Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python

It is used along with NumPy to provide an environment that is an effective open-source alternative to MatLab

Plt **.** title ( ' My Graph Title ' )

Plt **.** xlabel ( ' My x-axis label ' )

Plt **.** ylabel( ' My y-axis label ' )

plt**.**show()

Plt . legend() -> 

to plot sin(x) and cos(x) as separate plots, but side by side? This is where ***subplots(nrows, ncols)*** comes in handy

x = np.linspace(1, 21)

fig, axes = plt.subplots(nrows=4, ncols=4, figsize=(14, 8))

axes[0][0].plot(x, x\*\*2)

axes[0][1].plot(x, x)

plt.show()

We can configure the ranges of the axes using the ***set\_ylim*** and ***set\_xlim***, or ***axis('tight')*** for automatically getting "tightly fitted" axes ranges:

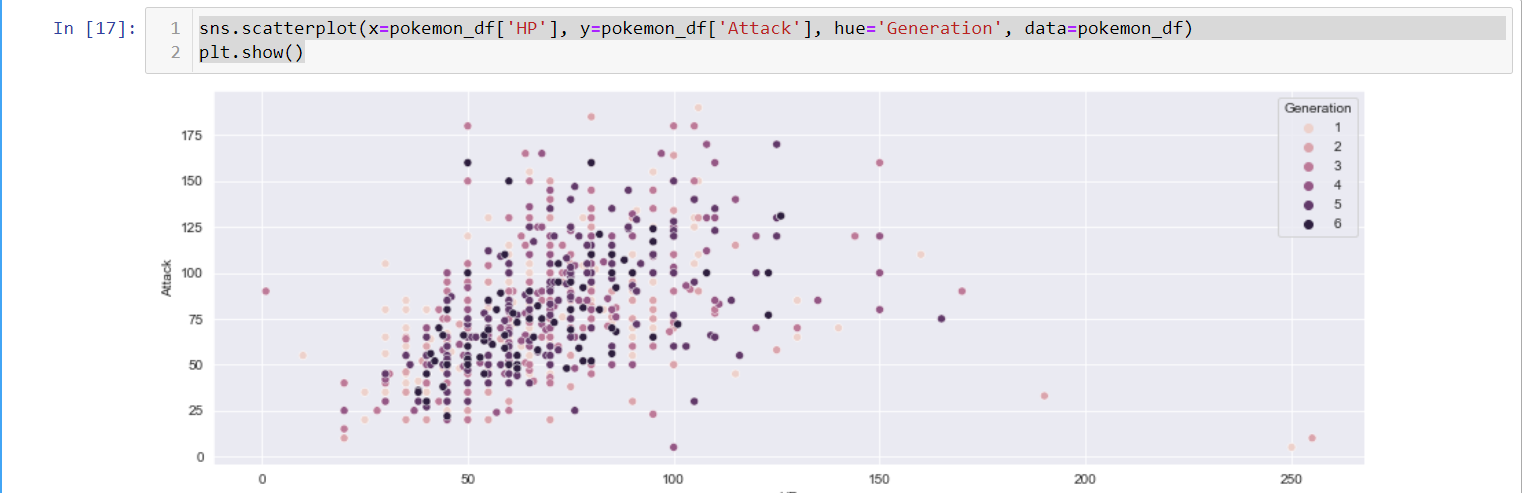
Axes [ 0 ] **.** plot (x , x **\*\***2 , x, x**\*\***4)

Axes [ 0 ]**.** Set\_title ( ' default axis range ' )

how we can create a scatter plot of pokemon HP and Attack

plt.scatter(x = pokemon\_df [ ' HP ' ], y = pokemon\_ df [ ' Attack ' ] )

plt.show()



Plt . hist ( x = pokemon \_ df [ ' Attack ' ] )

type\_1\_values = pokemon\_df['Type 1'].value\_counts().value

type\_1\_names = pokemon\_df['Type 1'].value\_counts().index

***Note: Make sure you don't have categorical variables in your DataFrame or you'll get an error.***

Pokemon \_ df [ [ ' Speed ' ] ] . corr ( )

Sns **.**heatmap ( pokemon \_ df [ [ ' HP ' , ' Attack ' , ' Defense ' , ' Sp . Atk' , ' Sp. Def ', ' Speed ' ] ] **.** corr( ) ) -> It will show us visualization form of correlation data, so it starts from 0 to 1.

Plt **.** show( )

**Seaborn for heatmaps**

import seaborn as sns