



# Climate Data Visualization with Python

By

**Tanu Sharma**

([tanu35bhp@gmail.com](mailto:tanu35bhp@gmail.com))

MoES JRF (MRFP)

Climate Monitoring and Prediction Group, IMD Pune

Training on “Climate Data Analysis and Visualization With Python”

11-15 September 2023

# Various Visualization Libraries



matplotlib



Seaborn



# Visualization using **matplotlib**

- Matplotlib is an amazing visualization library in Python mainly for 2-D plots of arrays.
- It is built on Numpy arrays and was introduced by **John Hunter** in the year 2002.
- Matplotlib comes with a wide variety of plots. We can create high quality plots like line, scatter, bar etc.
- Matplotlib allow us easily create multi-plots on the same figure (panel plot) using `.subplot()` function.

# How to plot using Matplotlib

## Matplotlib for beginners

Matplotlib is a library for making 2D plots in Python. It is designed with the philosophy that you should be able to create simple plots with just a few commands:

### 1 Initialize

```
import numpy as np
import matplotlib.pyplot as plt
```

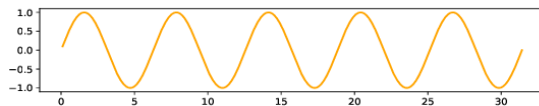
### 2 Prepare

```
X = np.linspace(0, 4*np.pi, 1000)
Y = np.sin(X)
```

### 3 Render

```
fig, ax = plt.subplots()
ax.plot(X, Y)
plt.show()
```

### 4 Observe



## Choose

Matplotlib offers several kind of plots (see Gallery):

```
X = np.random.uniform(0, 1, 100)
Y = np.random.uniform(0, 1, 100)
ax.scatter(X, Y)
```



```
X = np.arange(10)
Y = np.random.uniform(1, 10, 10)
ax.bar(X, Y)
```



```
Z = np.random.uniform(0, 1, (8,8))
ax.imshow(Z)
```



```
Z = np.random.uniform(0, 1, (8,8))
ax.contourf(Z)
```



```
Z = np.random.uniform(0, 1, 4)
ax.pie(Z)
```



```
Z = np.random.normal(0, 1, 100)
ax.hist(Z)
```



```
X = np.arange(5)
Y = np.random.uniform(0, 1, 5)
ax.errorbar(X, Y, Y/4)
```



```
Z = np.random.normal(0, 1, (100,3))
ax.boxplot(Z)
```



## Tweak

You can modify pretty much anything in a plot, including limits, colors, markers, line width and styles, ticks and ticks labels, titles, etc.

```
X = np.linspace(0, 10, 100)
Y = np.sin(X)
ax.plot(X, Y, color="black")
```



```
X = np.linspace(0, 10, 100)
Y = np.sin(X)
ax.plot(X, Y, linestyle="--")
```



```
X = np.linspace(0, 10, 100)
Y = np.sin(X)
ax.plot(X, Y, linewidth=5)
```



```
X = np.linspace(0, 10, 100)
Y = np.sin(X)
ax.plot(X, Y, marker="o")
```



## Organize

You can plot several data on the the same figure, but you can also split a figure in several subplots (named Axes):

```
X = np.linspace(0, 10, 100)
Y1, Y2 = np.sin(X), np.cos(X)
ax.plot(X, Y1, X, Y2)
```



```
fig, (ax1, ax2) = plt.subplots(2,1)
ax1.plot(X, Y1, color="C1")
ax2.plot(X, Y2, color="C0")
```



```
fig, (ax1, ax2) = plt.subplots(1,2)
ax1.plot(Y1, X, color="C1")
ax2.plot(Y2, X, color="C0")
```



## Label (everything)

```
ax.plot(X, Y)
fig.suptitle(None)
ax.set_title("A Sine wave")
```



```
ax.plot(X, Y)
ax.set_ylabel(None)
ax.set_xlabel("Time")
```



## Explore

Figures are shown with a graphical user interface that allows to zoom and pan the figure, to navigate between the different views and to show the value under the mouse.

## Save (bitmap or vector format)

```
fig.savefig("my-first-figure.png", dpi=300)
fig.savefig("my-first-figure.pdf")
```

Matplotlib 3.5.0 handout for beginners. Copyright (c) 2021 Matplotlib Development Team. Released under a CC-BY 4.0 International License. Supported by NumFOCUS.

# Spatial Plots using Cartopy

- Cartopy is designed for geospatial data processing and producing maps.
- Cartopy uses 'Numpy' and 'Shaply' library. It is used for advances mapping in 'matplotlib'.
- It does point, line, vector transformation from cartesian coordinates to spherical coordinates based on various projections.
- 'Shapefile' can be integrated using cartopy.



Thank You