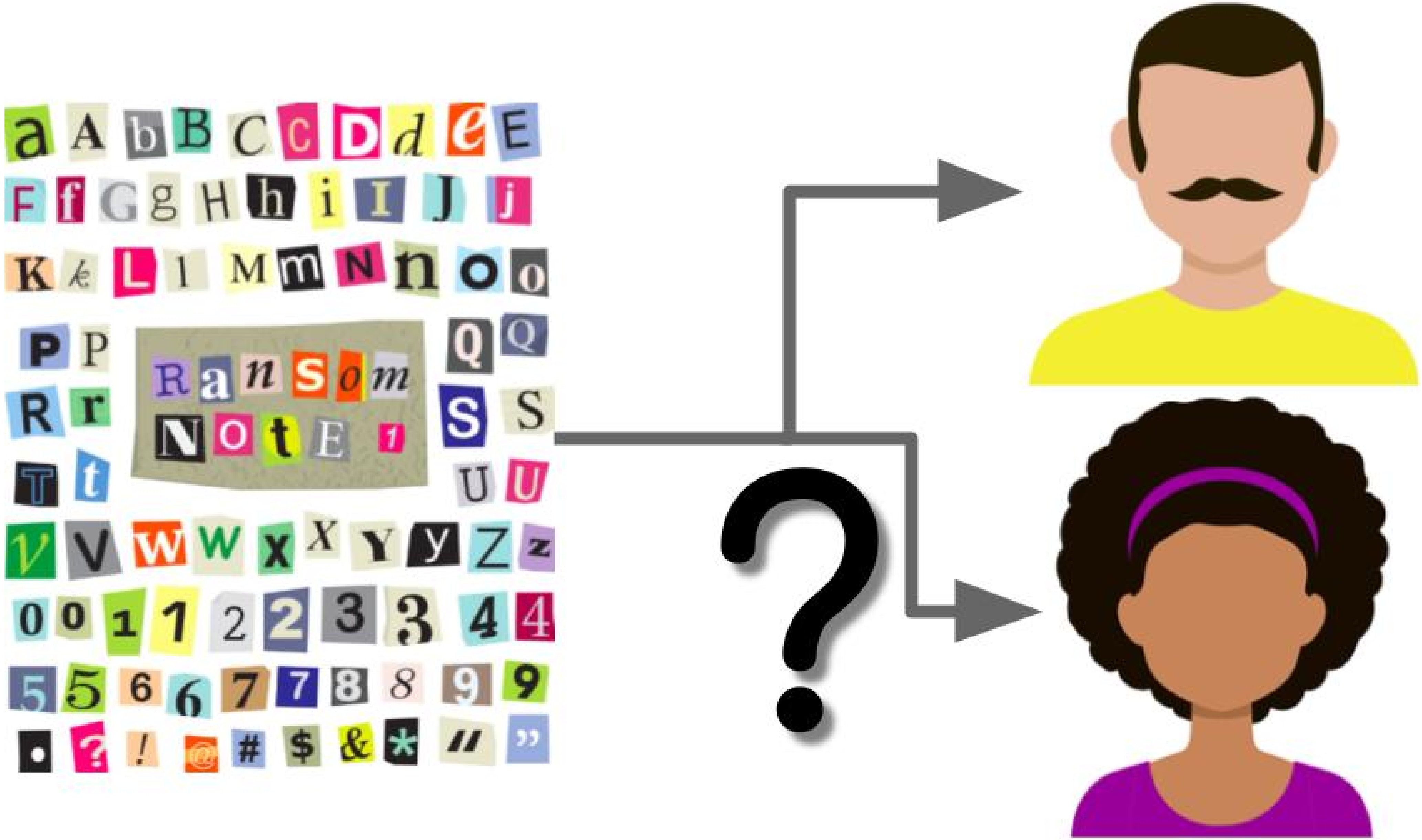
Creating line plots

INTRODUCTION TO DATA SCIENCE IN PYTHON

Hillary Green-Lerman
Lead Data Scientist, Looker

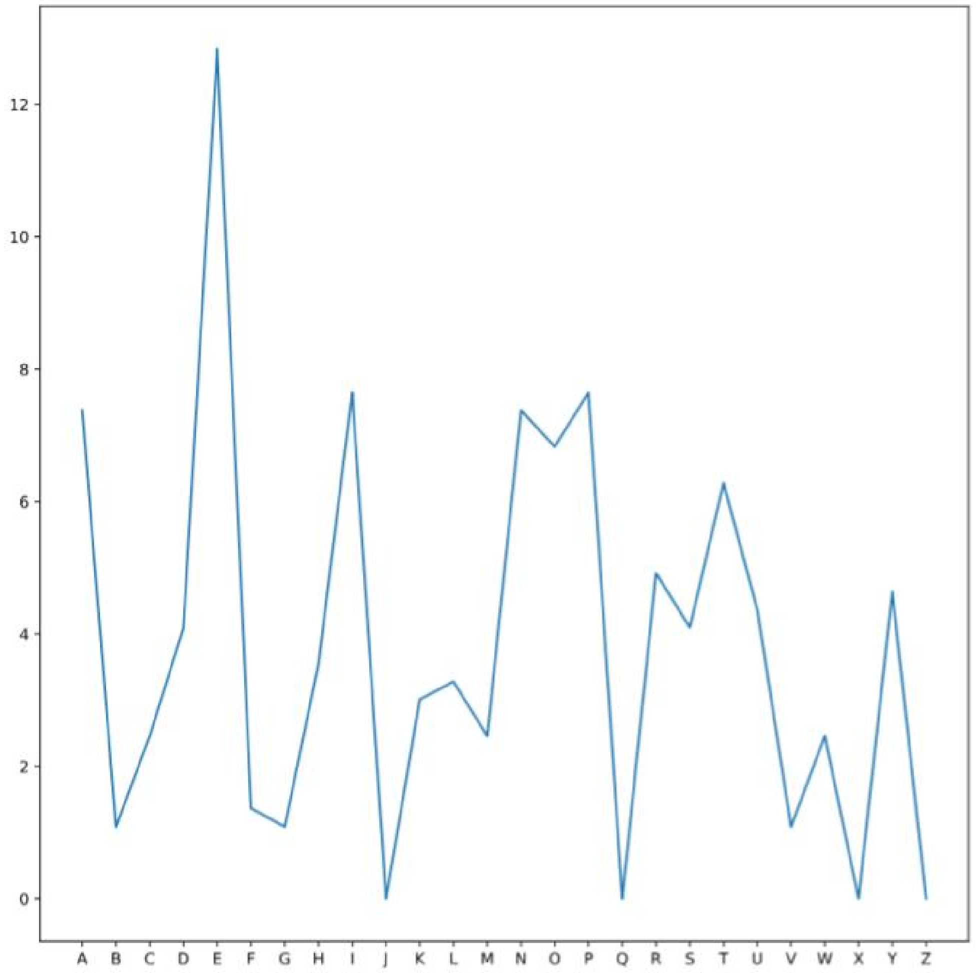
DataCamp

The plot thickens



From DataFrame to Visualization

letter_index	letter	frequency
1	A	7.38
2	B	1.09
3	C	2.46
4	D	4.10
...	...	...

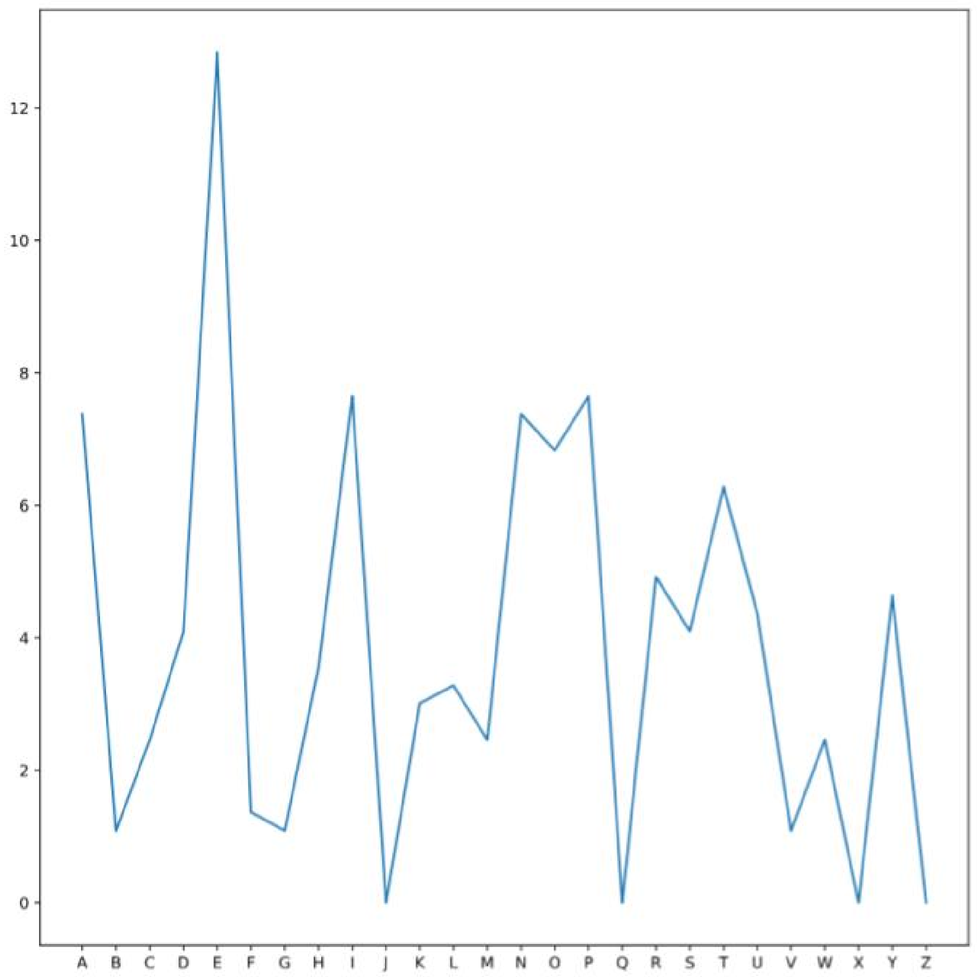


Introducing Matplotlib

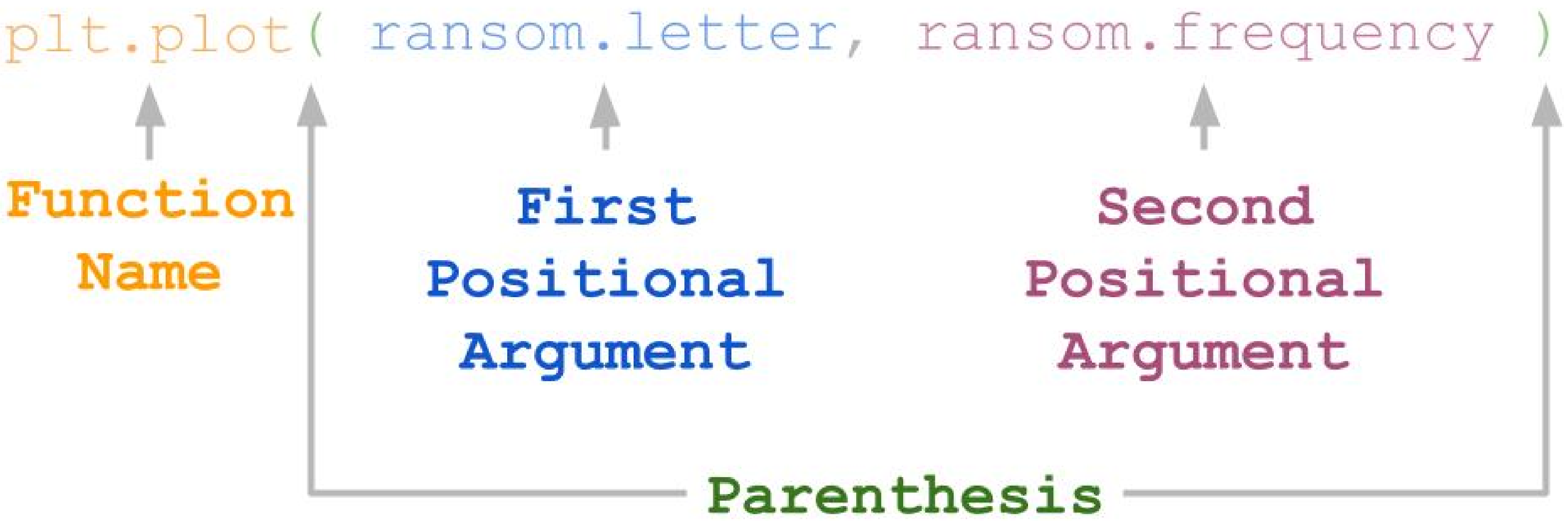
```python
from matplotlib import pyplot as plt
```

```python
plt.plot(x_values, y_values)

plt.show()
```



Line Plot



```
plt.plot( ransom.letter, ransom.frequency )
```

Function Name

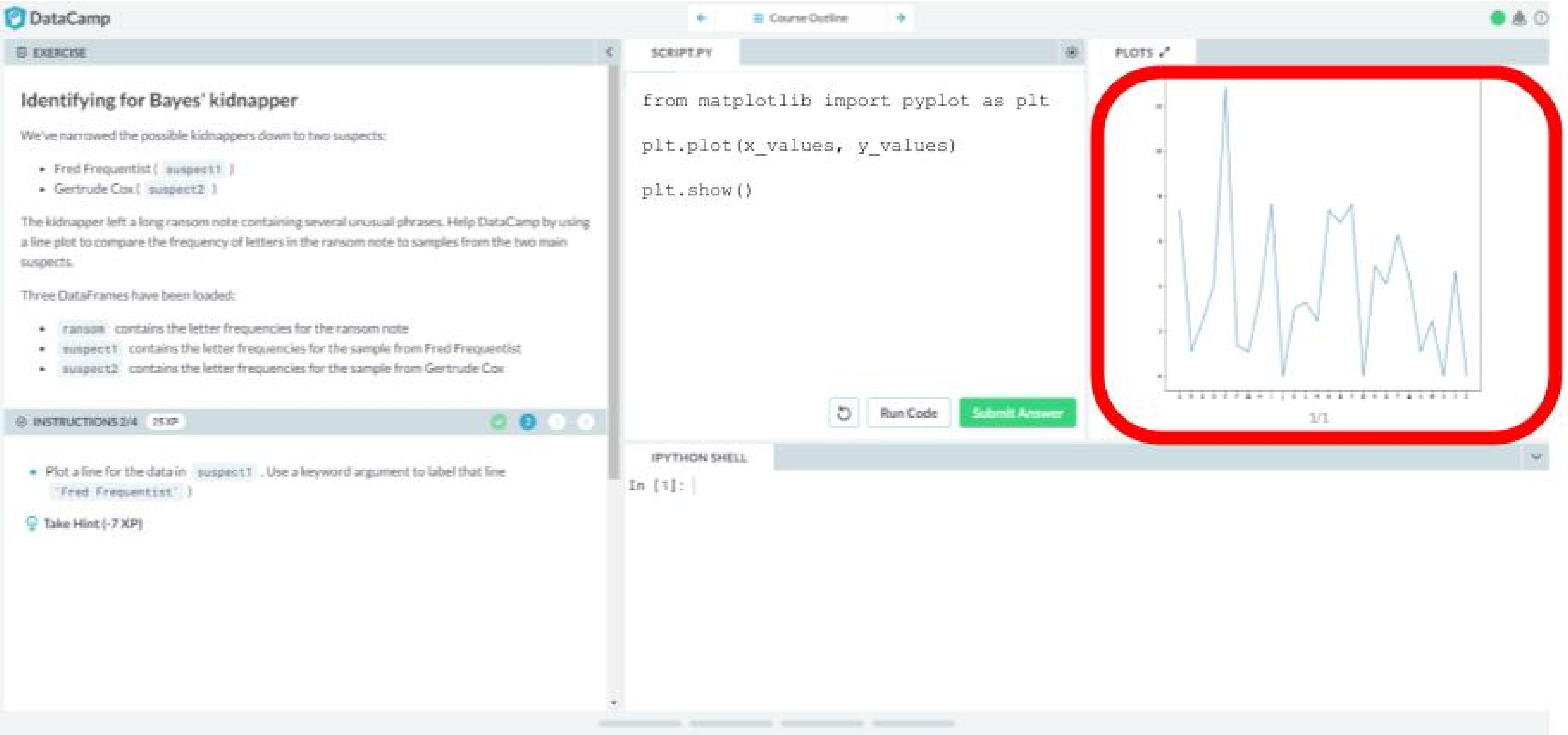
First Positional Argument

Second Positional Argument

Parenthesis

Displaying the Results

```
plt.show()
```



Identifying for Bayes' kidnapper

We've narrowed the possible kidnappers down to two suspects:

- Fred Frequentist (`suspect1`)
- Gertrude Cox (`suspect2`)

The kidnapper left a long ransom note containing several unusual phrases. Help DataCamp by using a line plot to compare the frequency of letters in the ransom note to samples from the two main suspects.

Three DataFrames have been loaded:

- `ransom` contains the letter frequencies for the ransom note
- `suspect1` contains the letter frequencies for the sample from Fred Frequentist
- `suspect2` contains the letter frequencies for the sample from Gertrude Cox

- Plot a line for the data in `suspect1` . Use a keyword argument to label that line `'Fred Frequentist'`)

```
from matplotlib import pyplot as plt

plt.plot(x_values, y_values)

plt.show()
```

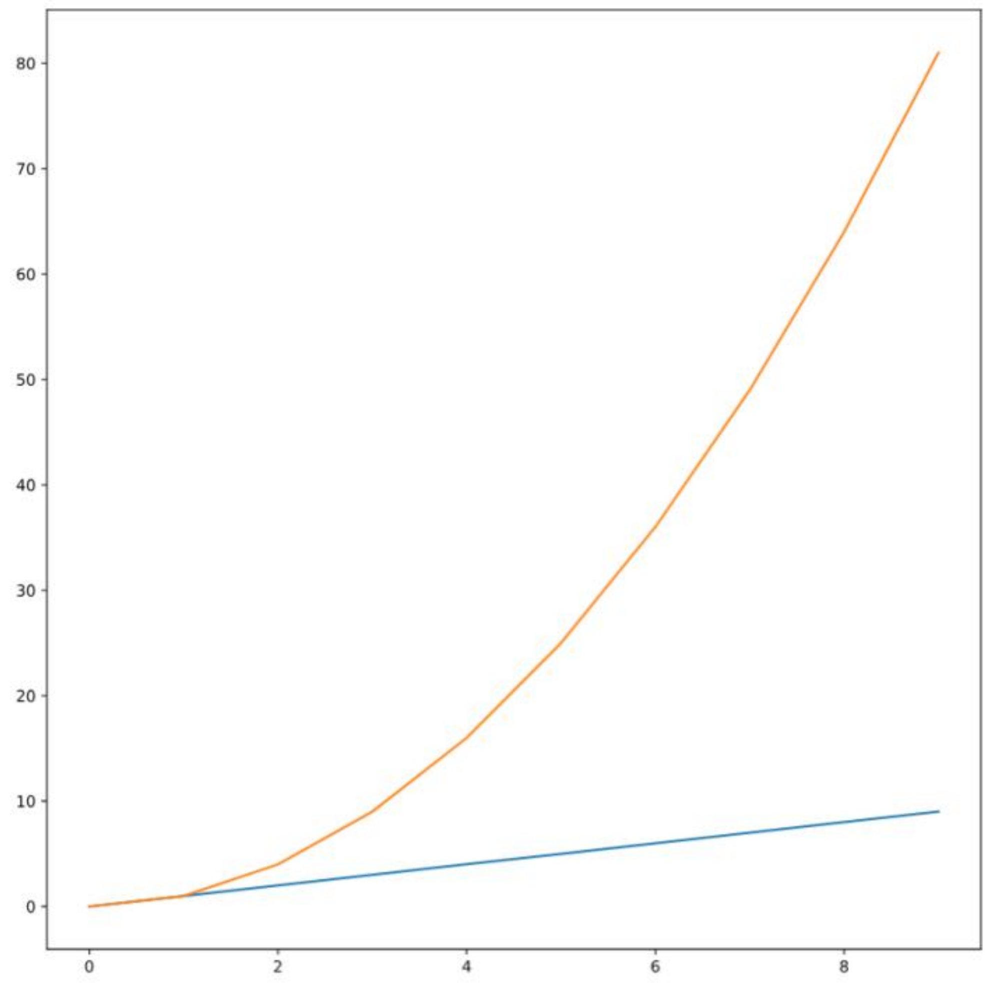
```
In [1]:
```

Multiple Lines

```
plt.plot(data1.x_values,
         data1.y_values)
```

```
plt.plot(data2.x_values,
         data2.y_values)
```

```
plt.show()
```



Let's Practice

INTRODUCTION TO DATA SCIENCE IN PYTHON

Adding labels and legends

INTRODUCTION TO DATA SCIENCE IN PYTHON

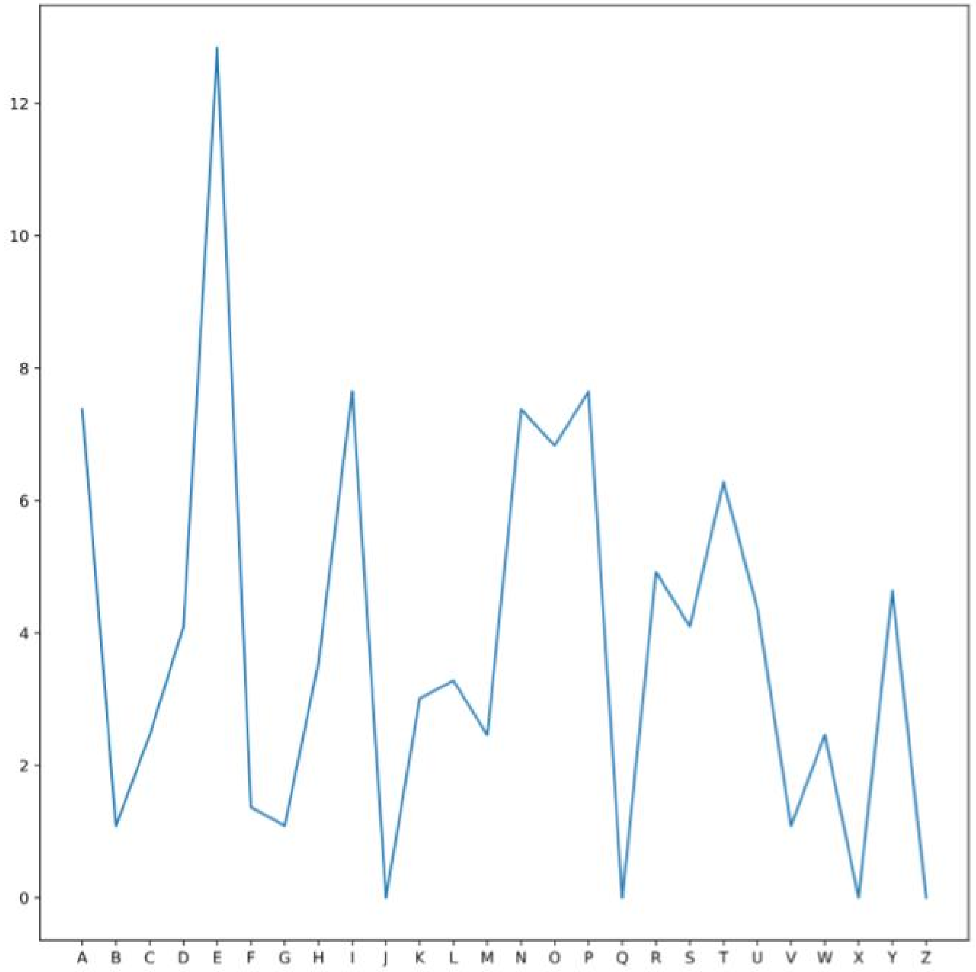
Hillary Green-Lerman
Lead Data Scientist, Looker

What did we just plot?

```
from matplotlib import pyplot as plt

plt.plot(ransom.letter,
         ransom.frequency)

plt.show()
```



Axes and title labels

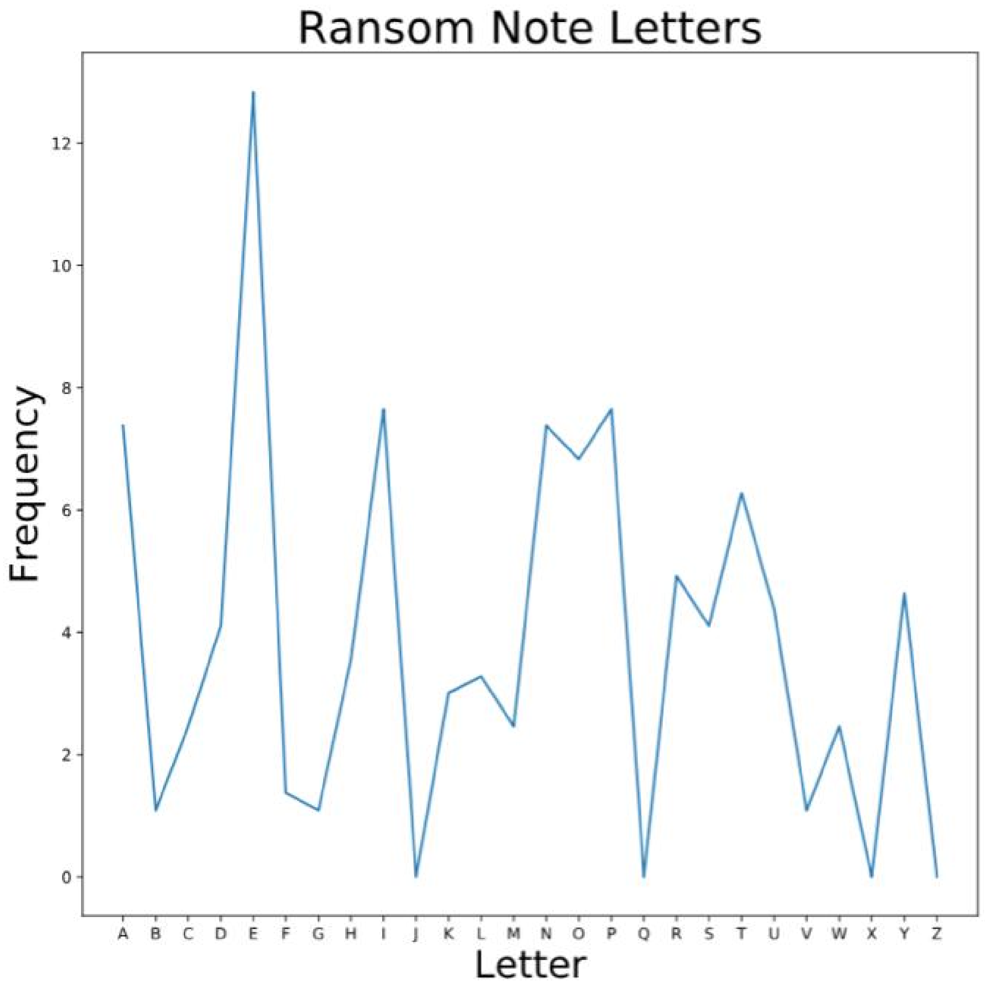
```
plt.xlabel("Letter")
```

```
plt.ylabel("Frequency")
```

```
plt.title("Ransom Note Letters")
```

Labels anywhere before

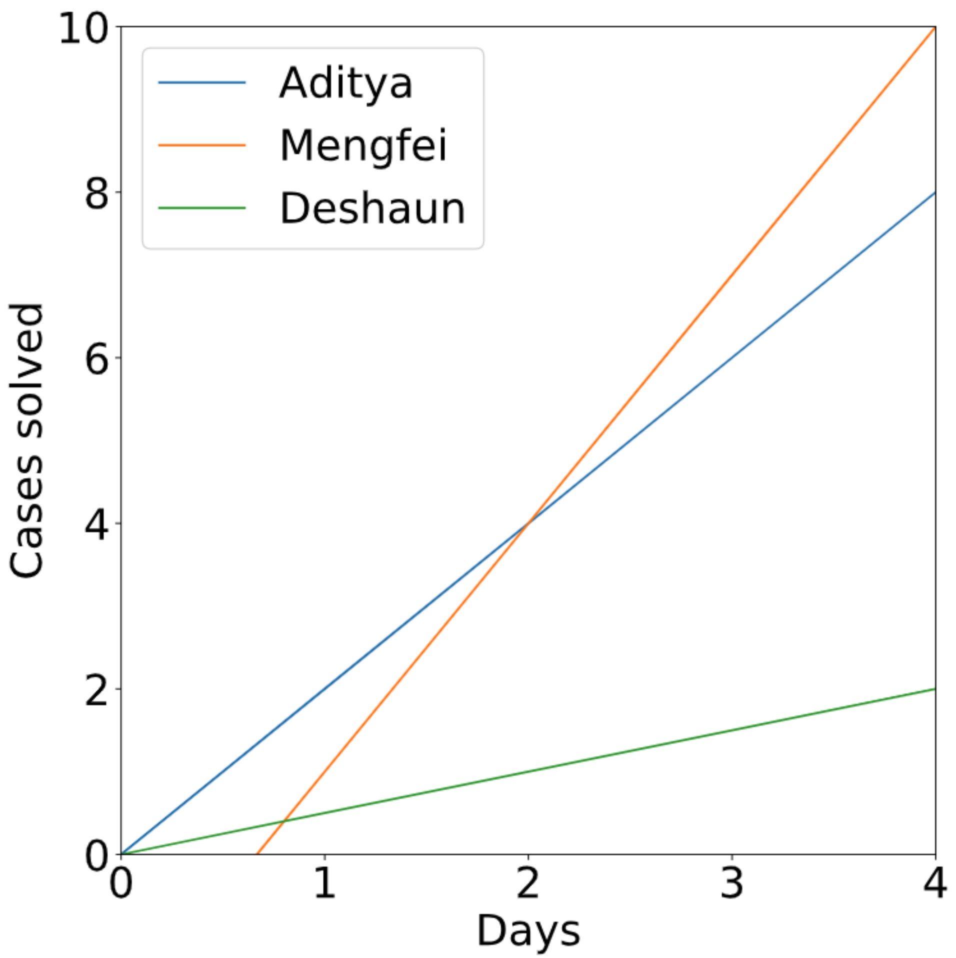
`plt.show()`



Legends

```
plt.plot(aditya.days,
         aditya.cases,
         label="Aditya")
plt.plot(deshaun.days,
         deshaun.cases,
         label="Deshaun")
plt.plot(mengfei.days,
         mengfei.cases,
         label="Mengfei")
```

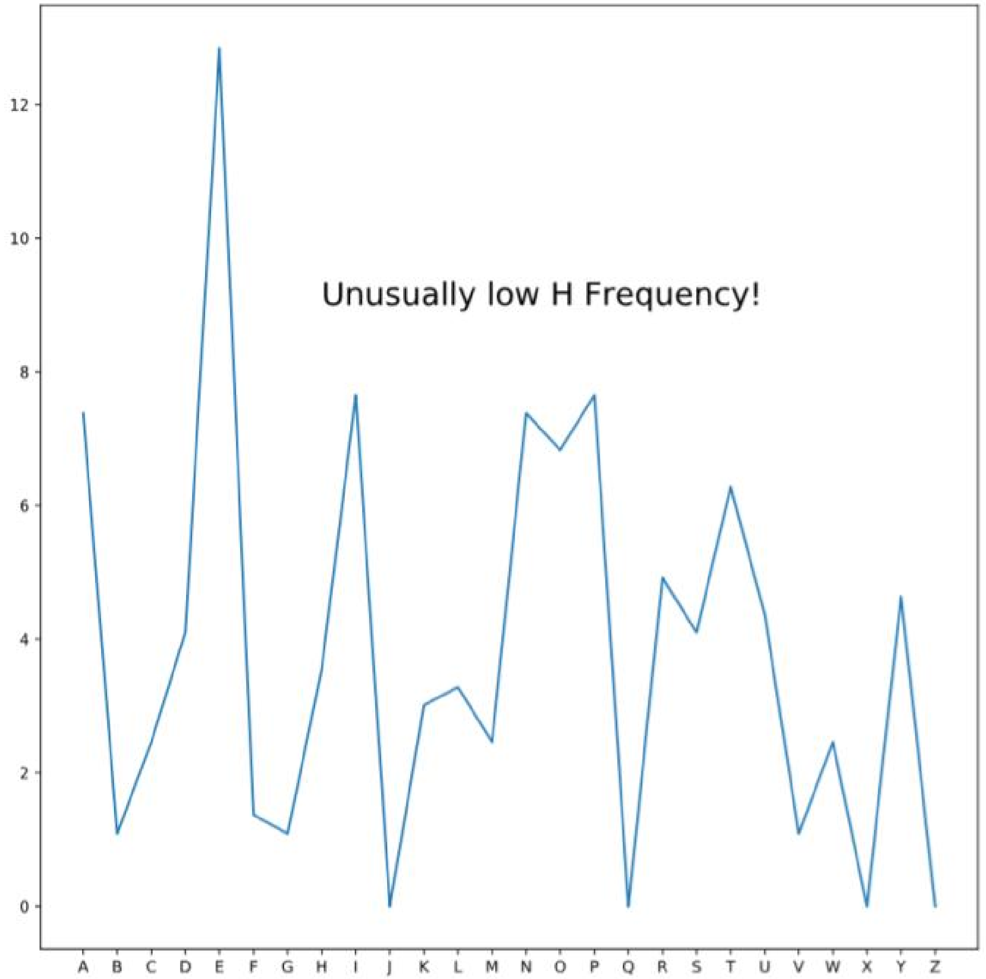
```
plt.legend()
```



Arbitrary text

```
plt.text(xcoord,
    ycoord,
    "Text Message")
```

```
plt.text(5,
    9,
    "Unusually low H frequency!"
```



Modifying text

- Change font size

```
plt.title("Plot title", fontsize=20)
```

- Change font color

```
plt.legend(color="green")
```

https://en.wikipedia.org/wiki/Web_colors

Let's practice

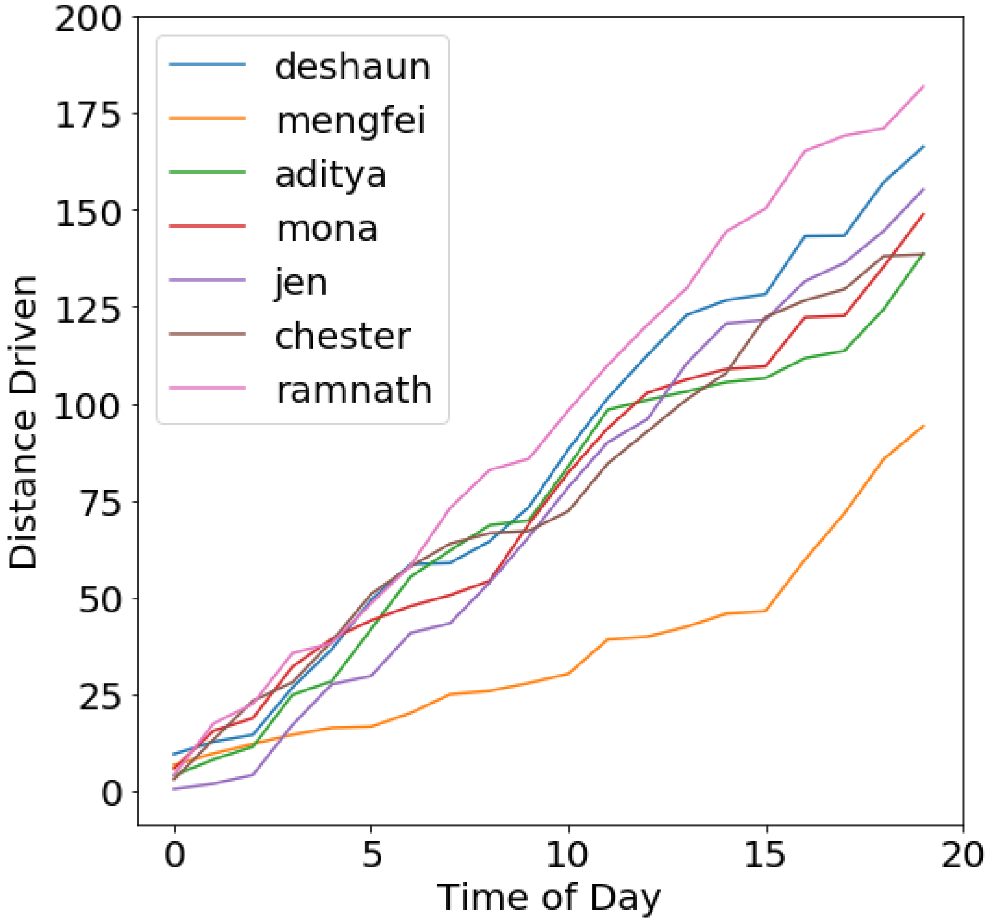
INTRODUCTION TO DATA SCIENCE IN PYTHON

Adding some style

INTRODUCTION TO DATA SCIENCE IN PYTHON

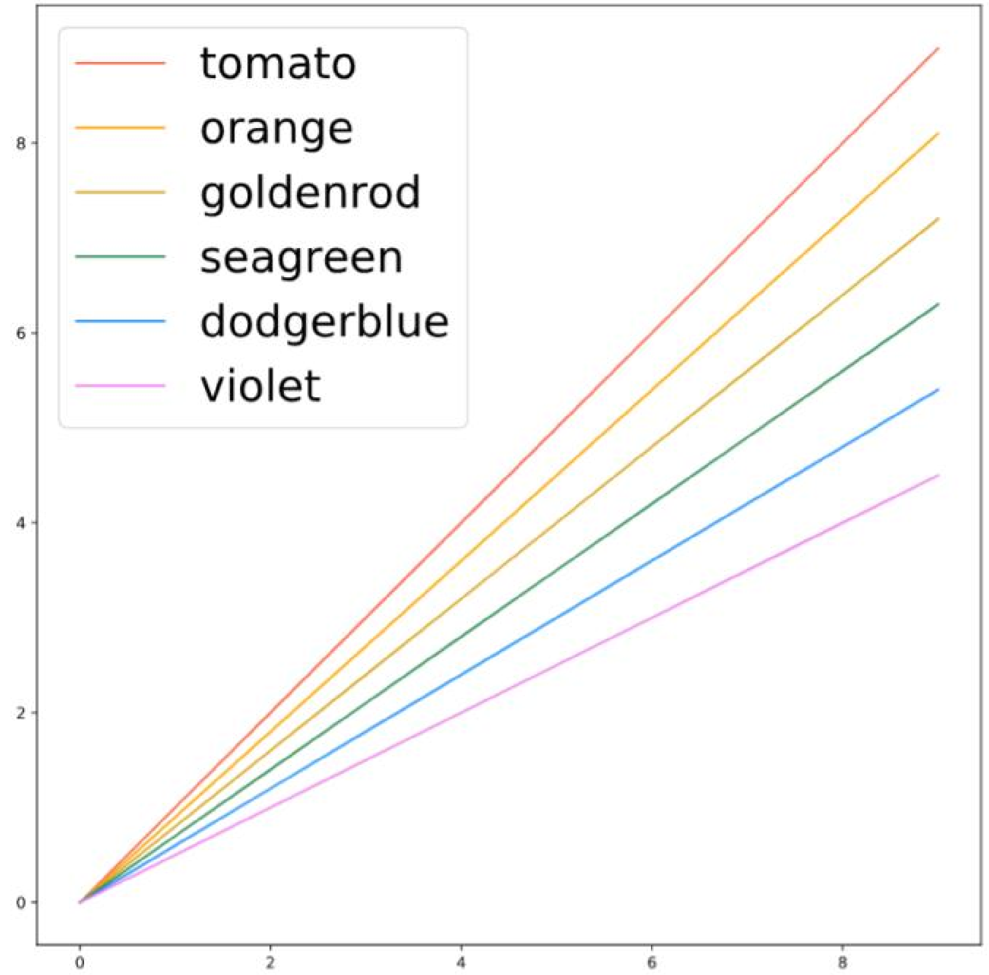
Hillary Green-Lerman
Senior Curriculum Lead, DataCamp

And miles to go



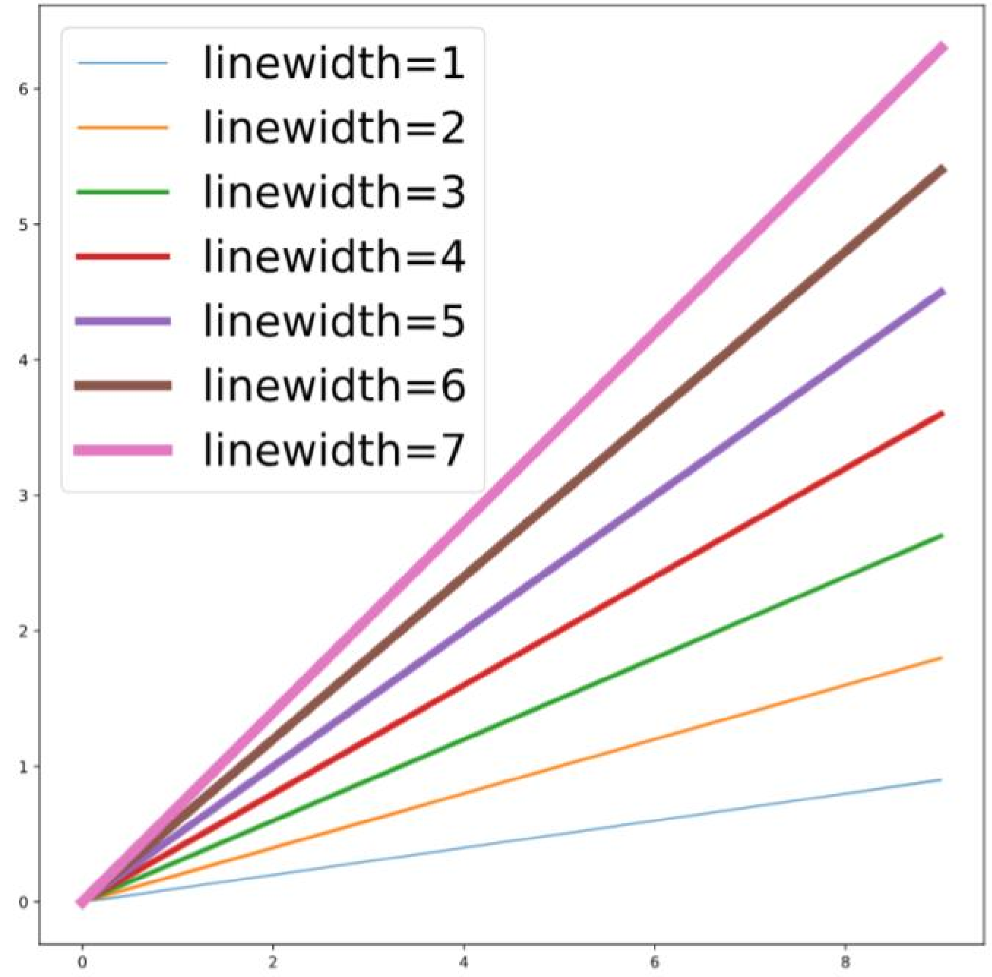
Changing line color

```
plt.plot(x, y1, color="tomato")
plt.plot(x, y2, color="organge")
plt.plot(x, y3, color="goldenrod")
plt.plot(x, y4, color="seagreen")
plt.plot(x, y5, color="dodgerblue")
plt.plot(x, y6, color="violet")
```



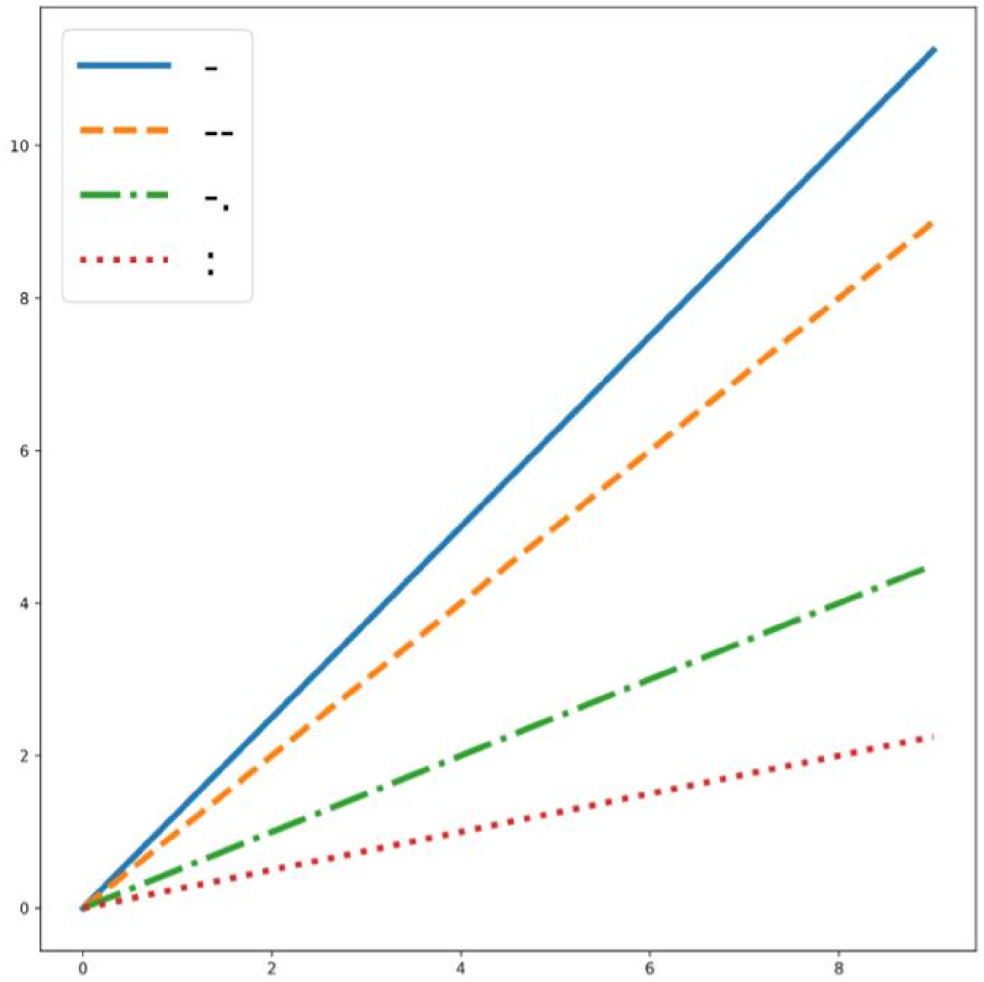
Changing line width

```
plt.plot(x, y1, linewidth=1)
plt.plot(x, y2, linewidth=2)
plt.plot(x, y3, linewidth=3)
plt.plot(x, y4, linewidth=4)
plt.plot(x, y5, linewidth=5)
plt.plot(x, y6, linewidth=6)
plt.plot(x, y7, linewidth=7)
```



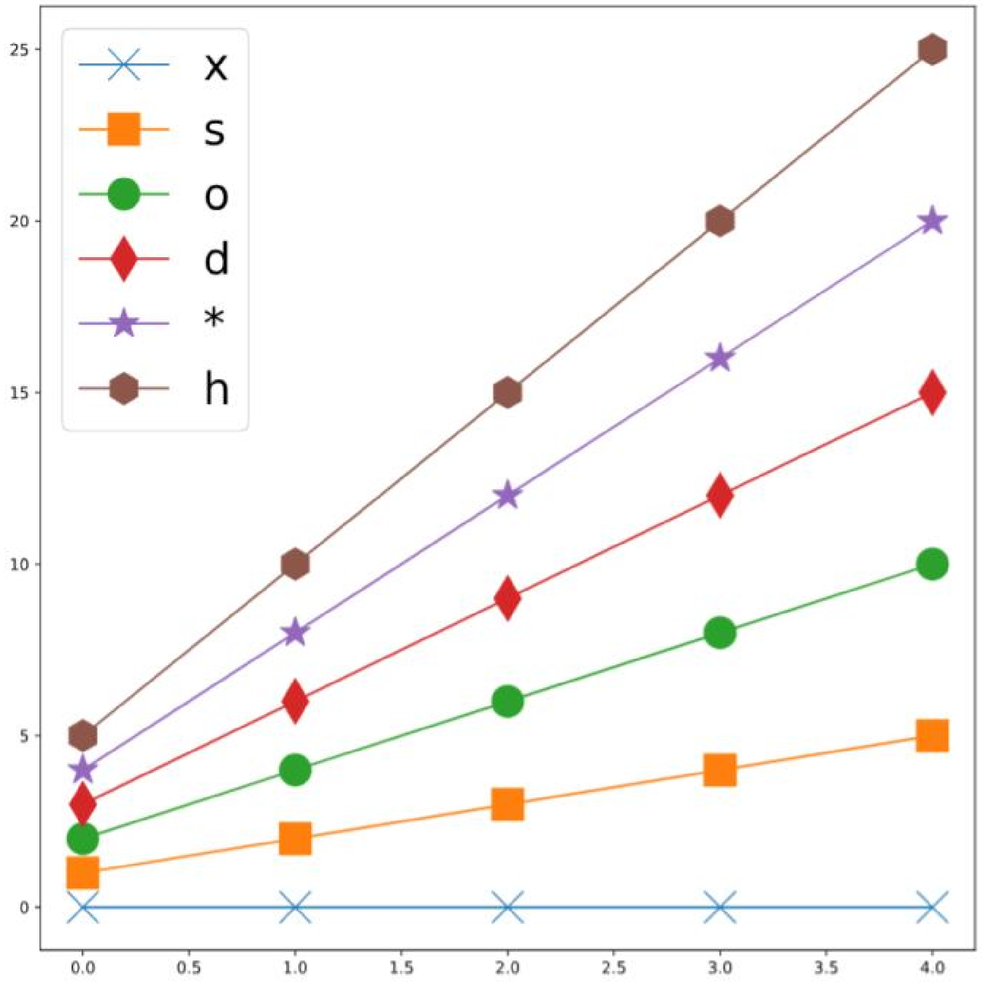
Changing line style

```
plt.plot(x, y1, linestyle='-')
plt.plot(x, y2, linestyle='--')
plt.plot(x, y3, linestyle='-.')
plt.plot(x, y4, linestyle=':')
```



Adding markers

```
plt.plot(x, y1, marker='x')
plt.plot(x, y2, marker='s')
plt.plot(x, y3, marker='o')
plt.plot(x, y4, marker='d')
plt.plot(x, y5, marker='*')
plt.plot(x, y6, marker='h')
```



Setting a style

Before any other plotting code:

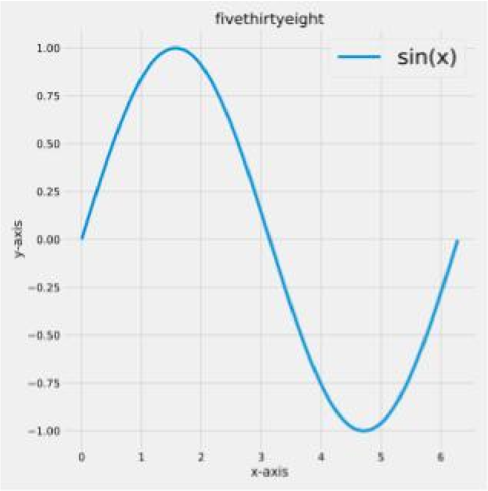
```
plt.style.use('fivethirtyeight')
```

```
plt.style.use('ggplot')
```

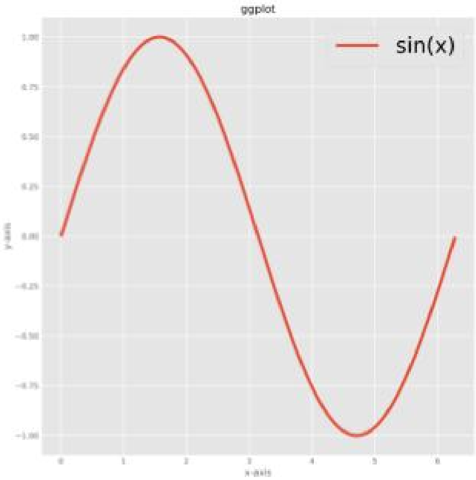
```
plt.style.use('seaborn')
```

```
plt.style.use('default')
```

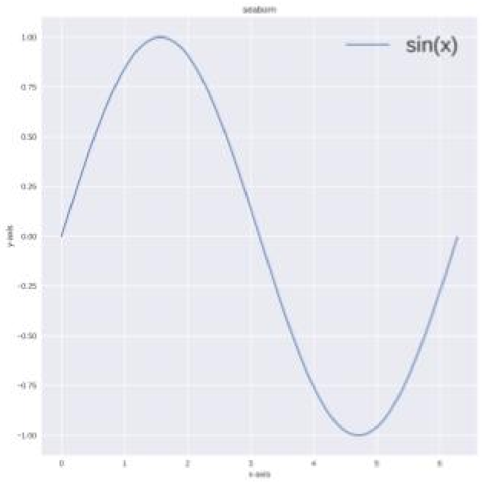
fivethirtyeight



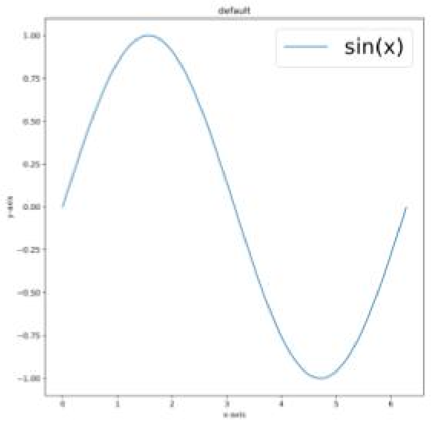
ggplot



seaborn



default



Let's Practice

INTRODUCTION TO DATA SCIENCE IN PYTHON