PH 366 Day 6:
Data Sets and Electric Potential

27 Jan 2025

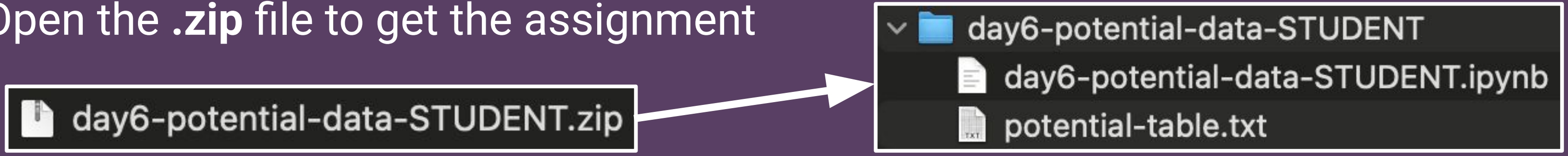
Announcements

Midterm mini-project next week

- Open-ended, more complex version of a regular assignment
- Monday and Wednesday classes set aside for project work time
- Collaboration and online resources allowed, but submit your own work
- Detailed description and grading rubric on Canvas

Locating a Data File

Open the **.zip** file to get the assignment



day6-potential-data-STUDENT.zip

day6-potential-data-STUDENT
- day6-potential-data-STUDENT.ipynb
- potential-table.txt

Keep the **.ipynb** and **.txt** files in the same folder

- Enables computer to locate **.txt** file by filename alone:

  `“potential-table.txt”`

Reading Files with pandas

pandas: library for data analysis and manipulation

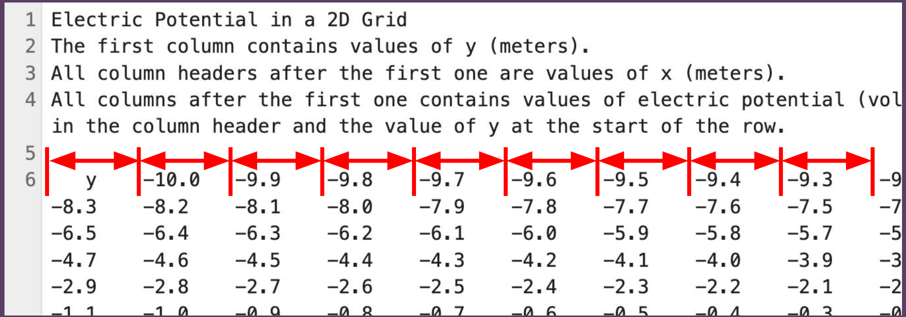
```
import pandas as pd
```

Functions `read_csv` and `read_fwf` can read data sets depending on the file format

- **csv:** comma-separated values (or separated by some other character)
- **fwf:** fixed-width file (each data point typed to take up same amount of space when reading)

Reading a Fixed-width File

Fixed-width file:



```
Electric Potential in a 2D Grid
The first column contains values of y (meters).
All column headers after the first one are values of x (meters).
All columns after the first one contains values of electric potential (vol
in the column header and the value of y at the start of the row.

    y   -10.0  -9.9   -9.8   -9.7   -9.6   -9.5   -9.4   -9.3   -9
-8.3   -8.2   -8.1   -8.0   -7.9   -7.8   -7.7   -7.6   -7.5   -7
-6.5   -6.4   -6.3   -6.2   -6.1   -6.0   -5.9   -5.8   -5.7   -5
-4.7   -4.6   -4.5   -4.4   -4.3   -4.2   -4.1   -4.0   -3.9   -3
-2.9   -2.8   -2.7   -2.6   -2.5   -2.4   -2.3   -2.2   -2.1   -2
-1.1   -1.0   -0.9   -0.8   -0.7   -0.6   -0.5   -0.4   -0.3   -0
```

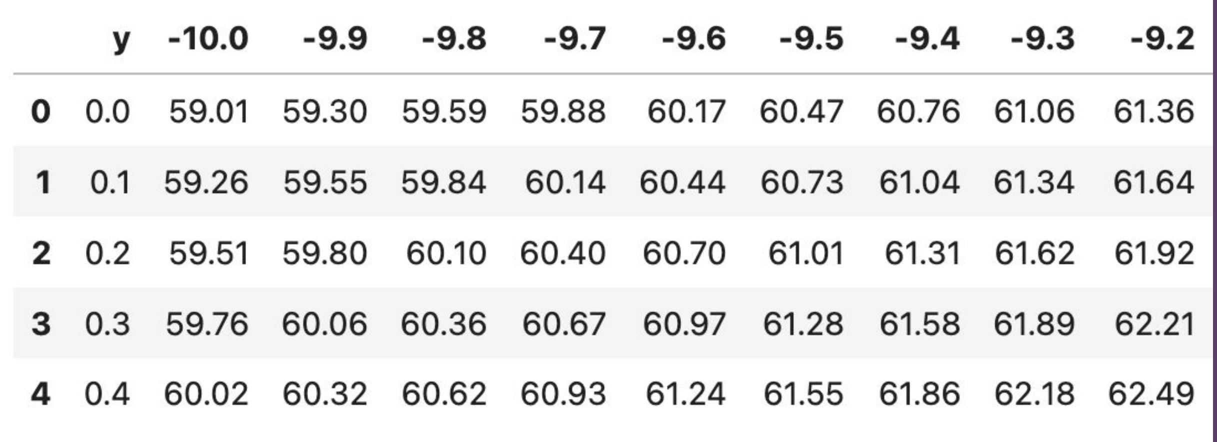
Notice…

- Each data point takes up a fixed amount of space
- The first five rows do not contain data at all, just information about what the file contains

```
data = pd.read_fwf("file-name.txt", skiprows=5)
```

Viewing Data with pandas

```
data.head()
```

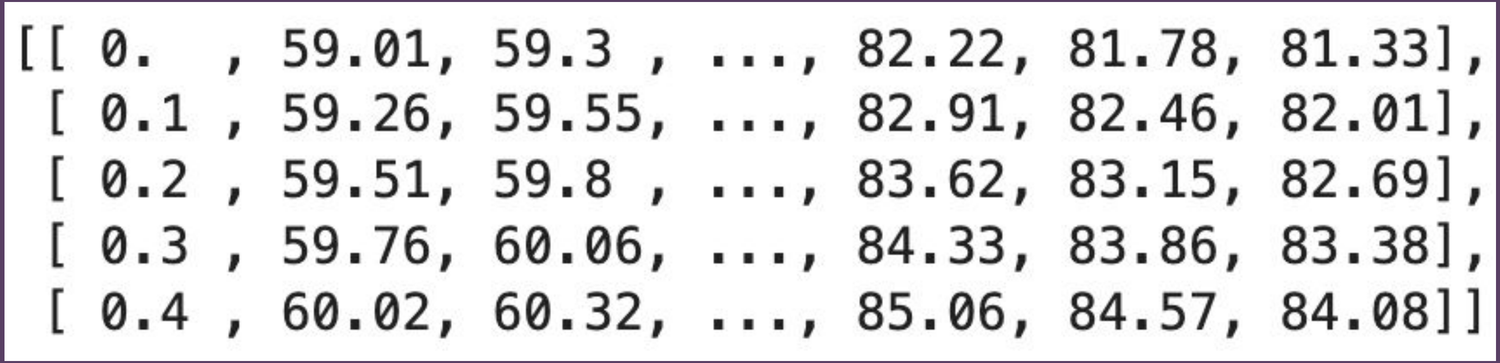


	y	-10.0	-9.9	-9.8	-9.7	-9.6	-9.5	-9.4	-9.3	-9.2
0	0.0	59.01	59.30	59.59	59.88	60.17	60.47	60.76	61.06	61.36
1	0.1	59.26	59.55	59.84	60.14	60.44	60.73	61.04	61.34	61.64
2	0.2	59.51	59.80	60.10	60.40	60.70	61.01	61.31	61.62	61.92
3	0.3	59.76	60.06	60.36	60.67	60.97	61.28	61.58	61.89	62.21
4	0.4	60.02	60.32	60.62	60.93	61.24	61.55	61.86	62.18	62.49

Indexing a 2D Array

Get your whiteboards out…

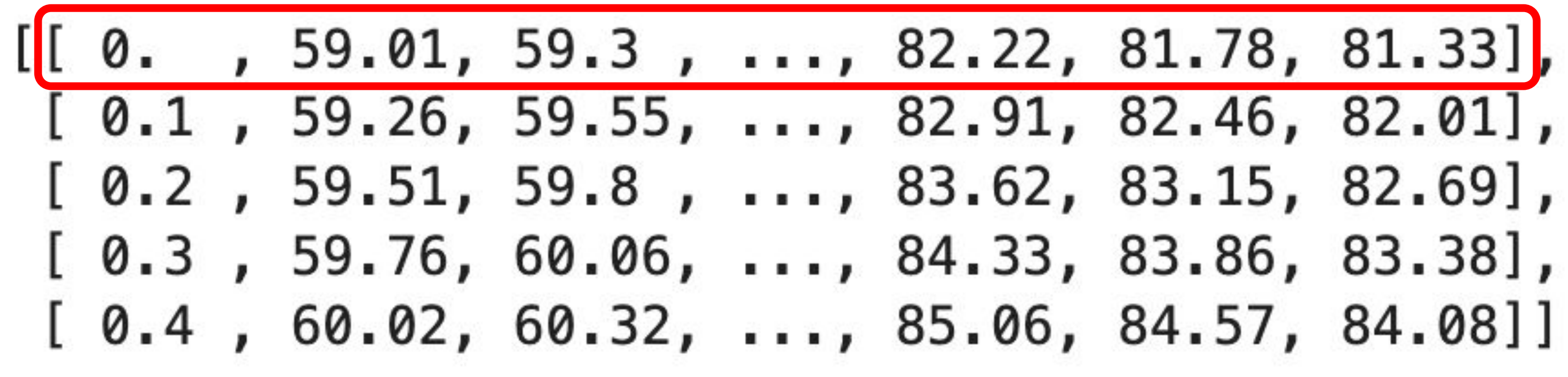
We are looking at the same data in 2D NumPy array form



```
[[ 0.   , 59.01, 59.3 , ..., 82.22, 81.78, 81.33],
 [ 0.1 , 59.26, 59.55, ..., 82.91, 82.46, 82.01],
 [ 0.2 , 59.51, 59.8 , ..., 83.62, 83.15, 82.69],
 [ 0.3 , 59.76, 60.06, ..., 84.33, 83.86, 83.38],
 [ 0.4 , 60.02, 60.32, ..., 85.06, 84.57, 84.08]]
```

Indexing a 2D Array

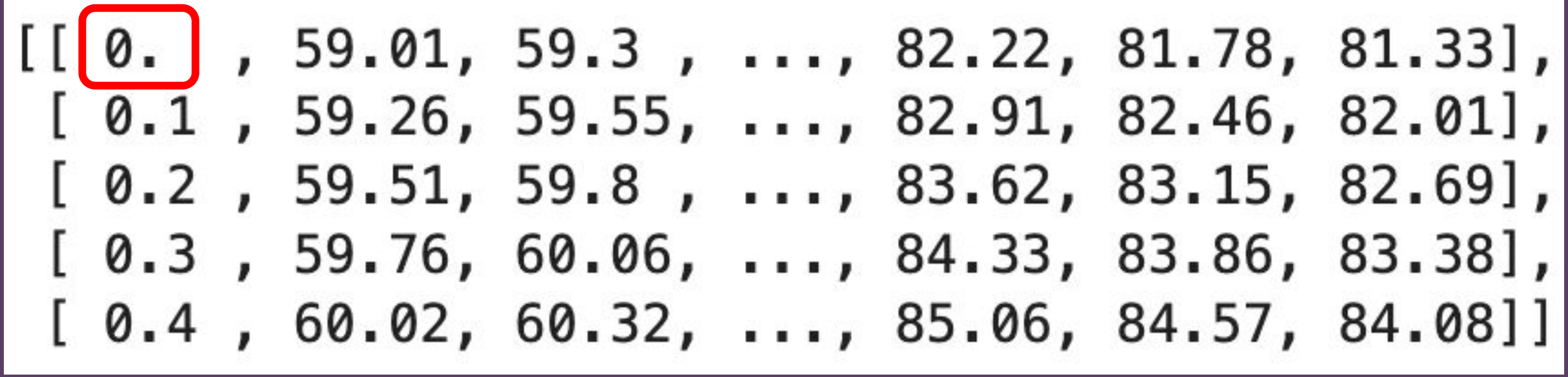
The same data in 2D NumPy array form



```
[[ 0.  , 59.01, 59.3 , ..., 82.22, 81.78, 81.33],
 [ 0.1 , 59.26, 59.55, ..., 82.91, 82.46, 82.01],
 [ 0.2 , 59.51, 59.8 , ..., 83.62, 83.15, 82.69],
 [ 0.3 , 59.76, 60.06, ..., 84.33, 83.86, 83.38],
 [ 0.4 , 60.02, 60.32, ..., 85.06, 84.57, 84.08]]
```

Indexing a 2D Array

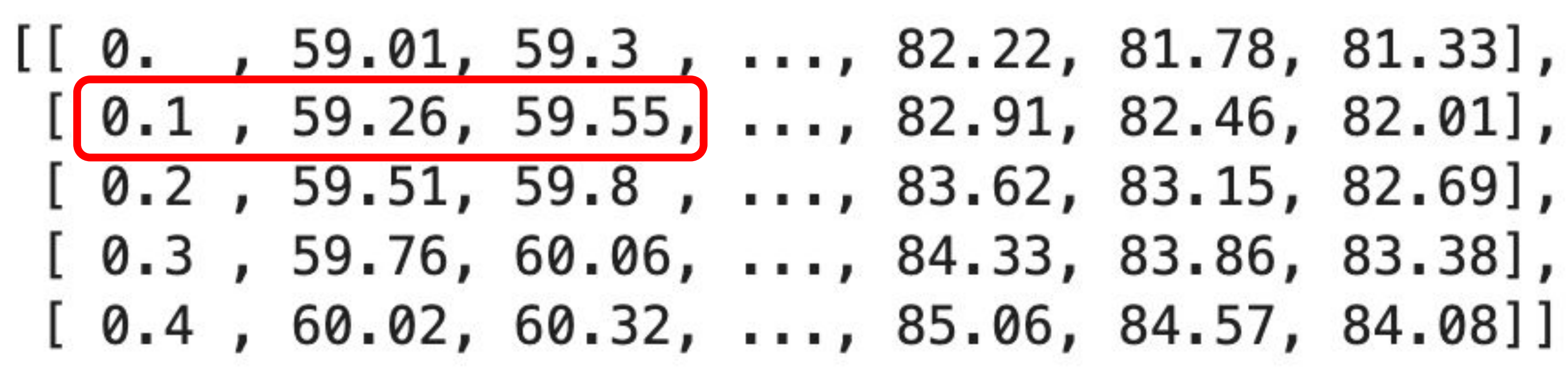
The same data as a NumPy array – 5 rows, 202 columns



```
[[ 0.  , 59.01, 59.3 , ..., 82.22, 81.78, 81.33],
 [ 0.1 , 59.26, 59.55, ..., 82.91, 82.46, 82.01],
 [ 0.2 , 59.51, 59.8 , ..., 83.62, 83.15, 82.69],
 [ 0.3 , 59.76, 60.06, ..., 84.33, 83.86, 83.38],
 [ 0.4 , 60.02, 60.32, ..., 85.06, 84.57, 84.08]]
```

Indexing a 2D Array

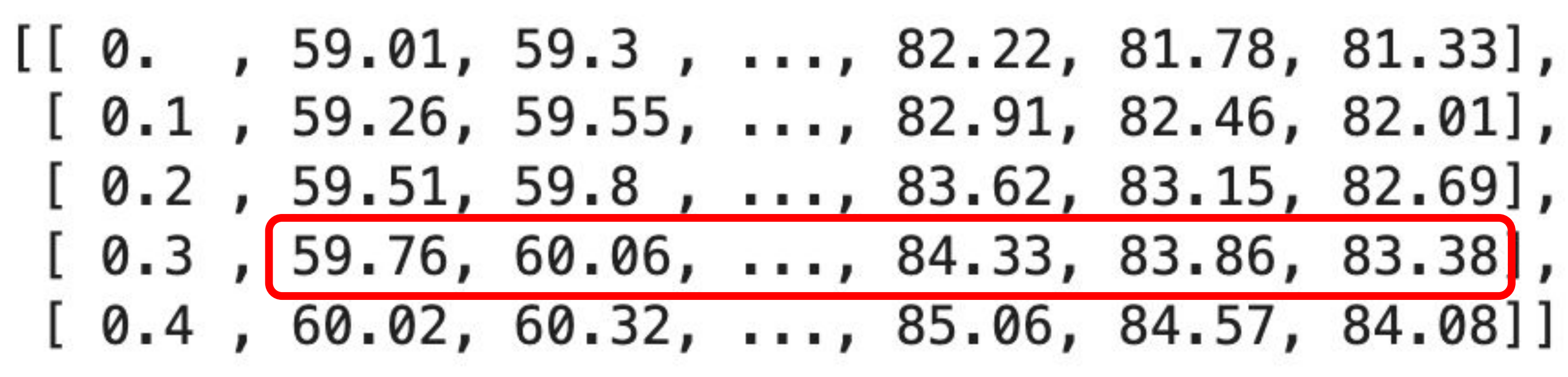
The same data as a NumPy array – 5 rows, 202 columns



```
[[ 0.   , 59.01, 59.3 , ..., 82.22, 81.78, 81.33],
 [ 0.1  , 59.26, 59.55, ..., 82.91, 82.46, 82.01],
 [ 0.2  , 59.51, 59.8 , ..., 83.62, 83.15, 82.69],
 [ 0.3  , 59.76, 60.06, ..., 84.33, 83.86, 83.38],
 [ 0.4  , 60.02, 60.32, ..., 85.06, 84.57, 84.08]]
```

Indexing a 2D Array

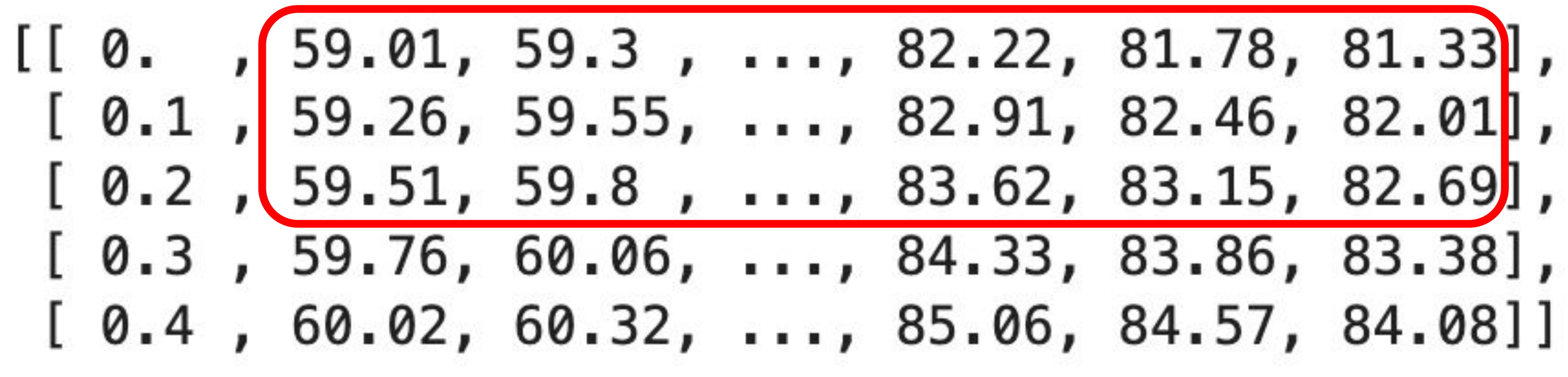
The same data as a NumPy array – 5 rows, 202 columns



```
[[ 0.   , 59.01, 59.3 , ..., 82.22, 81.78, 81.33],
 [ 0.1  , 59.26, 59.55, ..., 82.91, 82.46, 82.01],
 [ 0.2  , 59.51, 59.8 , ..., 83.62, 83.15, 82.69],
 [ 0.3  , 59.76, 60.06, ..., 84.33, 83.86, 83.38],
 [ 0.4  , 60.02, 60.32, ..., 85.06, 84.57, 84.08]]
```

Indexing a 2D Array

The same data as a NumPy array – 5 rows, 202 columns



```
[[ 0.  , 59.01, 59.3 , ..., 82.22, 81.78, 81.33],
 [ 0.1 , 59.26, 59.55, ..., 82.91, 82.46, 82.01],
 [ 0.2 , 59.51, 59.8 , ..., 83.62, 83.15, 82.69],
 [ 0.3 , 59.76, 60.06, ..., 84.33, 83.86, 83.38],
 [ 0.4 , 60.02, 60.32, ..., 85.06, 84.57, 84.08]]
```

Reminder

You are learning new tools today – both accessing data and plotting data

Fill out the **generative AI check-in tasks** to help us design better ways to help you learn computing