

wbs

WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

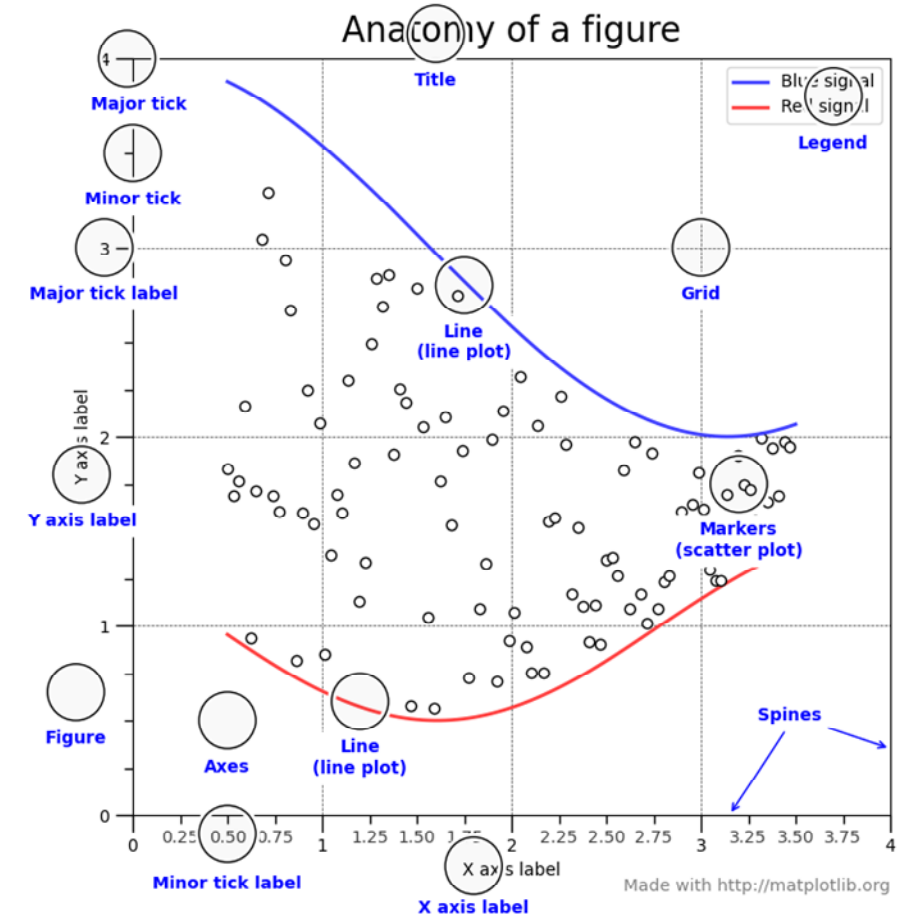
**For the
Change
Makers**

Advanced Programming for Data Science

**Week 4: Data Visualization
Information Systems and Management
Warwick Business School**

Customize Your Figure

- Axis
- Legend
- Annotation
- Text
- Size and position
- Please refer to the notebook for details.



Customize Your Figure: Size

- To set figure size when creating a new figure, you can use the argument `figsize`, the value should be a tuple setting width and height in inches.

```
plt.figure(figsize=(8,8))
```

- To set figure size of an existing figure, you need to retrieve the figure object first by calling `gcf()`, then use figure method `set_size_inches()`.

```
plt.figure()
```

```
plt.plot(x,y)
```

```
plt.gcf().set_size_inches(8, 8)
```

Customize Your Figure: Title

`pyplot.title()`

- Label set the text of the title
- Loc set the position, default is 'center'

```
plt.title('First Plot')
```

```
plt.title('First Plot', loc='left')
```

Customize Your Figure: Axis

pyplot approach

To set axis limit

```
plt.xlim() #left, right  
plt.ylim()
```

To set axis label

```
plt.xlabel() #message  
plt.ylabel()
```

To set axis ticks

```
plt.xticks() #ticks, Labels  
plt.yticks()
```

oop approach

To set axis limit

```
fig, ax = plt.subplots()  
ax.set_lim()
```

Customize Your Figure: Add Legend

- specify a name when plotting the line.
- `pyplot.legend()` will then use the name to add a legend

```
plt.figure()
```

```
plt.plot(x, y1, label='y=2x')
```

```
plt.plot(x, y2, label='y=3x')
```

```
plt.legend()
```

- By default, it will find the best position to place the legend. You may explicitly specify the location by passing the argument for `loc`.

```
plt.legend(loc='lower left')
```

Customize Your Figure: Add Text

`plt.text()`

- x, y will set the position of first character of the text
- s is the text string

```
plt.text(20, 150, 'Some explanation')
```

- There are many further customization you can add, please refer to the documentation for details.

```
plt.text(20, 150, 'Some explanation', color='red',  
        fontsize = 16)
```

Customize Your Figure: Add Annotation

`plt.annotate()`

- `s` sets the annotation message
- `xy` sets the coordinate (in a tuple) of annotation

```
plt.annotate('Point A', xy=(10,10))
```

There are many further customization you can add, please refer to the documentation for details.

```
plt.annotate('Point A', xy=(10,10), xytext=(+6, -5),  
textcoords='offset points')
```


Exercise 1-1

- Use Matplotlib to create a simple plot for x and y1 (shown as below) using

1) pyplot command function approach

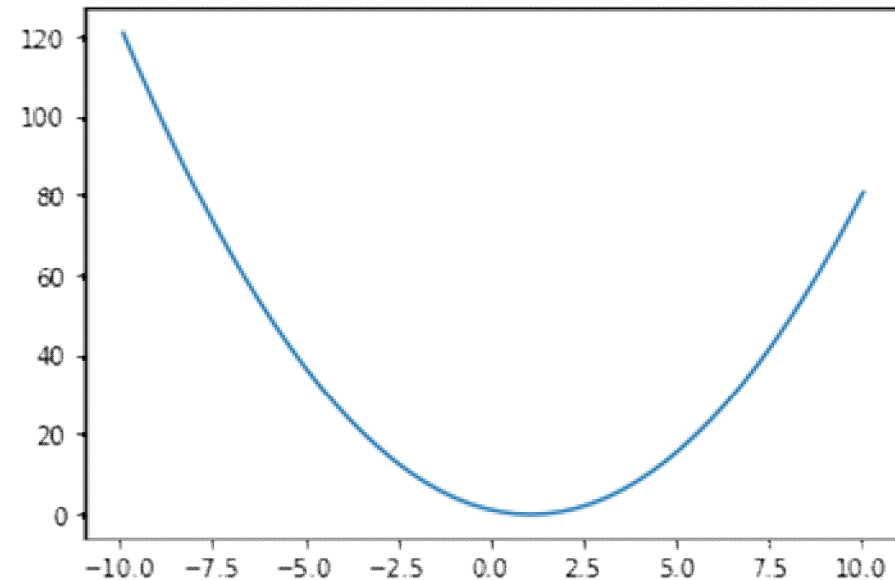
2) object-oriented approach

X and y can be generated by

import numpy as np

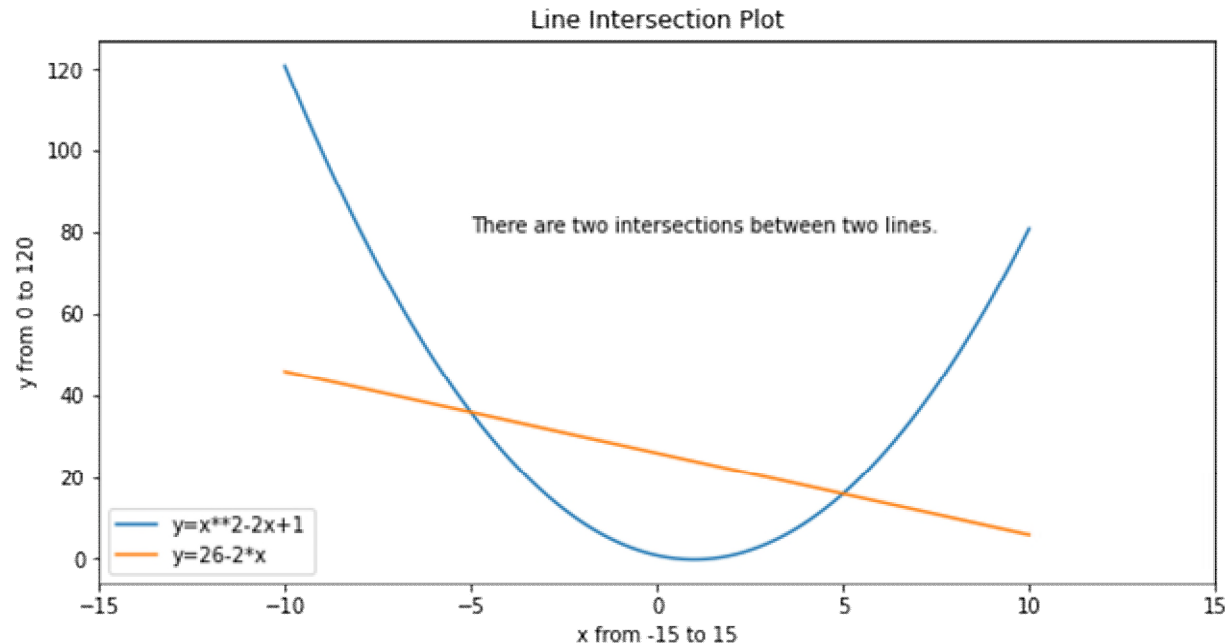
```
x = np.linspace(-10,10,50)
```

```
y1 = x**2 - 2*x + 1
```



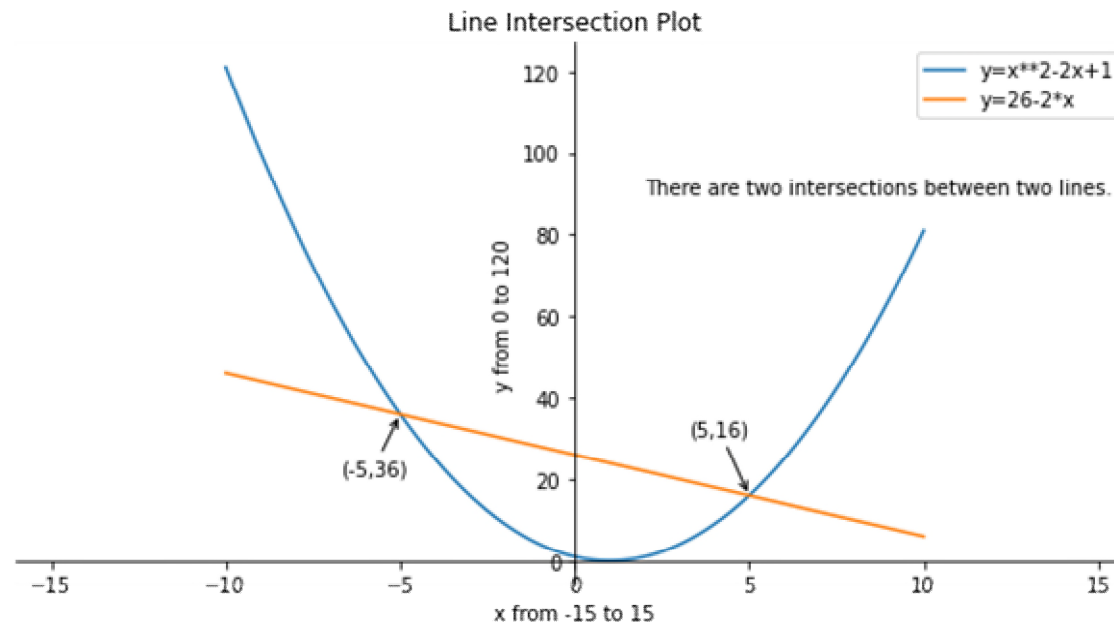
Exercise 1-2

- Add a second line to the plot, $y_2 = 26 - 2 * x$
- Make further edits to your previous plot so it will look like the example below



Exercise 1-3

- Optional, make further edits so your plot looks like the example below



Exercise 2

- Create a plot using seaborn with built-in dataset 'mpg' to illustrate the change of mpg over years.
- Make necessary edits so the plot looks like the example on the right.

