

# Python Data Science

## Contents

Python Data Science.....	1
Tutorial 1: Plot Daily Weather in Scottsbluff.....	1
Python Tutorial 1: Plot Daily Weather in Scottsbluff .....	3
Assignment 1: Weather Plots with Python .....	5
Assignment Submission.....	5

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\_2024.csv** in one of your GitHub repositories.

19. Commit the repository.

20. Got to the repository at [www.github.com](https://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

	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.

[https://raw.githubusercontent.com/itinstructor/JupyterNotebooks/main/Datasets/noaa\\_scottsbluff\\_2024.csv](https://raw.githubusercontent.com/itinstructor/JupyterNotebooks/main/Datasets/noaa_scottsbluff_2024.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())
```

```
# Extract the 'TMAX' and 'TMIN' columns from the DataFrame "noaa"
# assign to variables
temp_max = noaa['TMAX']
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)

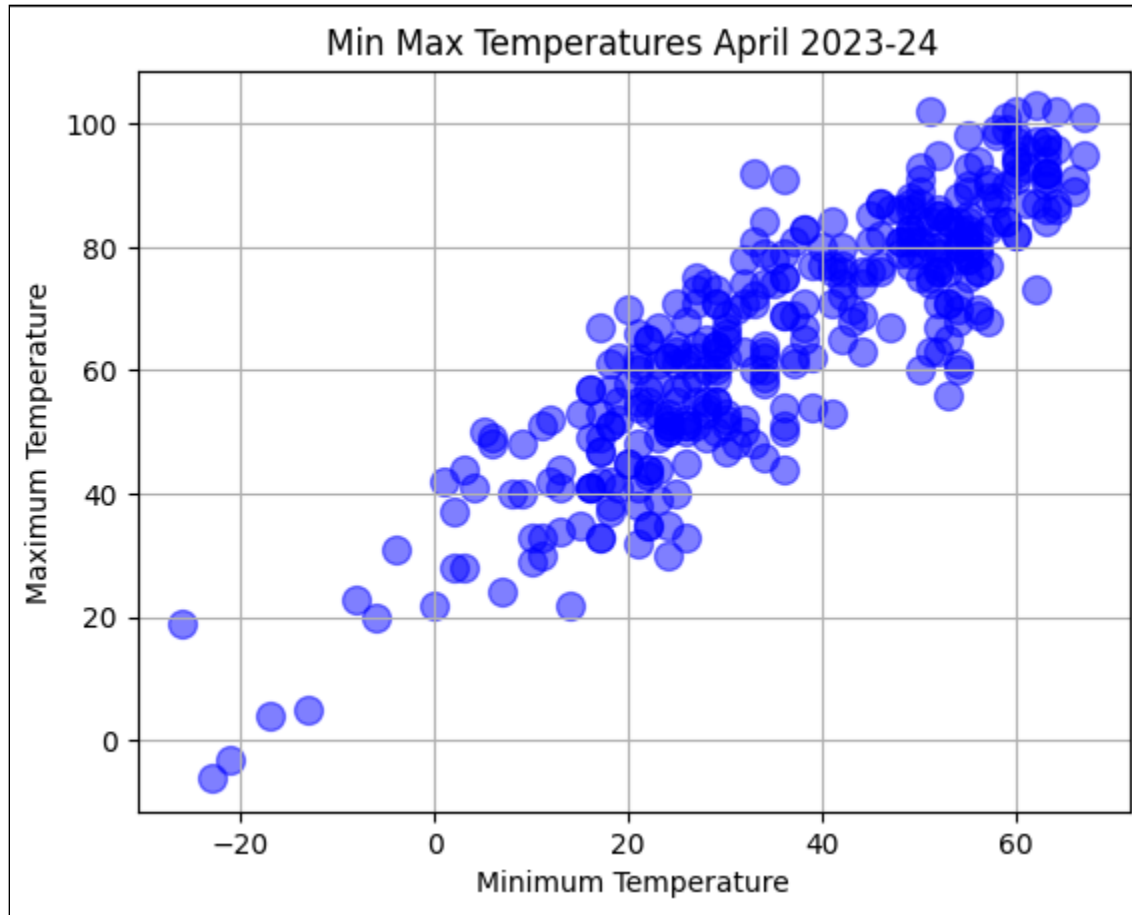
# Set the title of the plot to "Min Max Temperatures Jan-Mar 2024".
plt.title("Min Max Temperatures April 2023-24")

# 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:



## Assignment 1: Weather Plots with Python

You may have to do some research to figure out how to do these.

1. Create a line plot showing the TMIN and TMAX temperatures on the same plot.
2. Create a line plot showing the TAVG.
3. Create a bar plot showing the TAVG.

---

## Assignment Submission

1. Insert a shared link to your Google Colab notebook that anyone can open.
2. Attach all to the assignment in Blackboard.