Learning Objectives

Students will be able to...

- **Create a matplotlib visualization in involving subplots**
- **Practice generating multiple figures**
- **Control the figure size and axis limits**

definition

Assumptions

- Learners are comfortable reading and importing CSV data sets, extracting relevant data into data frames, and printing that data to the console.
- Learners are comfortable using ggplot2 to create visualization charts.

Positioning our plots

Import our libraries

```
import matplotlib.pyplot as plt
```

We previously covered a bunch of different charts. This lesson , will emphasize more on stuff that were implied but not explicitly brought up in terms of presenting our graph.

If we where to create two scatterplots. Given the following data:

```
x=[27, 32, 38, 94, 70, 29, 17, 8, 48, 82, 52, 14, 91, 22, 58,
        96, 73, 99, 75, 76]
y=[78, 35, 28, 75, 2, 22, 25, 17, 72, 45, 86, 96, 24, 41, 73,
        51, 58, 76, 90, 77]
plt.scatter(x, y)
plt.scatter(y,x)
plt.show()
```

We know that when we run different plt commands. It will keep plotting over it. When needed to create a separate graph we use the `plt.figure`. `plt.figure` creates a new instance of a graph. For example, below we will create two different figures and label them and change the color of the second one.

```
import matplotlib.pyplot as plt
x=[27, 32, 38, 94, 70, 29, 17, 8, 48, 82, 52, 14, 91, 22, 58,
        96, 73, 99, 75, 76]
y=[78, 35, 28, 75, 2, 22, 25, 17, 72, 45, 86, 96, 24, 41, 73,
        51, 58, 76, 90, 77]
plt.scatter(x, y)
plt.title('Line Plot 1')
plt.figure()
plt.scatter(y,x,color="orange")
plt.title('Line Plot 2')
plt.show()
```

We can use the same technique to create a 3rd figure and so forth.

Subplots

Subplot is just figures with multiple plots in it. We are going to use the same data we used above to create a subplot.
Couple things to take note of when working with subplots is that it takes 3 arguments of the form :
`plt.subplot(# of rows, # of columns, index)`
The number of rows and columns will not change per subplot. The rows and column number are used to create a matrix, and the index is the only variable that changes and that one represent the current position in the matrix. We are going to try putting our figures in different matrices. Try the following in different cells:

```
plt.figure()
plt.subplot(1,3,1)
plt.scatter(x, y)
plt.title('Line Plot 1')
plt.subplot(1,3,2)
plt.scatter(y,x,color="red")
plt.title('Line Plot 2')
plt.subplot(1,3,3)
plt.plot(x,y)
plt.title('Line Plot 3')
plt.show()
```

and

```
plt.figure()
plt.subplot(2,2,1)
plt.scatter(x, y)
plt.title('Line Plot 1')
plt.subplot(2,2,2)
plt.scatter(y,x,color="red")
plt.title('Line Plot 2')
plt.subplot(2,2,3)
plt.plot(x,y)
plt.title('Line Plot 3')
plt.show()
```

Size and axes

figsize

We can use the same `figure()` we used to create figures, in order to also set the size. We just need to pass the width and height inside figure as as argument in that order. For Example, we can use the same data from the previous page

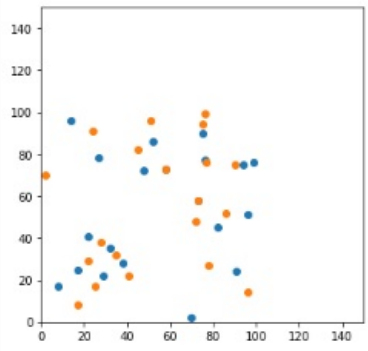
```
plt.figure(figsize=(5,5))
x=[27, 32, 38, 94, 70, 29, 17, 8, 48, 82, 52, 14, 91, 22, 58,
96, 73, 99, 75, 76]
y=[78, 35, 28, 75, 2, 22, 25, 17, 72, 45, 86, 96, 24, 41, 73,
51, 58, 76, 90, 77]
plt.scatter(x, y)
plt.scatter(y,x)
plt.show()
```

Let is try switching the variable number for figsize to for example `(10,5)` and let see how much our figure has changed.

Changing axes size

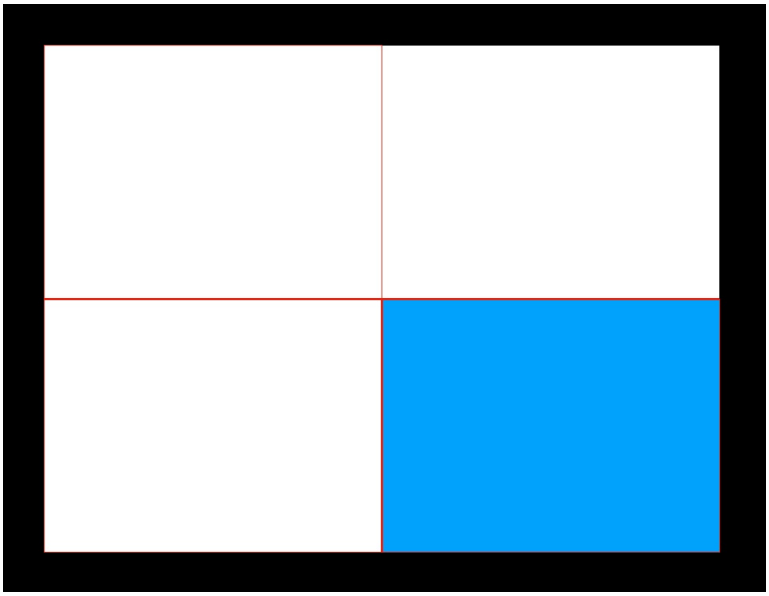
Now that we know how to change the size of our figures. We can also tackle how to modify the x and y limits of our figure.
For that we need to use two new function `xlim()` and `ylim()`. Lets give it a try!

```
plt.xlim(0,150)
plt.ylim(0,150)
```



images/exa1

Formative Assessment



images/assses