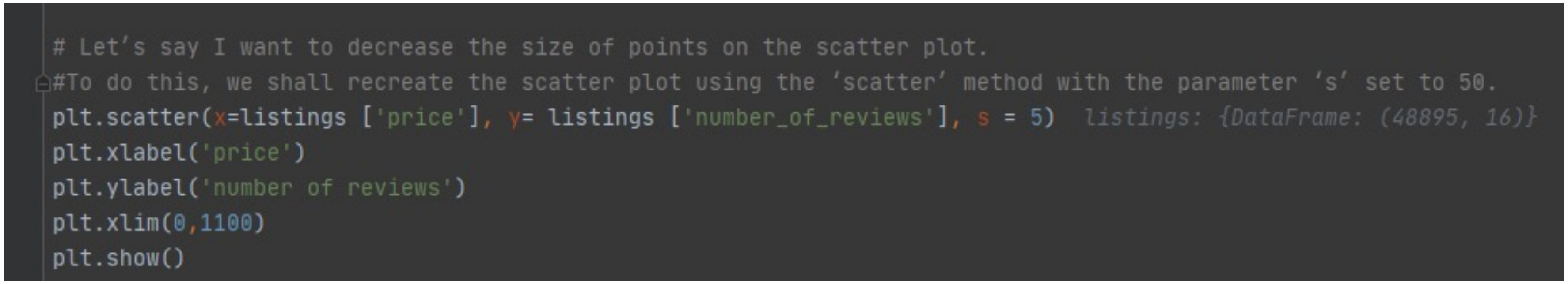


```
# Let's say I want to decrease the size of points on the scatter plot.
#To do this, we shall recreate the scatter plot using the 'scatter' method with the parameter 's' set to 50.
plt.scatter(x=listings ['price'], y= listings ['number_of_reviews'], s = 5)  listings: {DataFrame: (48895, 16)}
plt.xlabel('price')
plt.ylabel('number of reviews')
plt.xlim(0,1100)
plt.show()
```

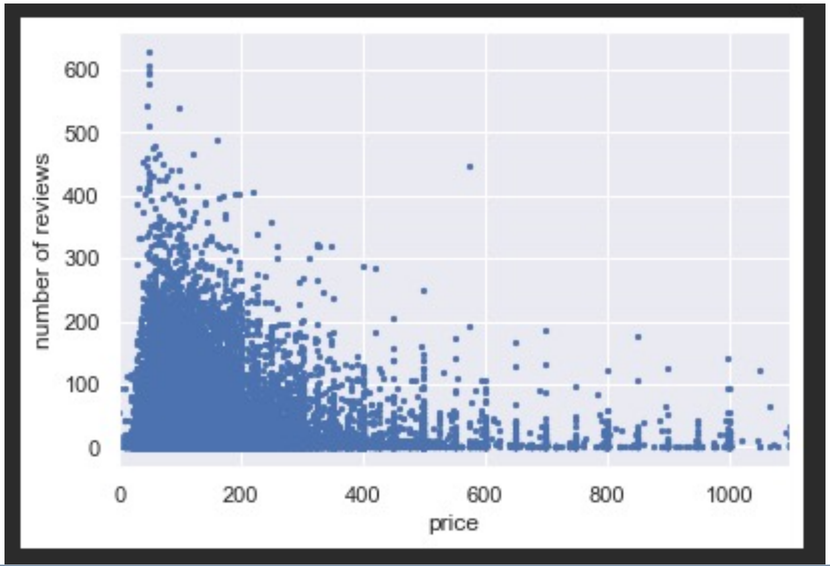


Performing Data Visualization

❑ Case Study:



```
# Let's say I want to decrease the size of points on the scatter plot.
#To do this, we shall recreate the scatter plot using the 'scatter' method with the parameter 's' set to 50.
plt.scatter(x=listings ['price'], y= listings ['number_of_reviews'], s = 5)  listings: {DataFrame: (48895, 16)}
plt.xlabel('price')
plt.ylabel('number of reviews')
plt.xlim(0,1100)
plt.show()
```



The points on this scatter plot are a bit smaller, it is a little bit easier to look at.

❑ Performing Data Visualization:

- ✓ Looking at the plot at first instance, one could observe some obvious trend…
- ✓ That is, listings with Lower Prices have more reviews.

- ✓ This leads me to ask the following questions?
- ✓ Is there an association between Price and the Number of Reviews for Listings?

- ✓ This is just one type of the questions that could come up when you are Visualizing your Data

- ✓ Thus, this may prompted to conduct further analysis to investigate this question.

❑ Performing Data Visualization:

✓ In summary, Visualizing your data helps you observe trends which leads you to ask questions, that then inform the kind of further analysis you will perform on the data.

✓ Now you know how to know use Histogram and Scatter plots to visualize your data.

Questions and Comments!

Thank You

中国科学院深圳先进技术研究院
SHENZHEN INSTITUTES OF ADVANCED TECHNOLOGY
CHINESE ACADEMY OF SCIENCES