Avveer's Intro To Graphing Tutorial (matplotlib)

By: AvveerSchoolAccount

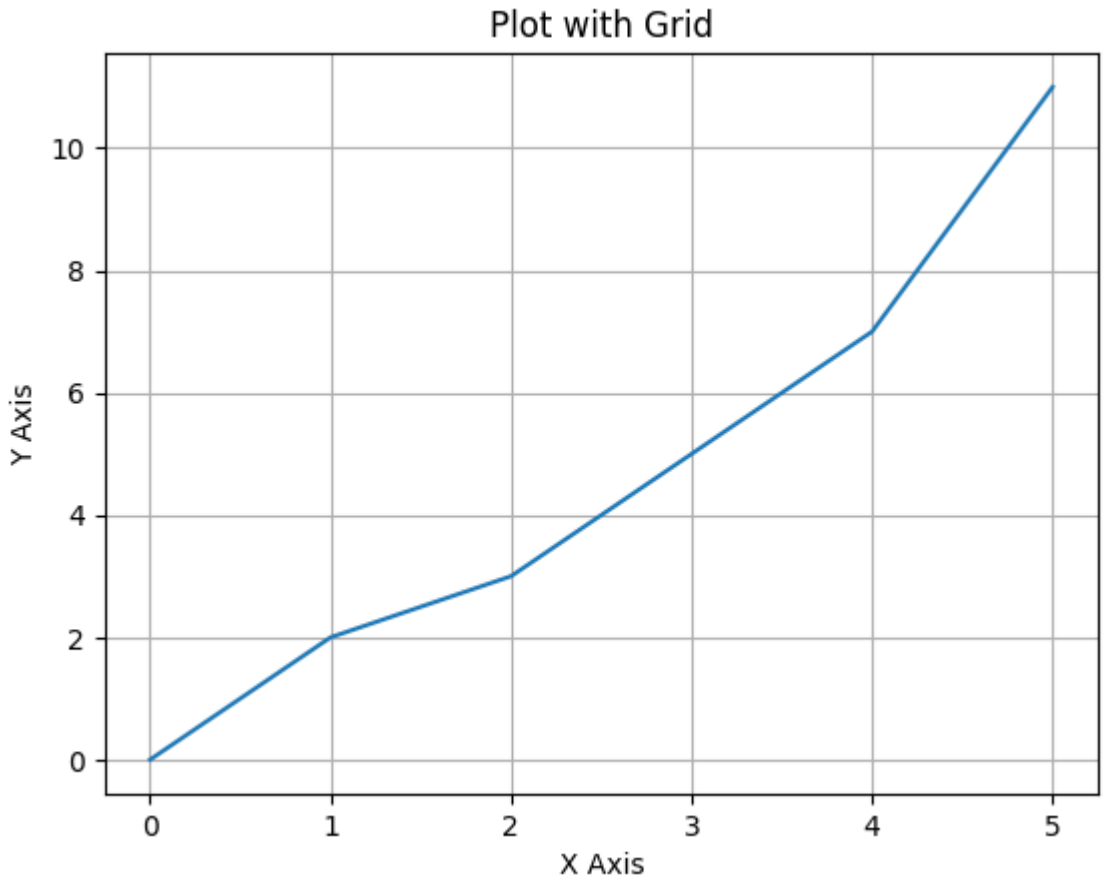
Github Repo: Matplotlib_SecondLesson

Second Tutorial Contains:

1. Adding Grid
2. Histogram
3. Subplots Basics
4. Basic Info (No Seperate Cell: Each One Gradually Adds Concepts)

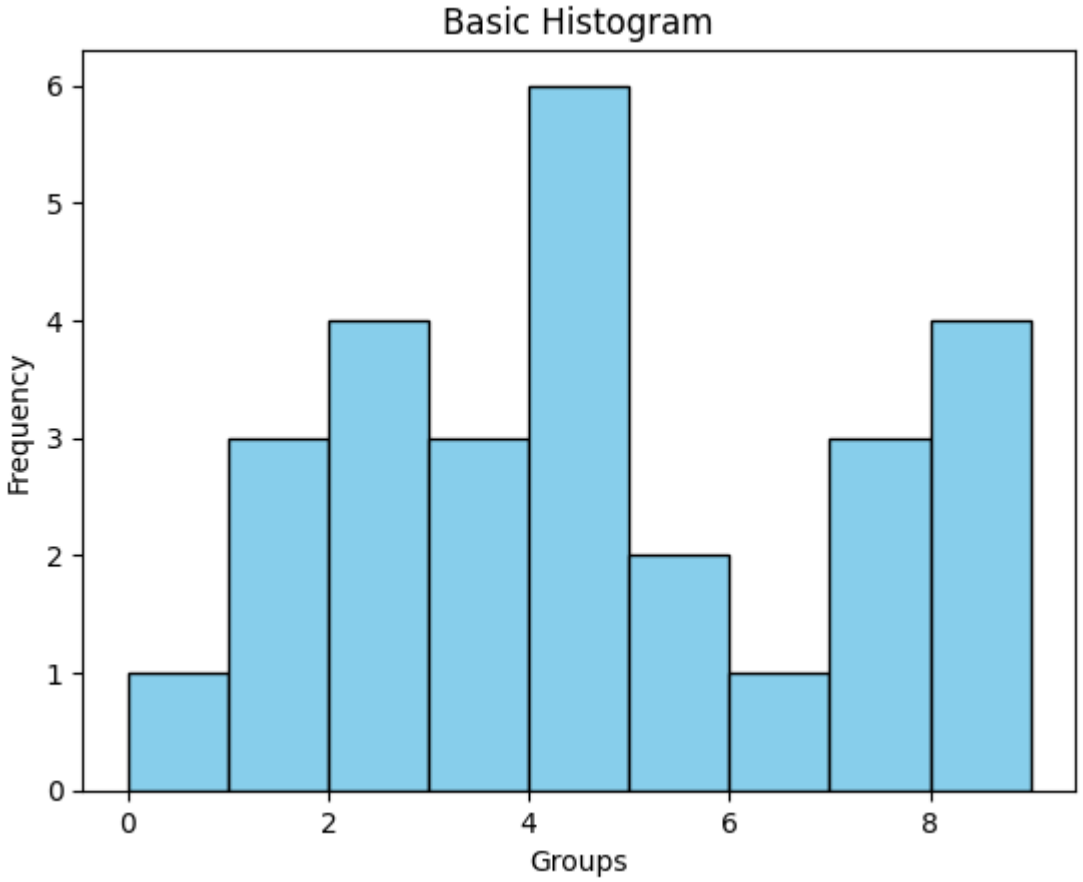
1. Adding Grids:

In [26]:
```
# Import matplotlib
import matplotlib.pyplot as plt

# Create x & y variables:
x = [0, 1, 2, 3, 4, 5]
y = [0, 2, 3, 5, 7, 11]

# Create plot of choice
plt.plot(x, y)

# syntax to add a grid behind the plot
plt.grid(True)

# Name labels and title
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.title("Plot with Grid")

# Display plot
plt.show()
```



2. Histograms:

In [19]:
```
# Create variables:
x1 = [0, 1, 2, 3, 4, 5, 3, 4, 5, 1, 3, 2, 6, 4, 8, 9, 2, 7, 4, 9, 1, 4, 2

# Create plot with 9 bins/categories, skyblue color, and black edges
plt.hist(x1, bins=9, color="skyblue", edgecolor="black")

# Name labels and title
plt.xlabel("Groups")
plt.ylabel("Frequency")
plt.title("Basic Histogram")

# Display plot
plt.show()
```



3. Subplot Basics

Explanation

Before understanding how **subplots** work, it is important to understand what **figures** and **axes** are.

Figure

- A **figure** is the overall container for the plots.
- It represents the entire drawing canvas or window.
- You can increase or decrease the figure size based on your needs.

Axes

- **Axes** are the regions inside the figure where the plots are actually drawn.
- They contain the x-axis, y-axis, labels, titles, and the plotted data.
- A single figure can contain **multiple axes**.

Analogy

- Think of a **figure** as a **old fashioned window** (like the traditional ones on gigerbread houses with 4 smaller glass quares in them).
- Think of **axes** as each **glass square**.

- You can draw multiple graphs on one window by **adding more quare glass slots** or allocating space for each graph (add more axes).

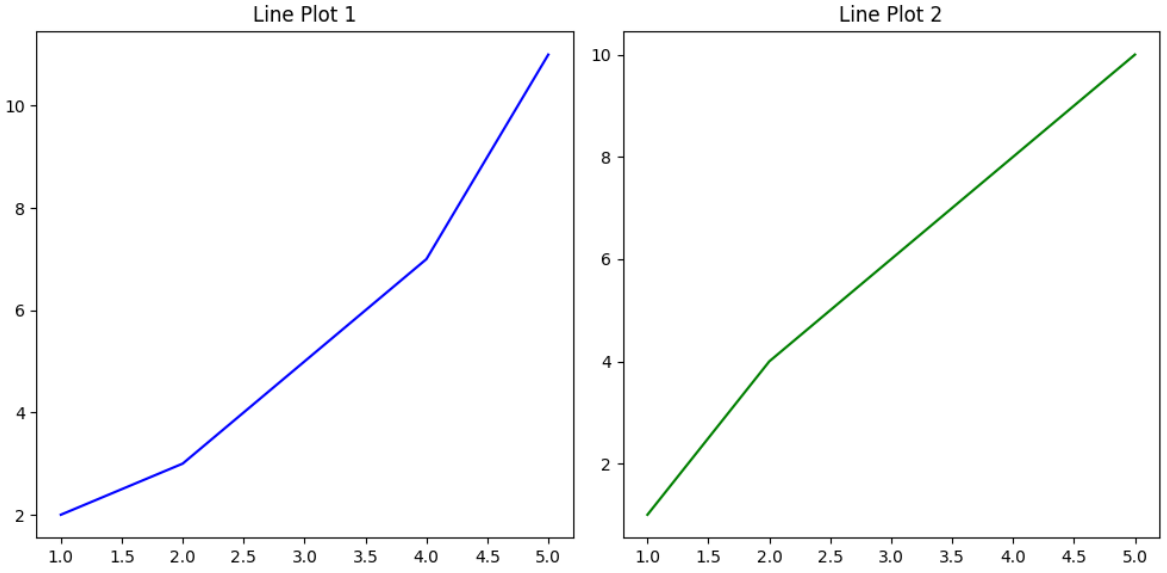
What Are Subplots?

- **Subplots** are used to **add glass squres** to **one window**
- They are used to **create a grid of axes within a single figure**. They allow you to neatly organize and display multiple graphs together, making it easier to compare and analyze related visualizations.

---

a) Single Row Subplot (Without Axles) (USE THIS TO UNDERSTAND BASIC SUBPLOT SYNTAX):

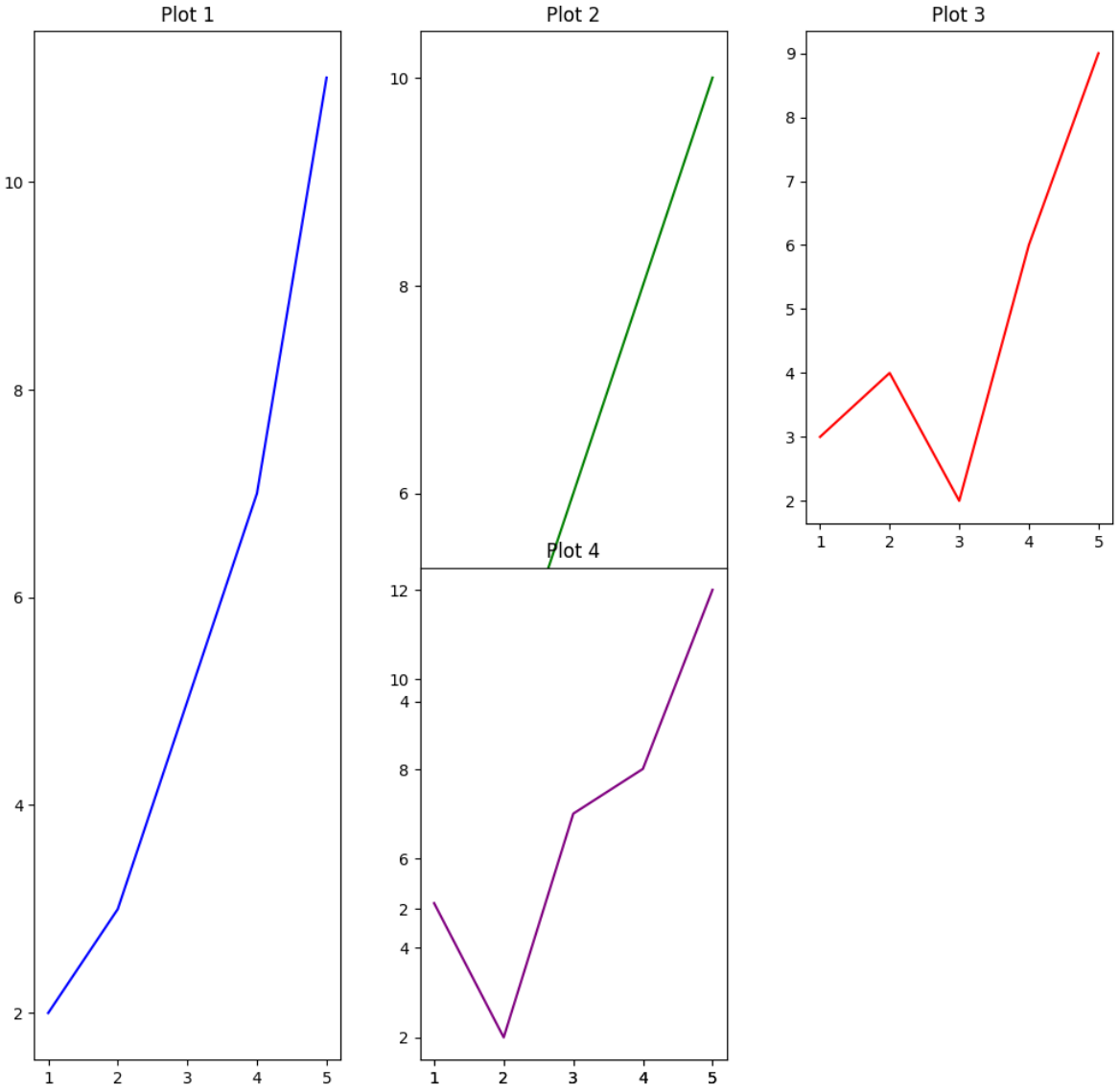
In [3]:
```
# Create x & y1 variables for first plot
x = [1, 2, 3, 4, 5]
y1 = [2, 3, 5, 7, 11]

# Create y2 variable for second plot (Note: since both plots in this exam
y2 = [1, 4, 6, 8, 10]

# Create a figureand specify size (You should change the change the numbe
plt.figure(figsize=(10, 5))

# Plot 1:
plt.subplot(1, 2, 1) # (1,2,1) means that the graph will be in Row 1, whi
plt.plot(x, y1, color='blue')
plt.title("Line Plot 1")

# Plot 2:
plt.subplot(1, 2, 2) # (1,2,1) means that the graph will be in Row 1, whi
plt.plot(x, y2, color='green')
plt.title("Line Plot 2")

# Show the plots
plt.tight_layout()  # Adjust layout for better spacing which prevents scr
plt.show()
```



b) Multiple Row Subplot (Without Axels) (**WRONG**):

In [4]:
```
# Data for the plots
x = [1, 2, 3, 4, 5]
y1 = [2, 3, 5, 7, 11]
y2 = [1, 4, 6, 8, 10]
y3 = [3, 4, 2, 6, 9]
y4 = [5, 2, 7, 8, 12]

plt.figure(figsize=(10, 10))

# Create sub plot
plt.subplot(1, 3, 1)  # top-left
plt.plot(x, y1, color='blue')
```

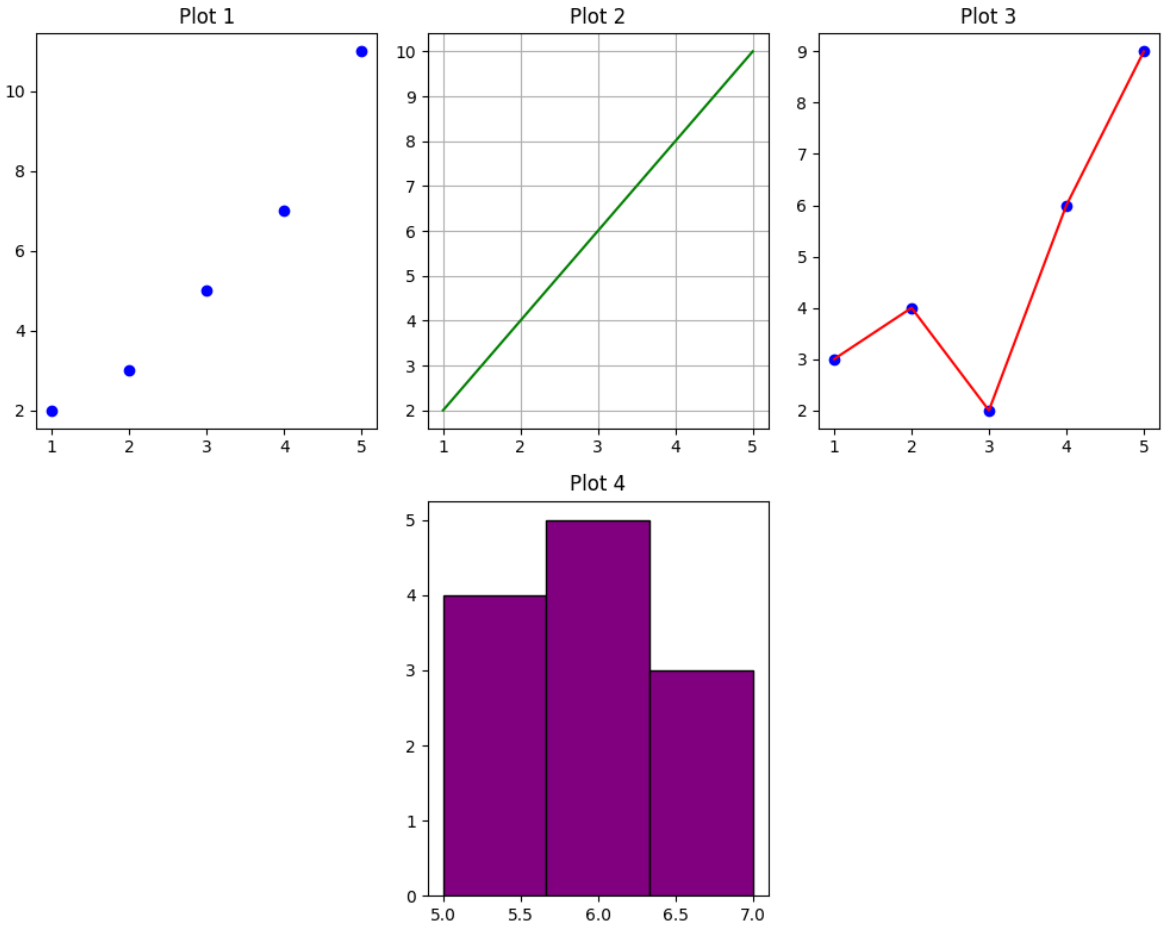
```
plt.title("Plot 1")

plt.subplot(1, 3, 2)  # top-middle
plt.plot(x, y2, color='green')
plt.title("Plot 2")

plt.subplot(2, 3, 3)  # top-right
plt.plot(x, y3, color='red')
plt.title("Plot 3")

plt.subplot(2, 3, 5)  # bottom middle (The fifth graph would have been lo
plt.plot(x, y4, color='purple')
plt.title("Plot 4")

plt.tight_layout()
plt.show()
```



c) Multiple Row Subplot (With Axels) (**CORRECT**):

In [25]:
```
# Import GridSpec to help make sure axels are sized properly
from matplotlib.gridspec import GridSpec

# Data for the plots
x = [1, 2, 3, 4, 5]
```

```
y1 = [2, 3, 5, 7, 11]
y2 = [2, 4, 6, 8, 10]
y3 = [3, 4, 2, 6, 9]
y4 = [5, 7, 7, 5, 5, 6, 6, 6, 6, 5, 6, 7]

# Create a figure and store it into a variable to make code neater (Avoid
fig = plt.figure(figsize=(10, 8))

# Makes a 3x2 "grid" for the axels to exist (Sytax: Gridspec( total rows,
gs = GridSpec(2, 3, figure=fig)

# Create Subplots (alx can be named as plot 1, this is just to help under
ax1 = fig.add_subplot(gs[0, 0]) #gs means 3x2 grid is used, the [0,0] is
ax1.scatter(x, y1, color='blue')
ax1.set_title("Plot 1")

ax2 = fig.add_subplot(gs[0, 1])  # Plot 2 (top-middle), 0=1 so that means
ax2.plot(x, y2, color='green')
plt.grid(True)
ax2.set_title("Plot 2")

ax3 = fig.add_subplot(gs[0, 2])  # Plot 3 (top-right)
ax3.plot(x, y3, color='red')
ax3.scatter(x, y3, color='blue')
ax3.set_title("Plot 3")

ax4 = fig.add_subplot(gs[1, 1])  # Plot 4 (bottom-middle)
ax4.hist(y4, bins=3, color='purple', edgecolor = "black")
ax4.set_title("Plot 4")

plt.tight_layout()
plt.show()
```



Conclusion:

Thank you for reading though this document. I hope this guid helps you learn/review/ better graphing skills with matplotlib. To check out more of my projects please visit my website Avveer's Portfolio