

Tuesday, October 27, 2020 | **Class #7**

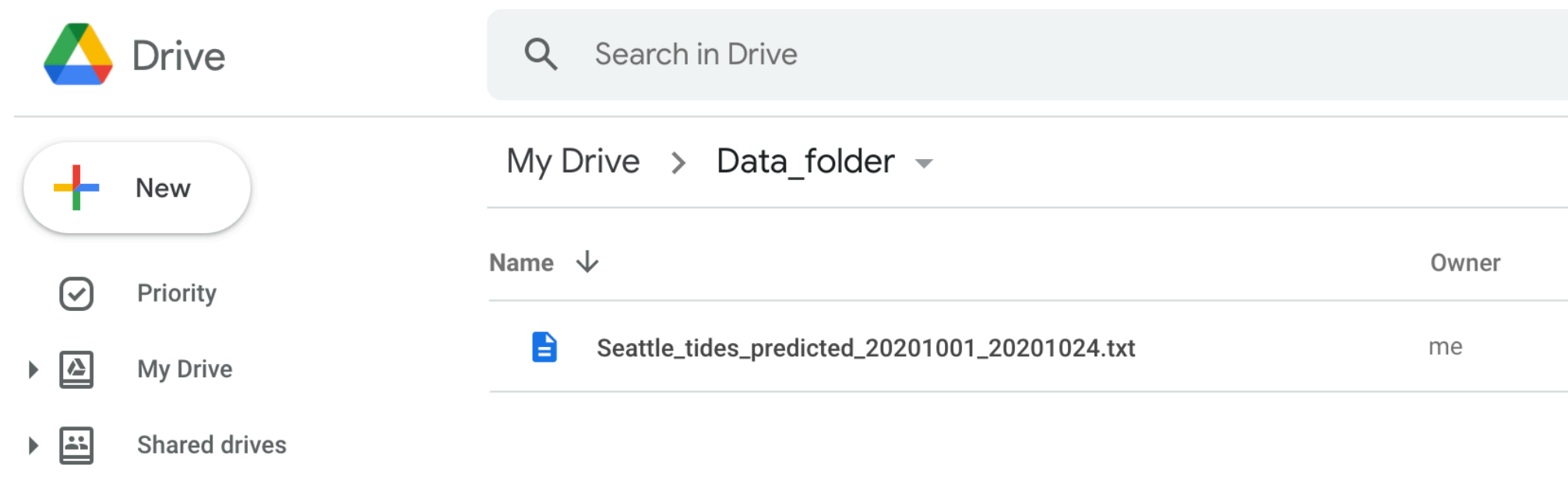
# File management, loading data, and plotting

OCEAN 215 | Autumn 2020

Ethan Campbell and **Katy Christensen**

# Getting data in Google Colab

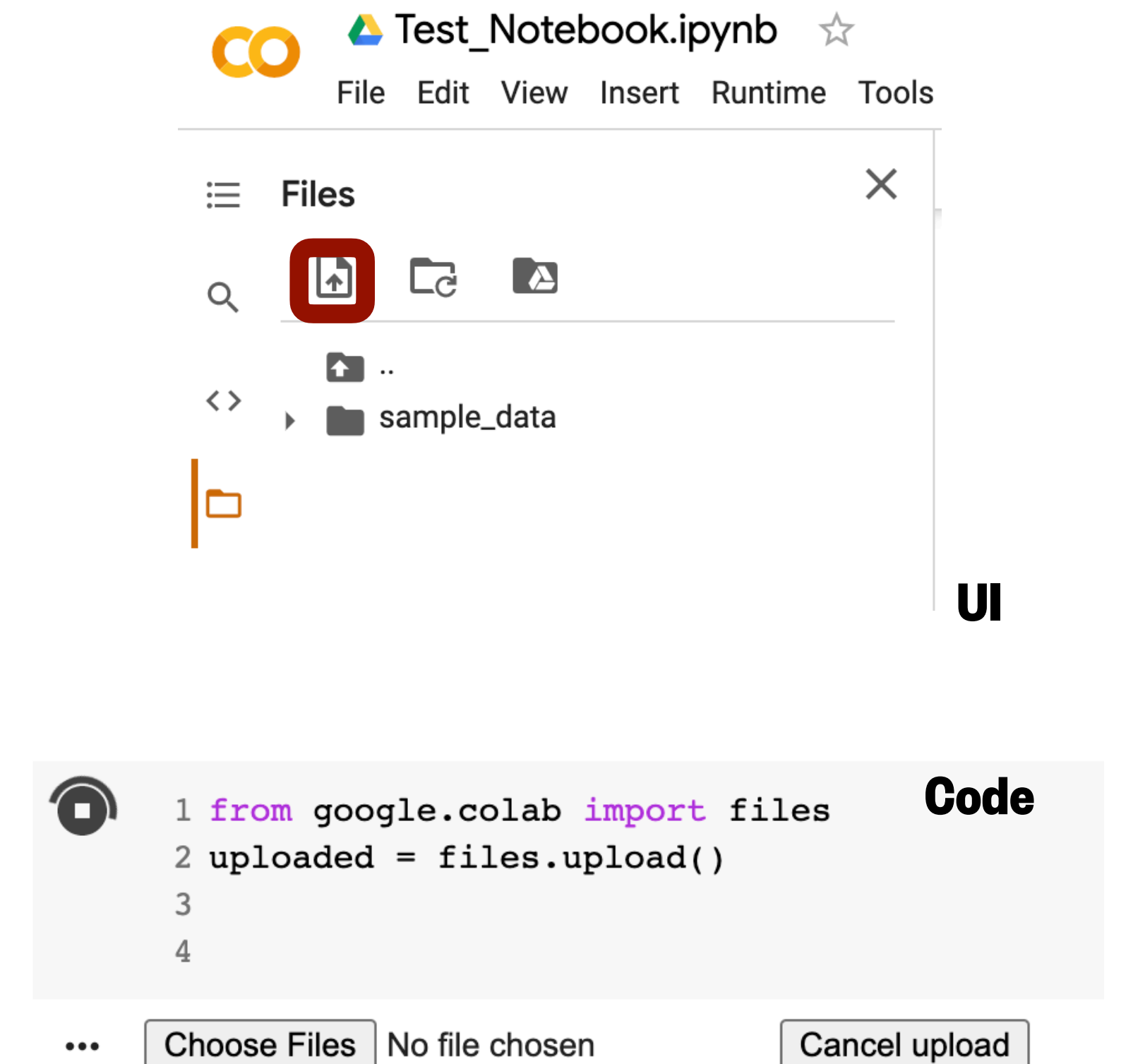
Upload the data file to your Google Drive data folder...



Then mount your Google Drive.



Or upload directly to Google Colab.



# Activity: getting data

---

1. Select any station from NOAA Tides and Currents to get tide prediction data.
2. Select a time range with 6 minute intervals (use STND as your datum).
3. Download the data as a txt file.
4. Have each person in your group access the file using a different method.
  1. Upload to Google colab
    1. User Interface (UI)
    2. Code
  2. Mount your Google Drive
    1. User Interface (UI)
    2. Code
5. Find how many header lines are in your file using **readline()** and a for loop.
6. Try restarting your runtime. Is your file still available?
7. Discuss which method of accessing data you prefer.



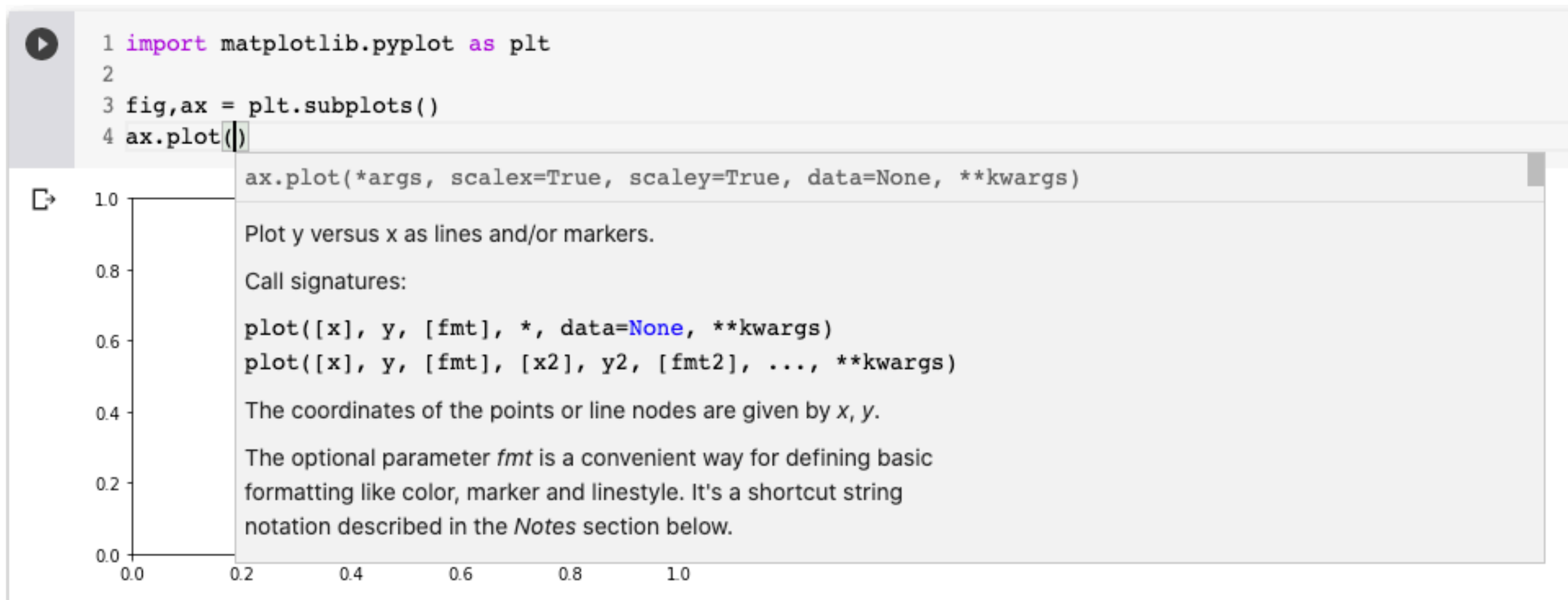
# Python is highly adaptable

**You can change anything if you know how.**

You can usually find how to do something by searching the documentation (API) or searching the internet.

Google Colab has a built in API:

```
1 import matplotlib.pyplot as plt
2
3 fig,ax = plt.subplots()
4 ax.plot(|)
```



The screenshot shows a Google Colab interface. At the top, a code cell contains the following Python code:

```
1 import matplotlib.pyplot as plt
2
3 fig,ax = plt.subplots()
4 ax.plot(|)
```

Below the code cell, a plot is displayed with a white background and a gray border. The x-axis ranges from 0.0 to 1.0 with major ticks every 0.2. The y-axis ranges from 0.0 to 1.0 with major ticks every 0.2. The plot area is currently empty. To the right of the plot, a tooltip or documentation window is open, showing the signature and description of the `ax.plot()` method:

```
ax.plot(*args, scalex=True, scaley=True, data=None, **kwargs)
```

Plot y versus x as lines and/or markers.

Call signatures:

```
plot([x], y, [fmt], *, data=None, **kwargs)
plot([x], y, [fmt], [x2], y2, [fmt2], ..., **kwargs)
```

The coordinates of the points or line nodes are given by `x, y`.

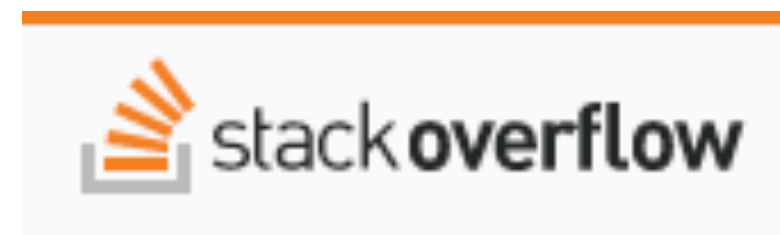
The optional parameter `fmt` is a convenient way for defining basic formatting like color, marker and linestyle. It's a shortcut string notation described in the *Notes* section below.

**Official matplotlib documentation:**

<https://matplotlib.org/3.3.2/index.html>

**Official numpy documentation:**

<https://numpy.org/doc/stable/reference/index.html>

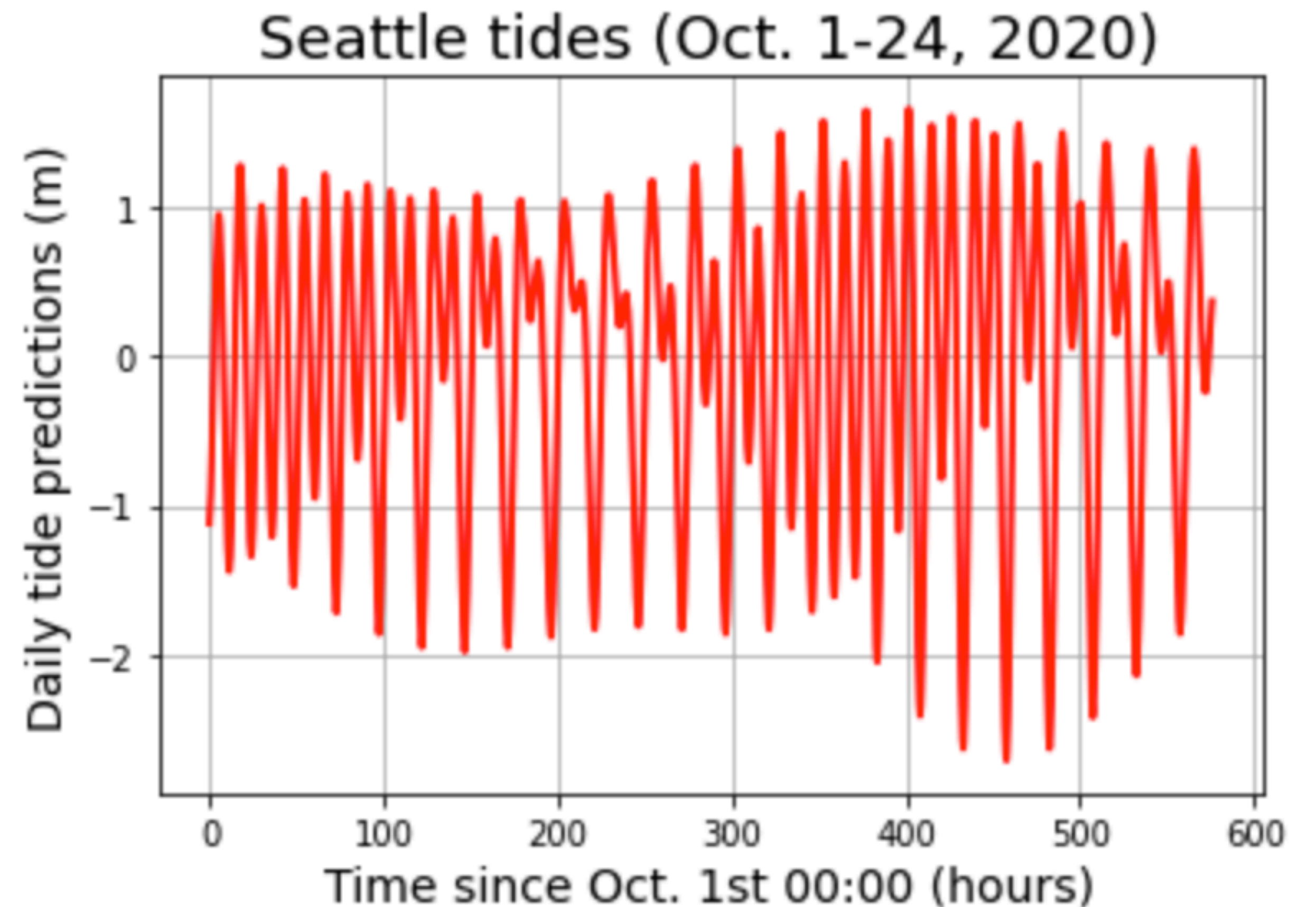


<https://stackoverflow.com/>

# Activity: loading and plotting data

---

1. Using `np.genfromtxt`, extract the predicted tide from the file you just downloaded (column 3).
2. Using the `np.linspace` function, create an array with the same shape as the tide data that gives the hours of data (e.g. `len(data)/10`).
3. Make a plot of the data, customizing the following:
  1. Line color
  2. Line width
  3. Line style
  4. Marker shape
  5. Marker color



**(Make sure your plot has all of the required plot elements!)**