Python Data Science

Contents

Time required: 90 minutes

1. Save Python code in a Google Colab Notebook.

Tutorial 1: Plot Daily Weather in Scottsbluff

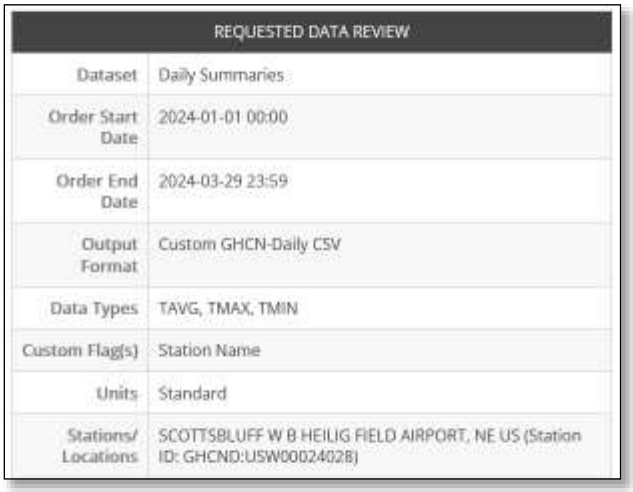
We are going to work with some daily weather data from NOAA (National Oceanic and Atmospheric Administration).

1. Go to: https://www.ncdc.noaa.gov/cdo-web/search
2. Select **Weather Observation Type/Dataset**
3. **Select Dataset**: Daily Summaries.
4. **Select Date Range**: A year ago to the current date. (You can select a much longer period if you wish)
5. **Search For:** ZIP Codes
6. **Enter a Search Term:** 69361 (or a different Zip Code if you wish.)
7. Click **Search.**
8. You will go a screen with a list of weather stations. **Scottsbluff, NE 69361 →** Click **Add To Cart.**
9. Click the **Cart (Free Data).**
10. **Select Cart Options → Custom GHCN-Daily CSV** (You can check and change the date range here if you wish.

11. Click **Continue.**

12. Custom Options: Daily Summaries → Select data types for customer output: **Air Temperature**. Click Continue.

13. You will get a page that says **Review Order**.



REQUESTED DATA REVIEW	
Dataset	Daily Summaries
Order Start Date	2024-01-01 00:00
Order End Date	2024-03-29 23:59
Output Format	Custom GHCN-Daily CSV
Data Types	TAVG, TMAX, TMIN
Custom Flag(s)	Station Name
Units	Standard
Stations/ Locations	SCOTTSBLUFF W B HEILIG FIELD AIRPORT, NE US (Station ID: GHCND:USW00024028)

14. Enter your email address to receive the link to your data.

15. **Submit Order**.

16. You should get a confirmation email that your order is being processed.

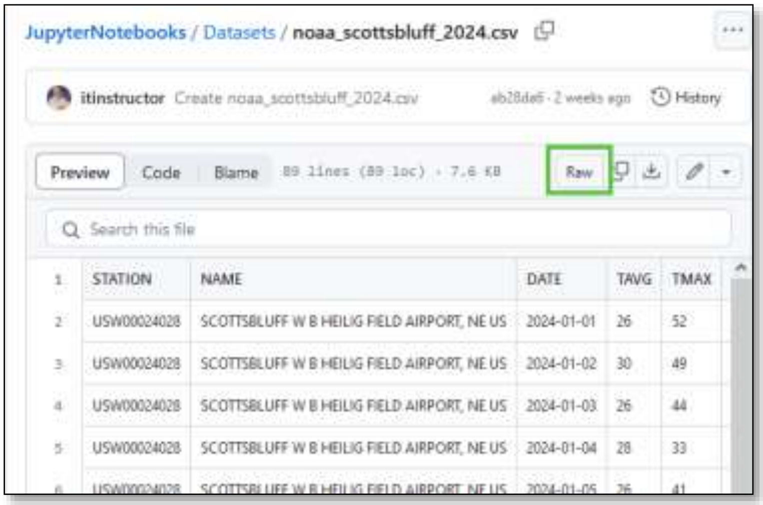
17. In a couple of minutes: You should get an email with a Download link.

18. Download the file as: **noaa_scottsbluff_2025.csv** in one of your GitHub repositories.

19. Commit the repository.

20. Got to the repository at www.github.com

21. Click on the csv file → Click the **raw** button.


JupyterNotebooks / Datasets / noaa_scottsbluff_2024.csv

itinstructor Create noaa_scottsbluff_2024.csv ab28da6 · 2 weeks ago History

Preview Code Blame 89 lines (89 loc) · 7.6 KB Raw

Search this file

1	STATION	NAME	DATE	TAVG	TMAX
2	USW00024028	SCOTTSBLUFF W B HEILIG FIELD AIRPORT, NE US	2024-01-01	26	52
3	USW00024028	SCOTTSBLUFF W B HEILIG FIELD AIRPORT, NE US	2024-01-02	30	49
4	USW00024028	SCOTTSBLUFF W B HEILIG FIELD AIRPORT, NE US	2024-01-03	26	44
5	USW00024028	SCOTTSBLUFF W B HEILIG FIELD AIRPORT, NE US	2024-01-04	28	33
6	USW00024028	SCOTTSBLUFF W B HEILIG FIELD AIRPORT, NE US	2024-01-05	26	41

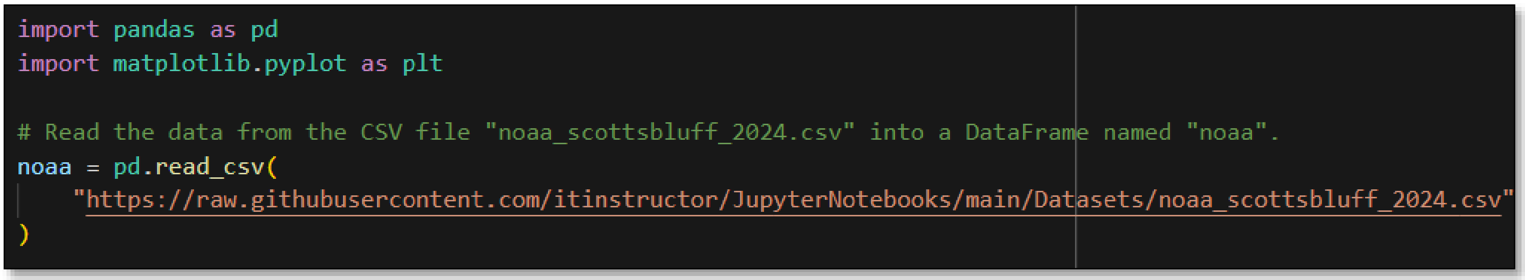
22. Select and copy the url. It should look something like this.

NOTE: This has the extra content from other weather stations other than Heilig Field removed.

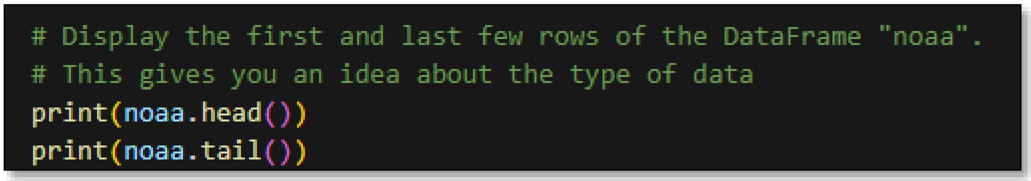
https://github.com/itinstructor/JupyterNotebooks/blob/main/Datasets/Weather/noaa_scottsbluff_2025.csv

Python Tutorial 1: Plot Daily Weather in Scottsbluff

1. In Google Colab → Create a Notebook named: **PythonNOAAScottsbluffDaily**
2. Enter the following code. Paste in your GitHub url for your csv file.

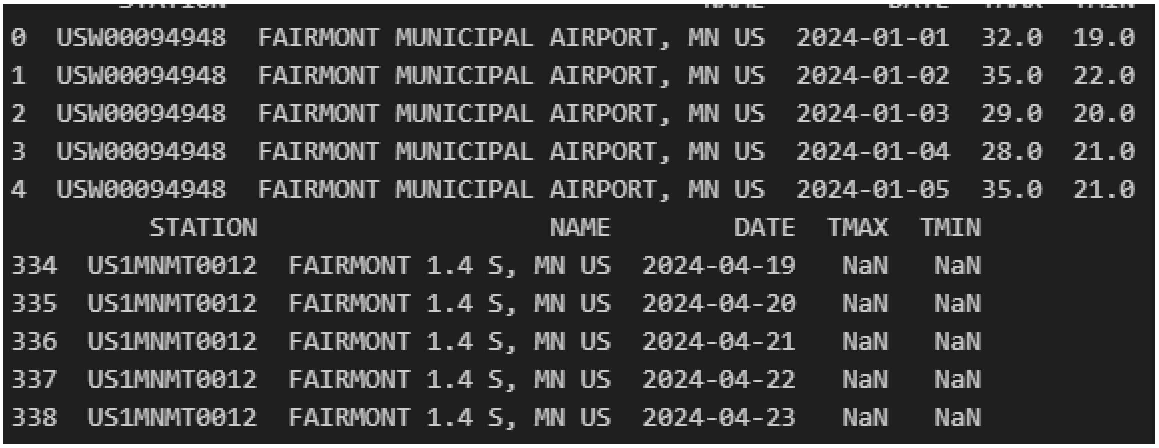


```
import pandas as pd
import matplotlib.pyplot as plt

# Read the data from the CSV file "noaa_scottsbluff_2024.csv" into a DataFrame named "noaa".
noaa = pd.read_csv(
    "https://raw.githubusercontent.com/itinstructor/JupyterNotebooks/main/Datasets/noaa_scottsbluff_2024.csv"
)
```

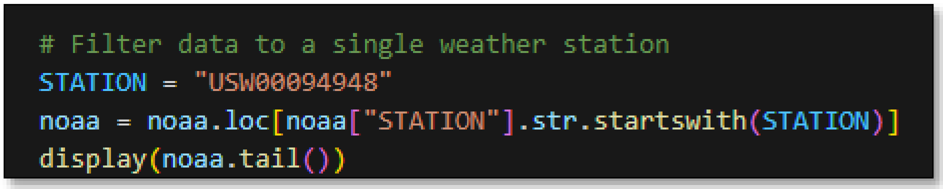


```
# Display the first and last few rows of the DataFrame "noaa".
# This gives you an idea about the type of data
print(noaa.head())
print(noaa.tail())
```

If you end up with more than one weather station, you will want to filter down to one.

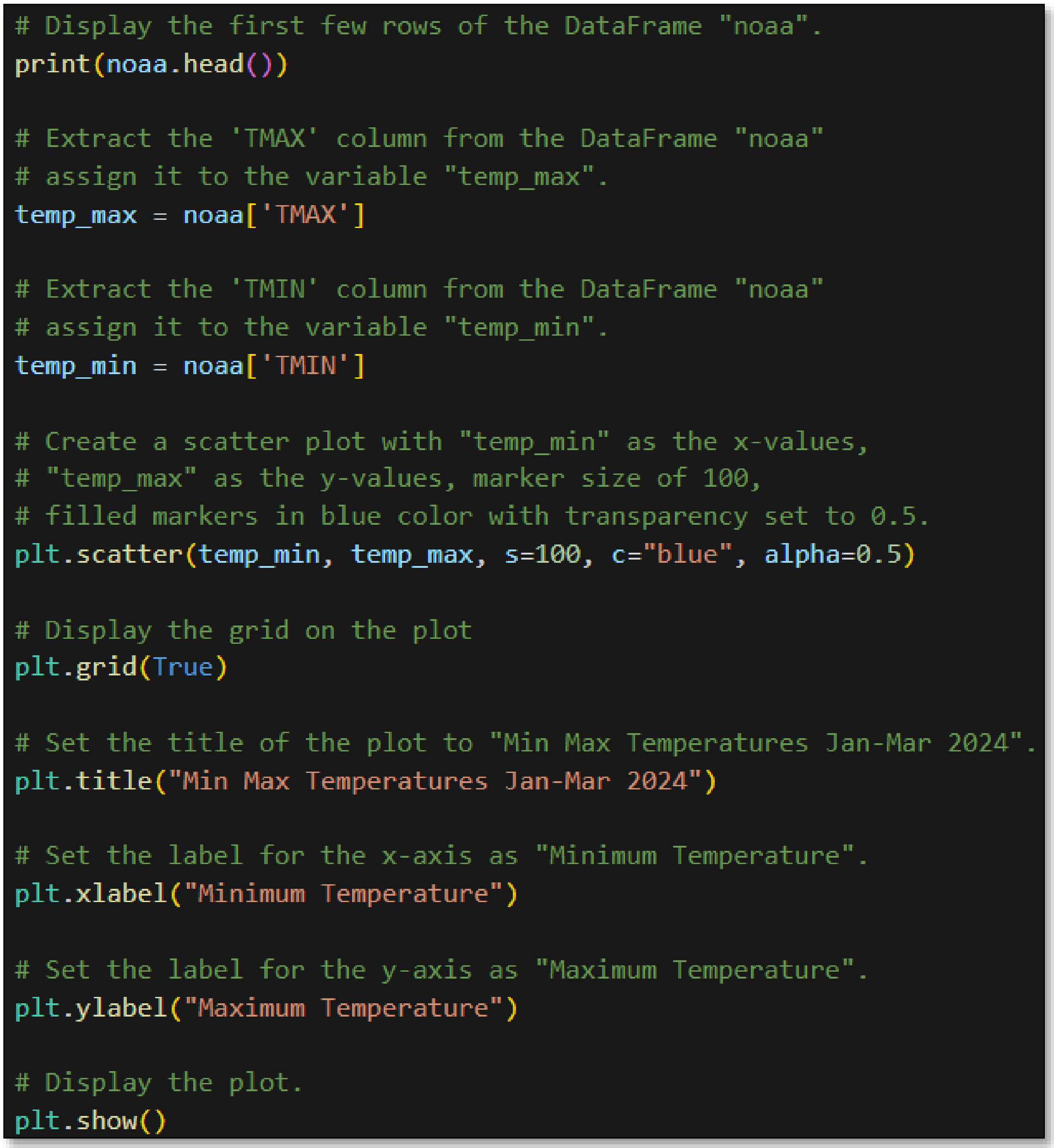


```
0  USW00094948  FAIRMONT MUNICIPAL AIRPORT, MN US  2024-01-01  32.0  19.0
1  USW00094948  FAIRMONT MUNICIPAL AIRPORT, MN US  2024-01-02  35.0  22.0
2  USW00094948  FAIRMONT MUNICIPAL AIRPORT, MN US  2024-01-03  29.0  20.0
3  USW00094948  FAIRMONT MUNICIPAL AIRPORT, MN US  2024-01-04  28.0  21.0
4  USW00094948  FAIRMONT MUNICIPAL AIRPORT, MN US  2024-01-05  35.0  21.0
         STATION                   NAME        DATE  TMAX  TMIN
334  US1MNMT0012  FAIRMONT 1.4 S, MN US  2024-04-19   NaN   NaN
335  US1MNMT0012  FAIRMONT 1.4 S, MN US  2024-04-20   NaN   NaN
336  US1MNMT0012  FAIRMONT 1.4 S, MN US  2024-04-21   NaN   NaN
337  US1MNMT0012  FAIRMONT 1.4 S, MN US  2024-04-22   NaN   NaN
338  US1MNMT0012  FAIRMONT 1.4 S, MN US  2024-04-23   NaN   NaN
```



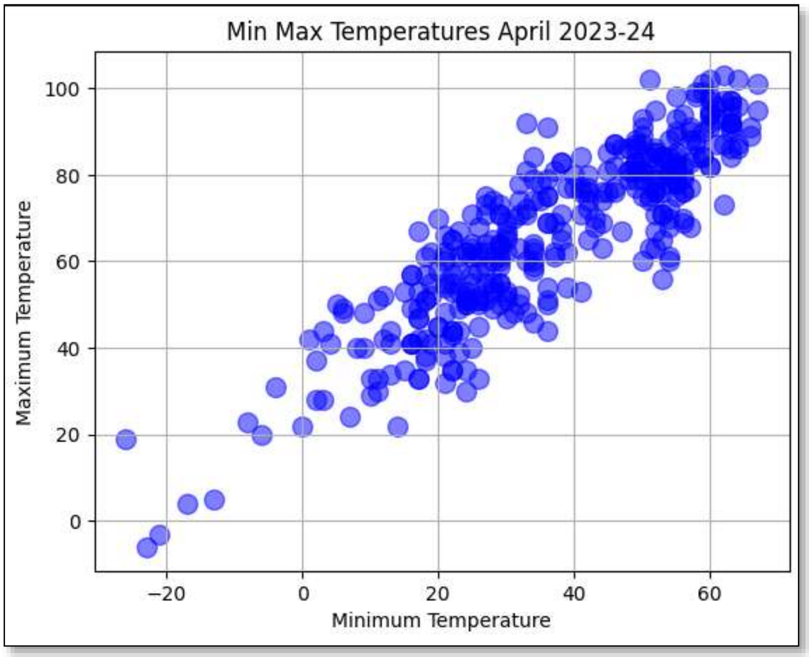
```python
# Filter data to a single weather station
STATION = "USW00094948"
noaa = noaa.loc[noaa["STATION"].str.startswith(STATION)]
display(noaa.tail())
```

This section of code will extract and display a scatter chart.



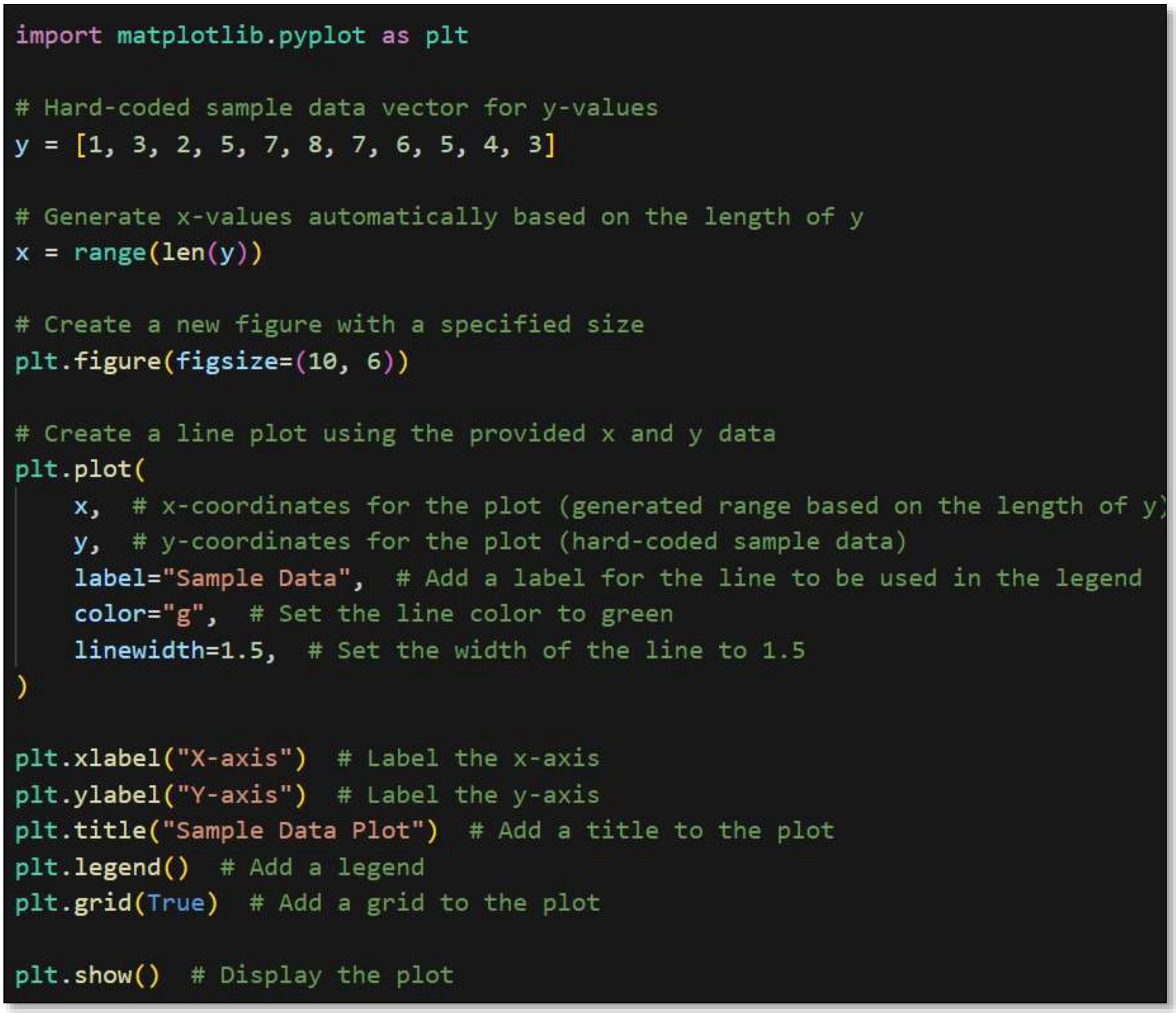
```
# Display the first few rows of the DataFrame "noaa".
print(noaa.head())

# Extract the 'TMAX' column from the DataFrame "noaa"
# assign it to the variable "temp_max".
temp_max = noaa['TMAX']

# Extract the 'TMIN' column from the DataFrame "noaa"
# assign it to the variable "temp_min".
temp_min = noaa['TMIN']

# Create a scatter plot with "temp_min" as the x-values,
# "temp_max" as the y-values, marker size of 100,
# filled markers in blue color with transparency set to 0.5.
plt.scatter(temp_min, temp_max, s=100, c="blue", alpha=0.5)

# Display the grid on the plot
plt.grid(True)

# Set the title of the plot to "Min Max Temperatures Jan-Mar 2024".
plt.title("Min Max Temperatures Jan-Mar 2024")

# Set the label for the x-axis as "Minimum Temperature".
plt.xlabel("Minimum Temperature")

# Set the label for the y-axis as "Maximum Temperature".
plt.ylabel("Maximum Temperature")

# Display the plot.
plt.show()
```

Example run:



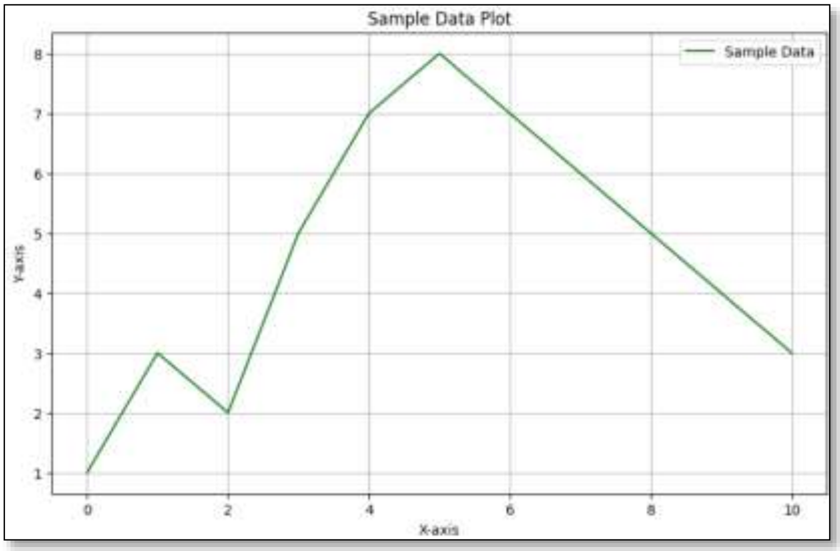
Tutorial 2: Line and Bar Plots with Python

plt.plot is a line plot. plt.bar is a bar plot.

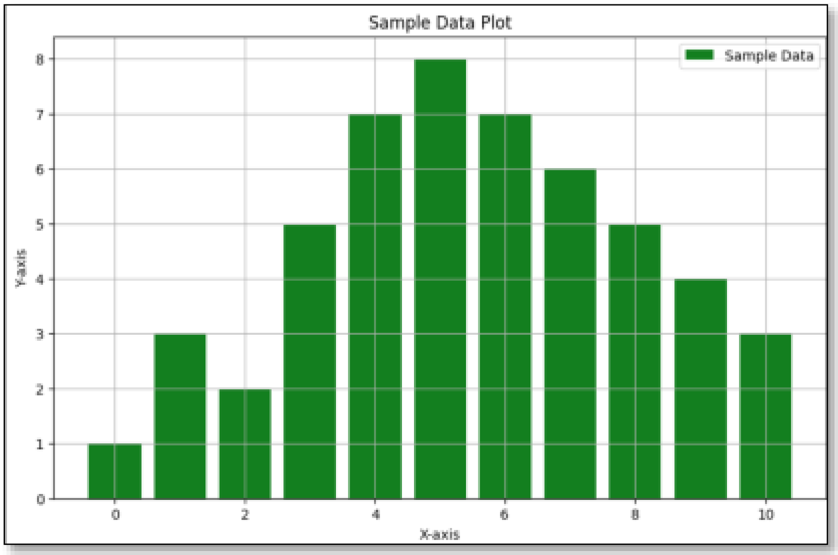


```python
import matplotlib.pyplot as plt

# Hard-coded sample data vector for y-values
y = [1, 3, 2, 5, 7, 8, 7, 6, 5, 4, 3]

# Generate x-values automatically based on the length of y
x = range(len(y))

# Create a new figure with a specified size
plt.figure(figsize=(10, 6))

# Create a line plot using the provided x and y data
plt.plot(
    x,  # x-coordinates for the plot (generated range based on the length of y)
    y,  # y-coordinates for the plot (hard-coded sample data)
    label="Sample Data",  # Add a label for the line to be used in the legend
    color="g",  # Set the line color to green
    linewidth=1.5,  # Set the width of the line to 1.5
)

plt.xlabel("X-axis")  # Label the x-axis
plt.ylabel("Y-axis")  # Label the y-axis
plt.title("Sample Data Plot")  # Add a title to the plot
plt.legend()  # Add a legend
plt.grid(True)  # Add a grid to the plot

plt.show()  # Display the plot
```

Example run:



To change this plot to a bar plot, change **plt.plot** to **plt.bar**

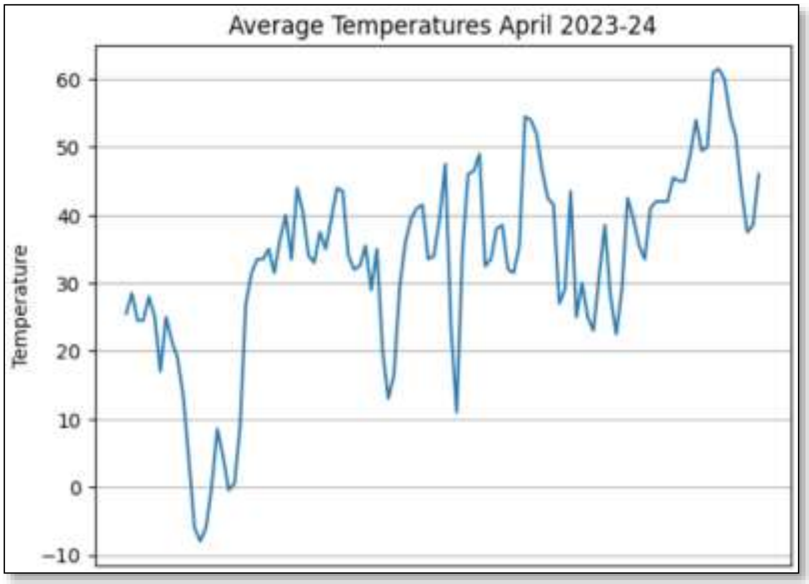


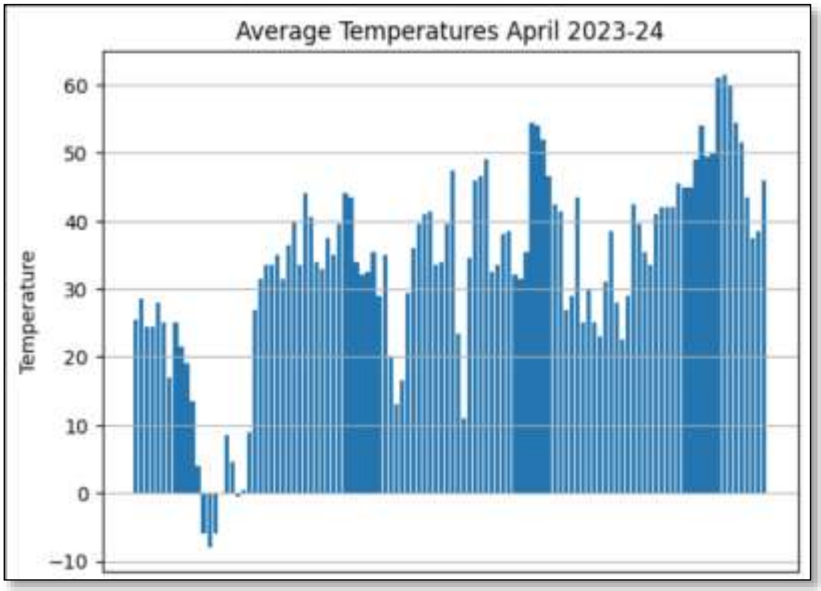
Assignment 1: Weather Plots with Python

You may have to do some research to figure out how to do these. You can use other readings if you wish.

1. Calculate the average temperatures using TMIN and TMAX.
2. Create a line plot showing the average temperatures.
3. Create a bar plot showing the average temperatures.

Example run:





Assignment Submission

1. In Google Colab → Click the Share button in the upper right hand side.

   a. Change General Access → Anyone with the link → Click Copy link.

2. Attach all to the assignment in Blackboard.