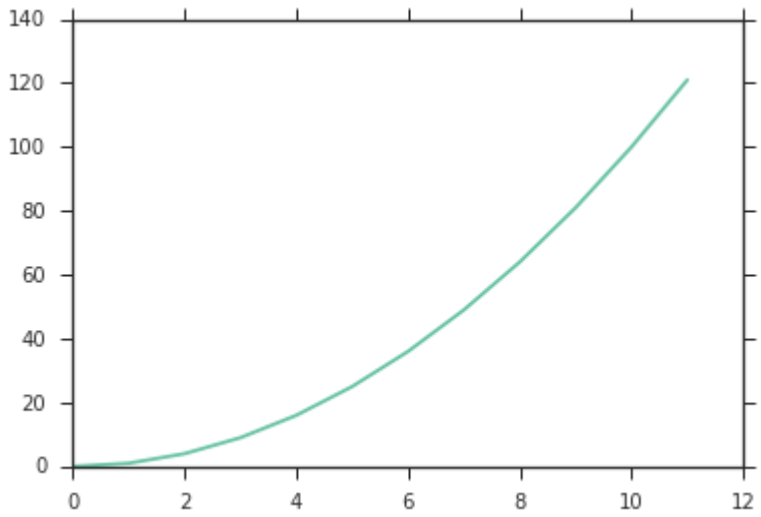
Visualization with Line Graphs

This lesson explains what a line graph is, why it is used, and how to visualize data with a line graph using Python libraries.

WE'LL COVER THE FOLLOWING

Introduction to line graphs

Line graphs are very useful for showing a value over time, such as a stock's price. Typically, one would use a line graph over a scatter plot if there is a connecting component between the values, such as time. Here is an example of a value growing exponentially over time:



Line graphs in Python

The `lineplot()` function of **Seaborn** is what plots the line graph. The *first parameter* is the list or array of x-values and the *second parameter* is the array of y-values.

That's it! Line graphs are simple to implement and can be very effective.

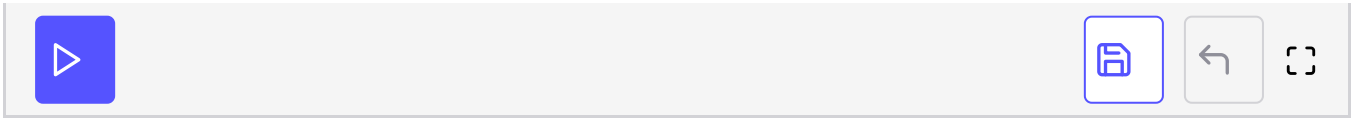
Let's take a look at an example using the flight dataset from Seaborn. This dataset has three columns. **Year**, **Month**, and **Passengers**. The `passengers` column represents the number of flight passengers for that year and month.

```
import seaborn as sns            # importing seaborn functionality
import pandas as pd
import matplotlib.pyplot as plt

flights_long=sns.load_dataset("flights")   # importing dataset

# filtering the dataset to obtain the January records for all years
flights_long=flights_long[flights_long.month == 'January']

#plotting a line graph
plot=sns.lineplot(flights_long.year, flights_long.passengers)
```



The above code loads the dataset, limits the data to only the month of January and then plots the number of passengers for every year as a line plot.

That's how the variations in data along time can be mapped with a line graph. Next, we'll look at heat maps.