Basics to Stunning Plots in Matplotlib

Minor in AI, IIT Ropar

4th March, 2025

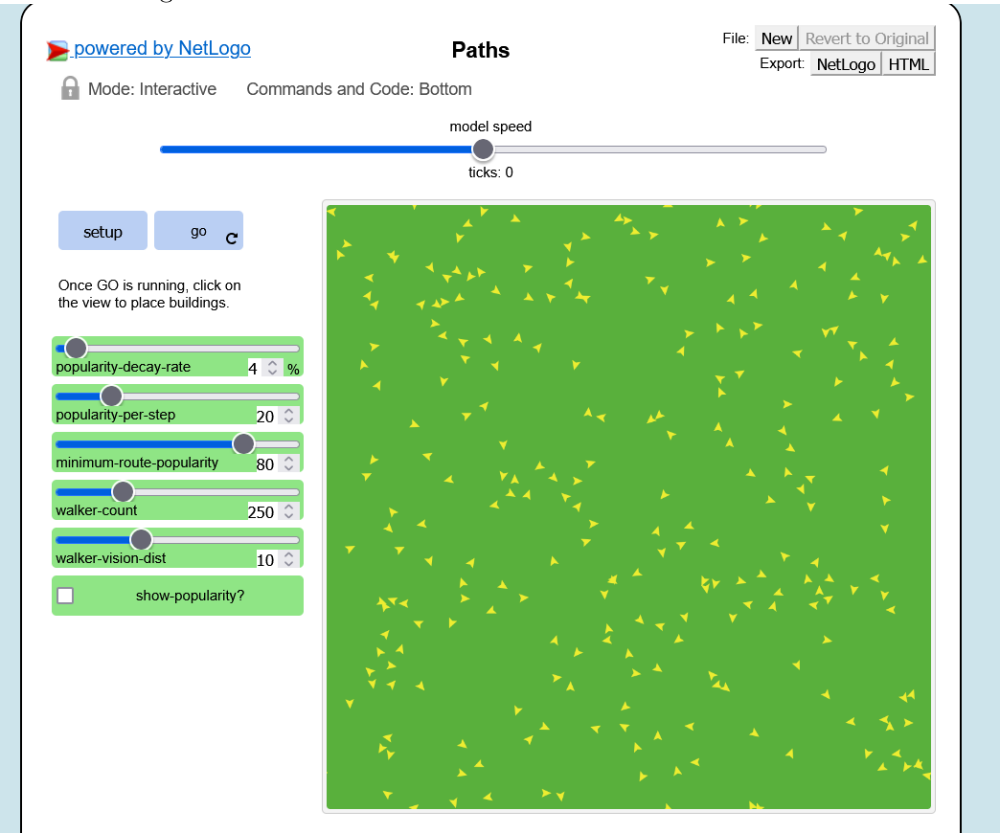
1 Design : We enforce or let it emerge?

1.1 Place Charles de Gaulle, 12 avenues

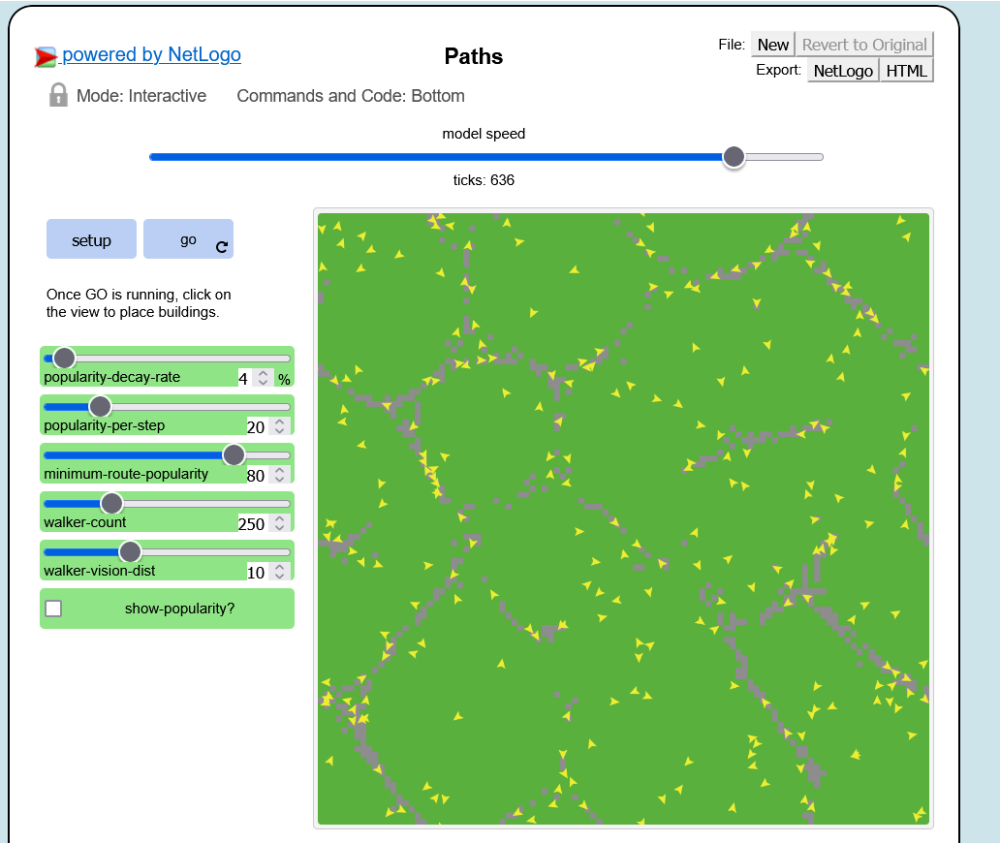
Place Charles de Gaulle is a large road junction in Paris, France, the meeting point of twelve straight avenues, its an example of complex design enforced in city planning.
There are designs that we enforce on people and, sometimes designs happens by themselves in nature.

1.2 Design emerging itself

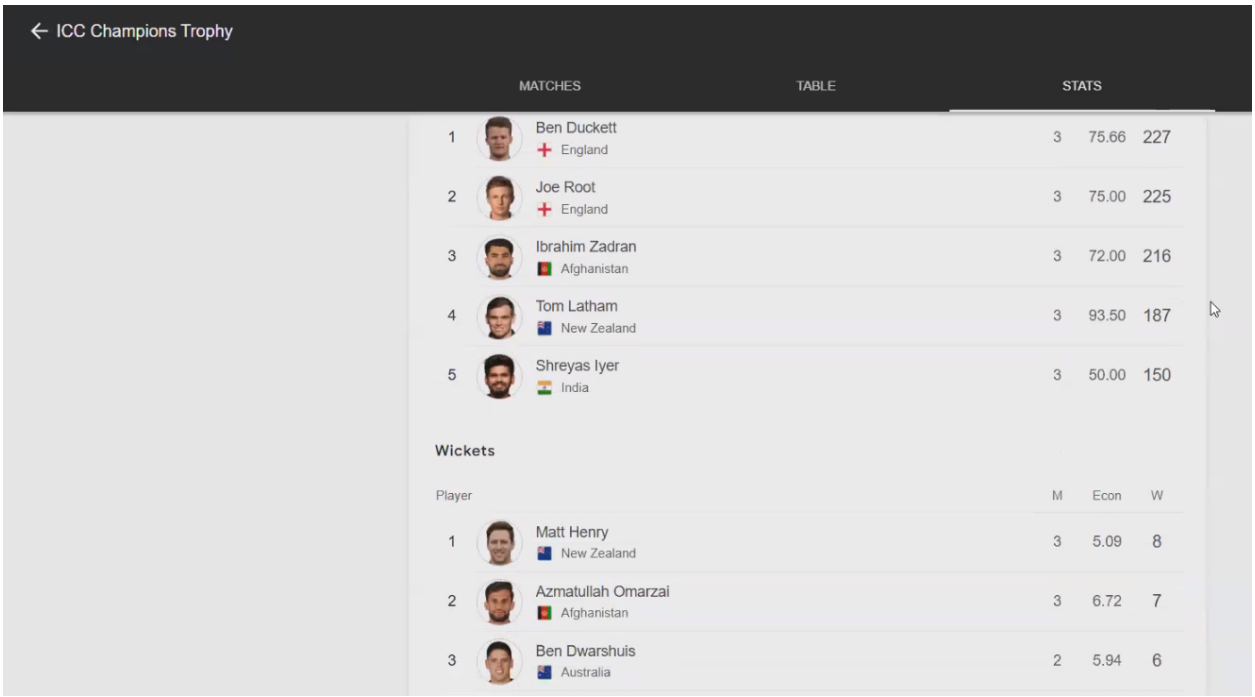
Check out NetLogo Website for simulations. Click here



After running the simulation by clicking on 1st setup and then on go button. As the agents/people walk in an area. The larger the number of people walk through a certain path or more frequent a certain path is being traversed upon then, its getting highlighted by pink colour in below pic.



2 We don't need plots at all places



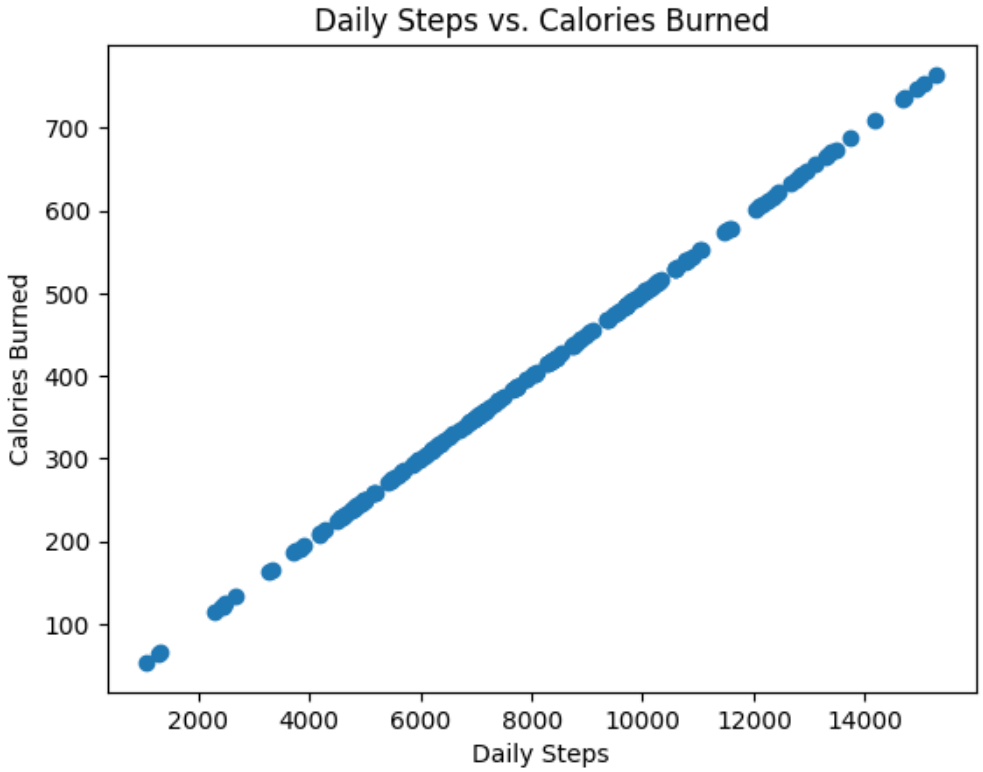
Here, the stats can be shown through line graph or, histogram or, other visualizations. But they have chosen to display in this format.

Why?
Because generally if a person wants to check out he would want to see player-wise stats. That's why when visualizing the data we should always know our audience so, that we can serve them the visuals better.

3 Daily Steps vs. Calories burned

Continuing from last class. We now add another column(attribute) named Calories Burned in the dataframe and drAw a scatter plot to find out the relationship b/w Daily steps and Calories burned.

```
# Calories burned with step counts

import pandas as pd
import matplotlib.pyplot as plt

# Sample Data
df = pd.read_csv("daily_steps.csv")
df['Calories_Burned'] = df['Daily_Steps'] * 0.05

# Plotting
plt.scatter(df['Daily_Steps'], df['Calories_Burned'])
plt.xlabel("Daily Steps")
plt.ylabel("Calories Burned")
plt.title("Daily Steps vs. Calories Burned")
plt.show()
```

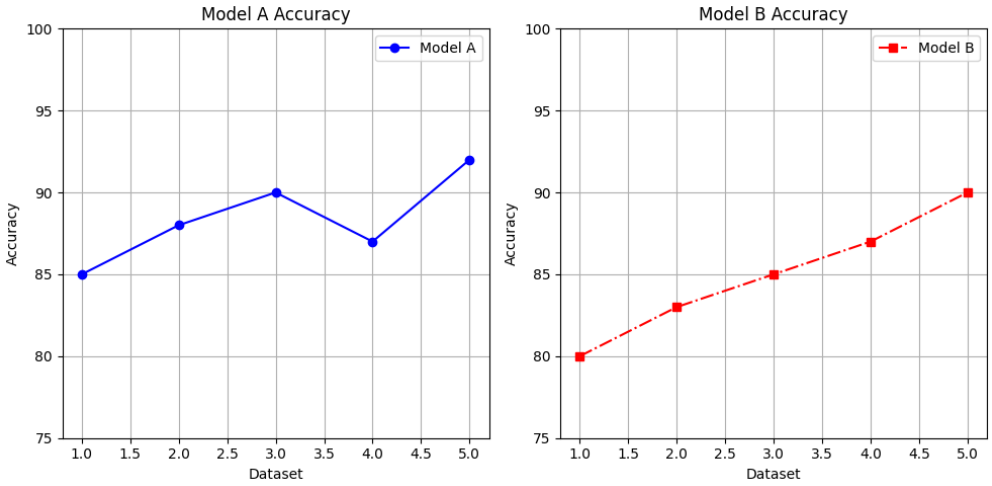


In above scatter plot we can see that the graph is linear with positive inclination. i.e. As the Daily steps increase so, does the calories burned. Therefore, There is **positive correlation** b/w Daily Steps and Calories burned.

4 Subplots

We can use subplots to plot multiple subplots in a plot. Play around with the below code to get more familiar.

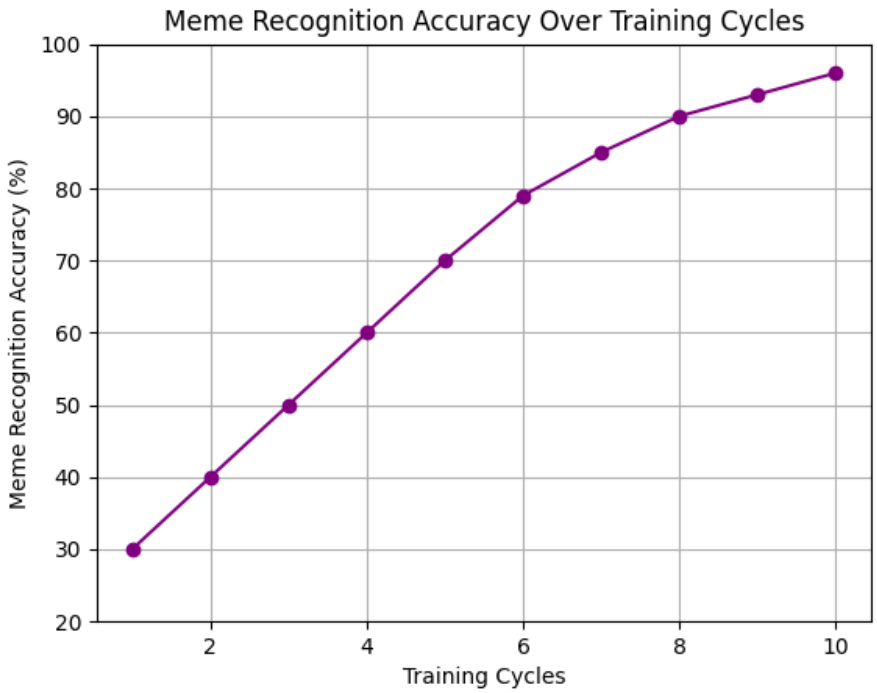
```
# Program for Subplots
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 5))
datasets = [1, 2, 3, 4, 5]

acc_model_a = [85, 88, 90, 87, 92]
acc_model_b = [80, 83, 85, 87, 90]


plt.subplot(1, 2, 1)
plt.plot(datasets, acc_model_a, marker='o', linestyle='-', color='blue', label='Model A')
plt.xlabel('Dataset')
plt.ylabel('Accuracy')
plt.title('Model A Accuracy')
plt.ylim(75, 100)
plt.grid()
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(datasets, acc_model_b, marker='s', linestyle='-.', color='red', label='Model B')
plt.xlabel('Dataset')
plt.ylabel('Accuracy')
plt.title('Model B Accuracy')
plt.ylim(75, 100)
plt.grid()
plt.legend()

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



5 Meme Accuracy

If we have a model that can recognise if a given image/content is meme or not. Then we can plot a line graph for depicting meme recognition accuracy for each training cycle as below.

```
# Meme Accuracy
import matplotlib.pyplot as plt

training_cycles = list(range(1, 11))
meme_recognition = [30, 40, 50, 60, 70, 79, 85, 90, 93, 96]

plt.plot(training_cycles, meme_recognition, marker='o', color = 'purple', linestyle = '-')
plt.title('Meme Recognition Accuracy Over Training Cycles')
plt.xlabel('Training Cycles')
plt.ylabel('Meme Recognition Accuracy (%)')
plt.ylim(20, 100)
plt.grid(True)

plt.savefig("meme_accuracy.png", dpi = 100)

plt.show()
```

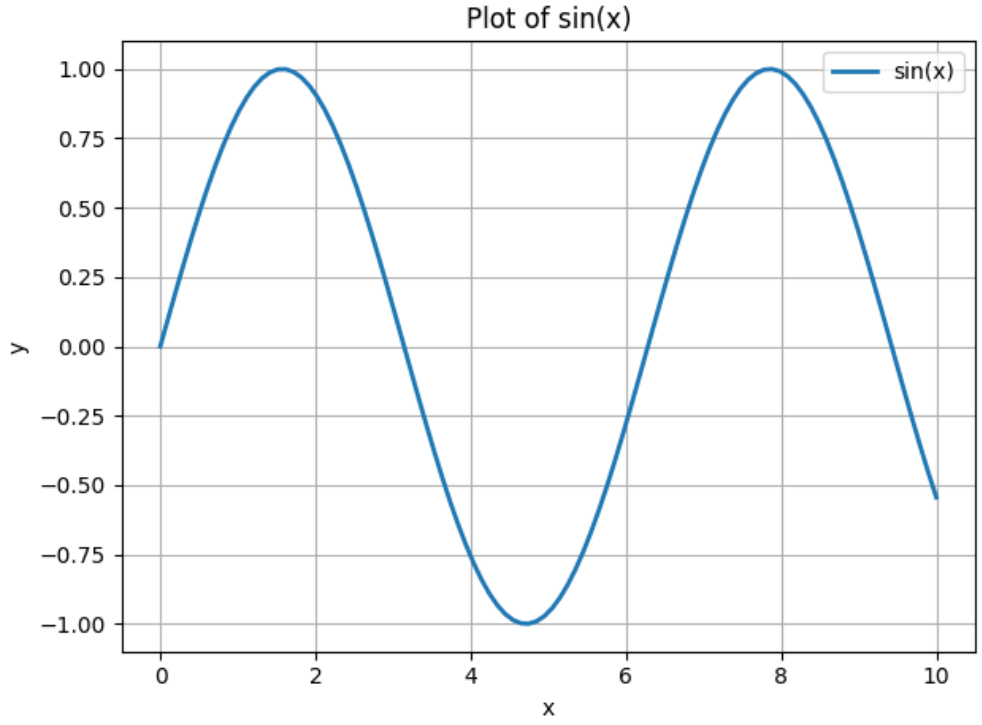


The graph is depicting an upward trend in meme recognition accuracy as the number of training cycles increases.

6 Curves

linspace() function in numpy can be used to create sine and cos plots. Try creating cos plot by examining how sine plot is created in below code.

```
# High resolution and optimizations
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 100)
y = np.sin(x)

plt.plot(x, y, label='sin(x)', linewidth=2)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Plot of sin(x)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("sine_plot.pdf", dpi=300)

plt.show()
```

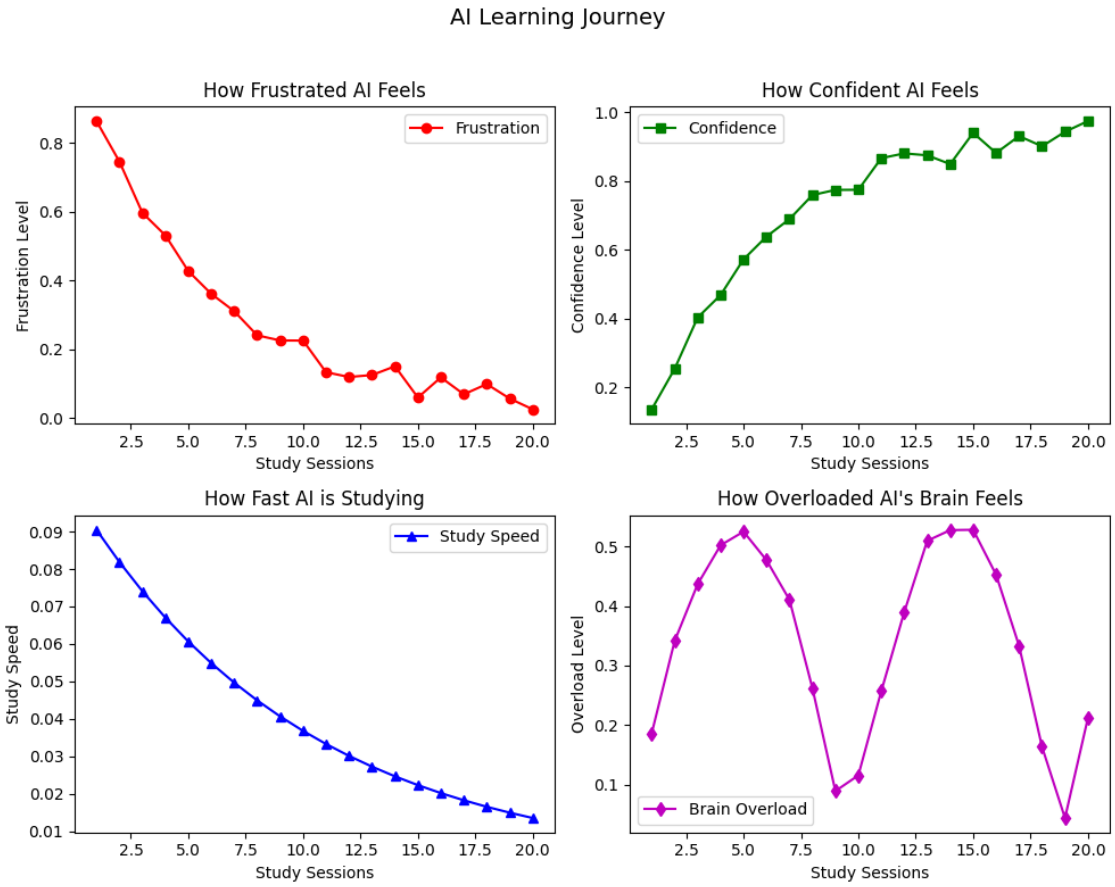


7 AI Learning Journey

Here we are simulating study sessions and different emotions the AI can feel like frustration, confidence, study speed, brain overload like a human.

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

# Simulated "mood swings" over 20 study sessions
sessions = np.arange(1, 21)
frustration = np.exp(-sessions/5) + np.random.uniform(0, 0.1, len(sessions))
```

```
confidence = 1 - frustration
study_speed = np.exp(-sessions/10) * 0.1
brain_overload = np.abs(np.sin(sessions / 3)) * 0.5 + np.random.uniform(0, 0.05, len(sessions))

# Create 4 subplots
fig, axes = plt.subplots(2, 2, figsize=(10, 8))
fig.suptitle("AI Learning Journey", fontsize=14)

# Frustration Level
axes[0, 0].plot(sessions, frustration, 'r-o', label="Frustration")
axes[0, 0].set_title("How Frustrated AI Feels")
axes[0, 0].set_xlabel("Study Sessions")
axes[0, 0].set_ylabel("Frustration Level")
axes[0, 0].legend()

# Confidence Level
axes[0, 1].plot(sessions, confidence, 'g-s', label="Confidence")
axes[0, 1].set_title("How Confident AI Feels")
axes[0, 1].set_xlabel("Study Sessions")
axes[0, 1].set_ylabel("Confidence Level")
axes[0, 1].legend()

# Study Speed
axes[1, 0].plot(sessions, study_speed, 'b-^', label="Study Speed")
axes[1, 0].set_title("How Fast AI is Studying")
axes[1, 0].set_xlabel("Study Sessions")
axes[1, 0].set_ylabel("Study Speed")
axes[1, 0].legend()

# Brain Overload
axes[1, 1].plot(sessions, brain_overload, 'm-d', label="Brain Overload")
axes[1, 1].set_title("How Overloaded AI's Brain Feels")
axes[1, 1].set_xlabel("Study Sessions")
axes[1, 1].set_ylabel("Overload Level")
axes[1, 1].legend()

# Adjust layout
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



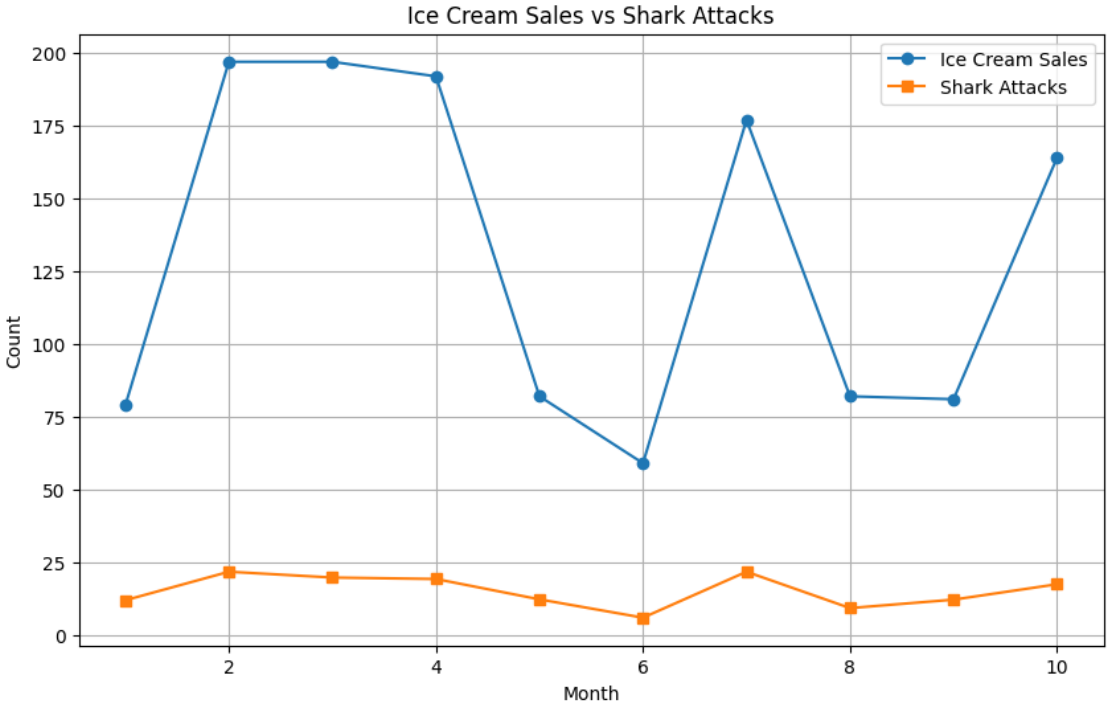
8 Correlation doesn't mean Causation

8.1 Ice Cream Sales and Shark Attacks

An Intern was asked to analyze data on ice cream sales and shark attacks over the summer months. He noticed that as ice cream sales go up, shark attacks also increase! To prove his point, he made a line plot connecting the two trends.
"Look! The lines move together! That must mean eating ice cream causes sharks to attack!" He is wrong to conclude that. In this case what we are observing is the as one variable increase another is also increasing, i.e. there is a positive correlation. BUT! that just means for the given data they are positively correlated it DOES NOT mean one variable is causing the other variable to increase or decrease with itself. // The possible reason for both lines to be increasing together is that the season would be summer therefore, people are more likely to eat ice cream and also more likely to go to open waters where sharks are thus, increase in shark attacks.

```
# prompt: An Intern was asked to analyze data on ice cream sales and shark attacks over the summer montl

import matplotlib.pyplot as plt
import numpy as np

# Simulate ice cream sales data
ice_cream_sales = np.random.randint(50, 200, size=10)  # 10 data points

# Simulate shark attack data (correlated with ice cream sales)
shark_attacks = ice_cream_sales * 0.1 + np.random.randint(0, 5, size=10)  # Introduce some noise
```

```
# Months
months = range(1, 11)


plt.figure(figsize=(10, 6))
plt.plot(months, ice_cream_sales, label='Ice Cream Sales', marker='o', linestyle='-')
plt.plot(months, shark_attacks, label='Shark Attacks', marker='s', linestyle='-')

plt.xlabel("Month")
plt.ylabel("Count")
plt.title("Ice Cream Sales vs Shark Attacks")
plt.legend()
plt.grid(True)
plt.show()
```



8.2 Concert Ticket Price

Intern was asked to analyze how concert ticket prices impact the number of people attending a concert for different events. He decided to use a histogram to show the relationship. "I made a histogram showing how many concerts had different ticket price ranges! That should explain everything, right?"

Think about this. It will be covered in next session.