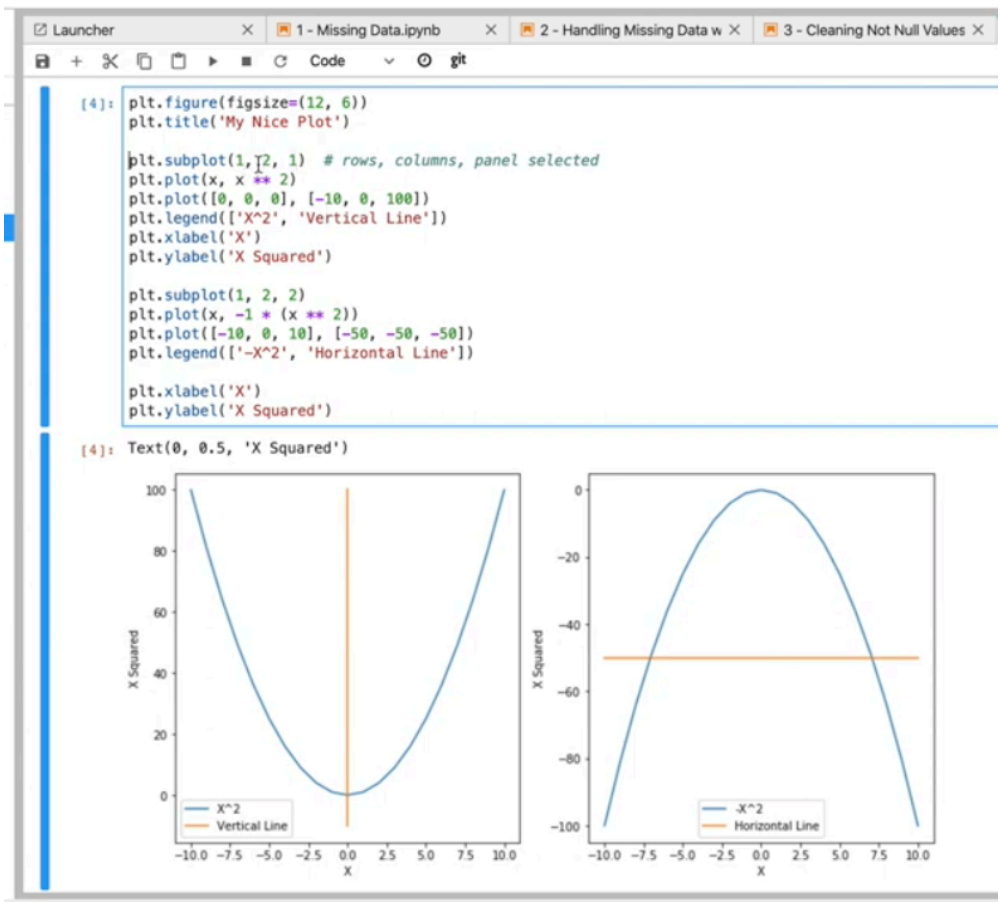
- -> notebooks from this lecture: https://github.com/ine-rmotr-curriculum/data-cleaning-rmotr-freecodecamp
- -> looking at the data from a visualisation perspective
- -> when a value is an outlier, it might be invalid and need cleaning
- -> the matplot lib library
- -> this can be accessed from pandas <- this relies in matplotlib
- ***-> this has two APIs***
  - -> the first is global
  - -> the second is the object oriented API
  - -> these are two different way of doing the same thing
  - -> the global API is older
  - -> he prefers the object oriented API
  - -> most of tehe solutions on Stack overflow are global
  - -> you might needs to translate one ot the other
  - -> we have imported the whole Python module
- ***-> example***
  - -> we are invoking plt.figure
  - -> then a title
  - -> then plotting to different graphs
  - -> the functions we are using are at module level
  - -> we are calling a function, which is modifying the final result of the plot
  - -> there is no object oriented way which says one of the figures
  - -> creating a figure and drawing it
  - -> we have one row and two columns
  - -> we have activated the plot and then are drawing it
  - -> having a legend and setting labels
  - -> then switching the pot
  - -> having a second plot
  - -> every line after this affects the second plot



- ***-> the OOP approach***
  - -> we are creating a figure and access
  - -> one figure is in red and the other is on the right
  - -> we are creating the two figures using an object oriented approach, and we are keeping references to them
  - -> we can have multiple axes
  - -> then we tell it what to plot on which axis
  - -> in this example there are four different axes
  - -> we use the plot method for this -> n rows and n columns
  - -> axis number 1 and 2

- -> we can change the order, but the results can be the same

- ***-> matplotlib has a plotting function***
  - -> passing all the values in x and y
  - -> then plotting everything in x and in y
  - -> using the straight line in green
  - -> linestyle marker and specific keyword arguments

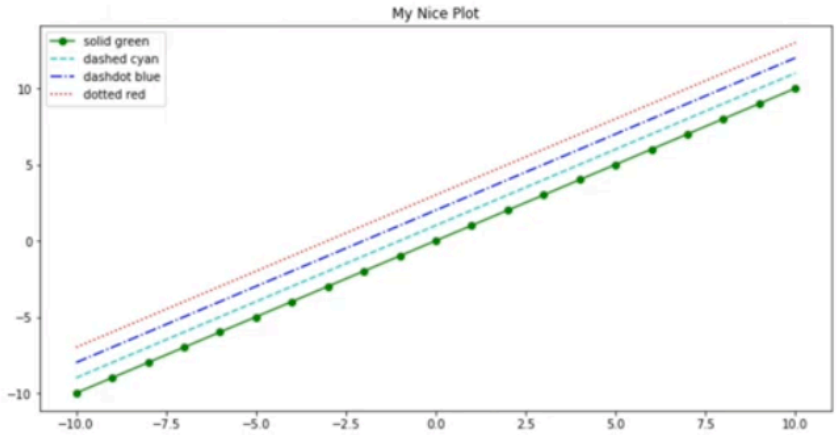
```
[8]: fig, axes = plt.subplots(figsize=(12, 6))

axes.plot(x, x + 0, '-og', label="solid green")
axes.plot(x, x + 1, '--c', label="dashed cyan")
axes.plot(x, x + 2, '-.b', label="dashdot blue")
axes.plot(x, x + 3, ':r', label="dotted red")

axes.set_title("My Nice Plot")

axes.legend()
```

```
[8]: <matplotlib.legend.Legend at 0x7fa54bc5f130>
```



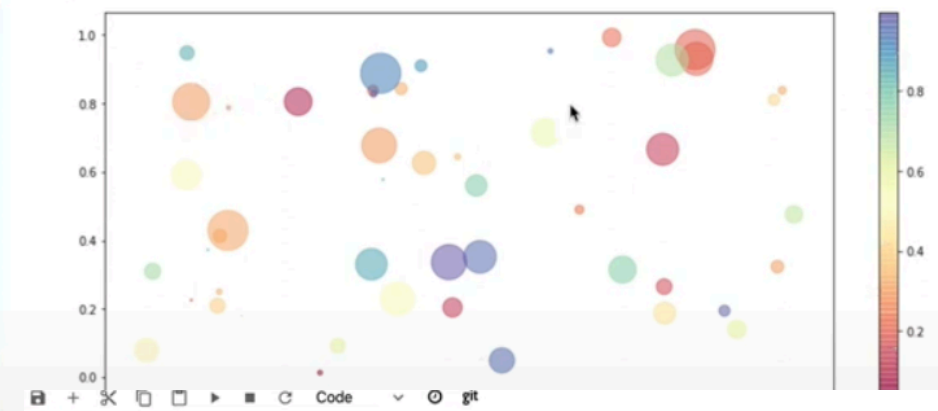
- ***scatterplots example***
  - -> we have different values, and colour maps
  - -> we can plot three to four different dimensions of the data
  - -> four dimensions in one figure

```
area = np.pi * (20 * np.random.rand(N))**2  # 0 to 15 point radii
```

```
[15]: plt.figure(figsize=(14, 6))

plt.scatter(x, y, s=area, c=colors, alpha=0.5, cmap='Spectral')
plt.colorbar()

plt.show()
```



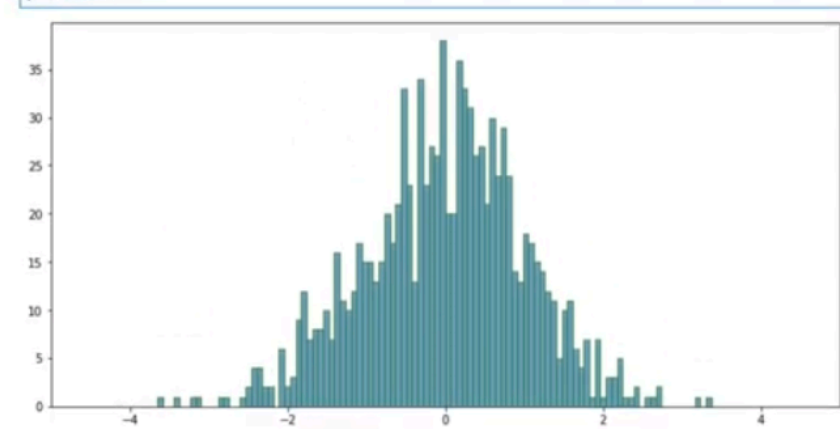
- ***historgram example***
  - -> this takes the value we are plotting and the amount of bins we want
  - -> we can also create kernel density estimator diagrams

```
[17]: values = np.random.randn(1000)
```

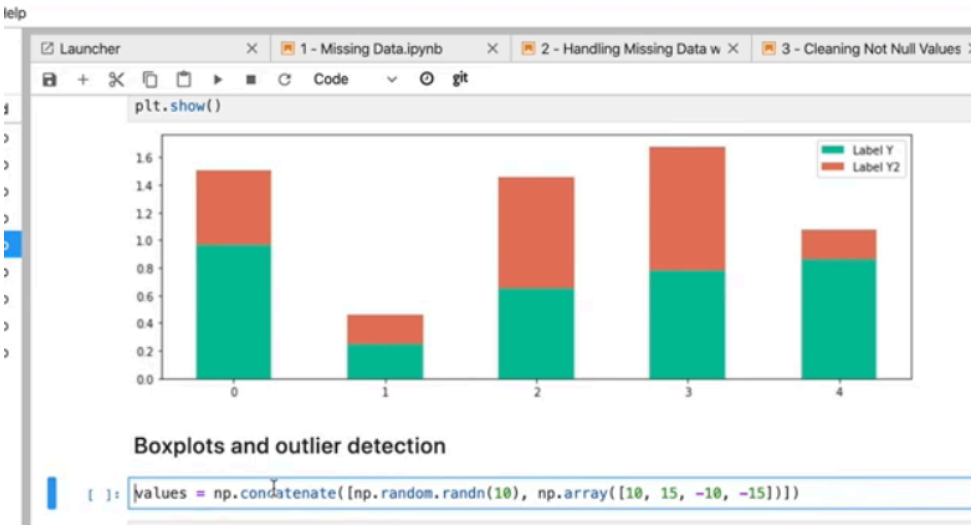
```
[18]: plt.subplots(figsize=(12, 6))

plt.hist(values, bins=100, alpha=0.8,
         histtype='bar', color='steelblue',
         edgecolor='green')
plt.xlim(xmin=-5, xmax=5)

plt.show()
```



- ***bar plots***
  - -> to stack the data
  - -> you can also use box plots to show outliers
    - -> 1.5 times above or below the UQ or LQ
    - -> these outliers can be treated as invalid values
    - -> invalid being, does not make sense for the context (for example, an age of 170)
    - -> 170 is an integer, but does not make sense for the age of a person



- ***-> question***

When using Matplotlib's global API, what does the order of numbers mean here?
plt.subplot(1, 2, 1)

- options
  - My figure will have one column, two rows, and I am going to start drawing in the first (left) plot.
  - I am going to start drawing in the first (left) plot, my figure will have two rows, and my figure will have one column.
  - My figure will have one row, two columns, and I am going to start drawing in the first (left) plot. <- This one