Matplotlib.practice

Javier René Quispe Tenorio

25/12/2021

Using the matplotlib.pyplot interface

There are many ways to use Matplotlib. In this course, we will focus on the pyplot interface, which provides the most flexibility in creating and customizing data visualizations.

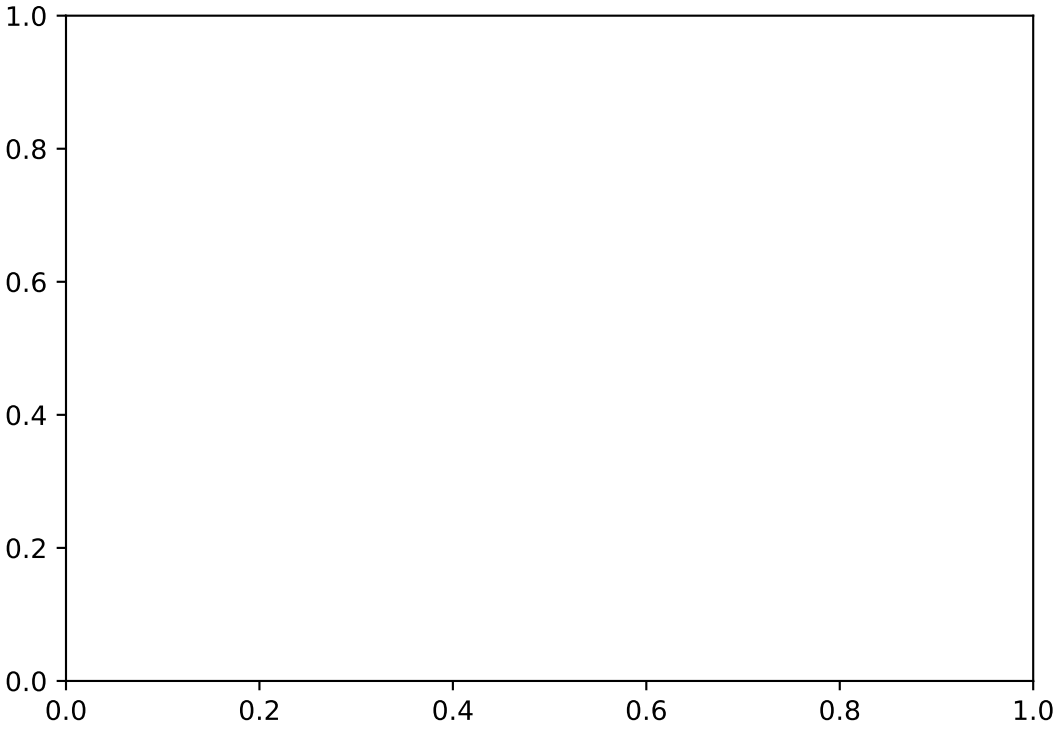
Initially, we will use the pyplot interface to create two kinds of objects: Figure objects and Axes objects.

```
# Import the matplotlib.pyplot submodule and name it plt
import matplotlib.pyplot as plt

# Create a Figure and an Axes with plt.subplots
fig, ax = plt.subplots()

# Call the show function to show the result
plt.show()
```

```
## Traceback (most recent call last):
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\backends\bac
##     self.draw()
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\backends\bac
##     self.figure.draw(self.renderer)
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\artist.py", l
##     return draw(artist, renderer, *args, **kwargs)
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\figure.py", l
##     renderer, self, artists, self.suppressComposite)
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\image.py", li
##     a.draw(renderer)
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\artist.py", l
##     return draw(artist, renderer, *args, **kwargs)
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\cbook\deprec
##     return func(*inner_args, **inner_kwargs)
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\axes\_base.py
##     self._update_title_position(renderer)
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\axes\_base.py
##     if (ax.xaxis.get_ticks_position() in ['top', 'unknown']
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\axis.py", lin
##     self._get_ticks_position()]
##   File "C:\Users\HOME\AppData\Local\R-MINI~1\envs\R-RETI~1\lib\site-packages\matplotlib\axis.py", lin
##     major = self.majorTicks[0]
## IndexError: list index out of range
```



Adding data to an Axes object

Adding data to a figure is done by calling methods of the Axes object. In this exercise, we will use the plot method to add data about rainfall in two American cities: Seattle, WA and Austin, TX.

The data are stored in two Pandas DataFrame objects that are already loaded into memory: seattle_weather stores information about the weather in Seattle, and austin_weather stores information about the weather in Austin. Each of the data frames has a "MONTHS" column that stores the three-letter name of the months. Each also has a column named "MLY-PRCP-NORMAL" that stores the average rainfall in each month during a ten-year period.

In this exercise, you will create a visualization that will allow you to compare the rainfall in these two cities.

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
StormEvents_2010= pd.read_csv(r'C:\Users\HOME\Desktop\Matlab\Exploratory Data Analysis\StormEvents\Storm
StormEvents_2011= pd.read_csv(r'C:\Users\HOME\Desktop\Matlab\Exploratory Data Analysis\StormEvents\Storm
StormEvents_2012= pd.read_csv(r'C:\Users\HOME\Desktop\Matlab\Exploratory Data Analysis\StormEvents\Storm
StormEvents_2013= pd.read_csv(r'C:\Users\HOME\Desktop\Matlab\Exploratory Data Analysis\StormEvents\Storm
StormEvents_2014= pd.read_csv(r'C:\Users\HOME\Desktop\Matlab\Exploratory Data Analysis\StormEvents\Storm
StormEvents_2015= pd.read_csv(r'C:\Users\HOME\Desktop\Matlab\Exploratory Data Analysis\StormEvents\Storm
StormEvents_2016= pd.read_csv(r'C:\Users\HOME\Desktop\Matlab\Exploratory Data Analysis\StormEvents\Storm
StormEvents_2017= pd.read_csv(r'C:\Users\HOME\Desktop\Matlab\Exploratory Data Analysis\StormEvents\Storm
StormEvents_2018= pd.read_csv(r'C:\Users\HOME\Desktop\Matlab\Exploratory Data Analysis\StormEvents\Storm
```

```
# Import the matplotlib.pyplot submodule and name it plt
import matplotlib.pyplot as plt
```

```
# Create a Figure and an Axes with plt.subplots
fig, ax = plt.subplots()

# Plot MLY-PRCP-NORMAL from seattle_weather against MONTH
ax.plot(StormEvents_2010["Month"], StormEvents_2010["Event_Type"])

# Plot MLY-PRCP-NORMAL from austin_weather against MONTH
ax.plot(StormEvents_2011["Month"], StormEvents_2011["Event_Type"])

# Call the show function
plt.show()
```