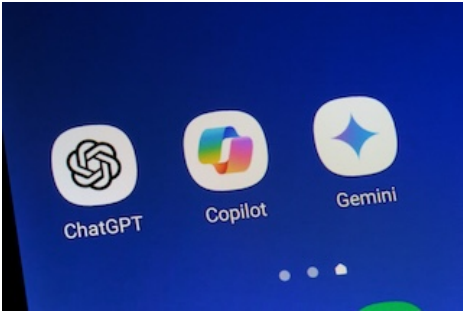


You are a data analyst working for a tech company closely monitoring the AI tools market. You want to understand the evolving popularity of three major AI tools, ChatGPT, Gemini, and Microsoft Copilot, and identify which tool is gaining the most traction and how they compare over time.



You'll work with real-world time series data that captures the global interest in these tools over several weeks. Your goal is to analyze this data, uncover patterns, and provide actionable insights to help your company make informed decisions. This might include determining where to focus marketing efforts, predicting future trends, or identifying potential areas for improvement.

Are you ready to help your company stay ahead of the curve in the competitive AI tools market? Let's get started!

The Data

The Google Trends data is available as a CSV file `ai_tools_comparison.csv`. The data contains the number of worldwide searches for chatGpt, Gemini, and Microsoft Copilot over the past 12 months as of September 2024.

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data
trends = pd.read_csv('ai_tools_comparison.csv')

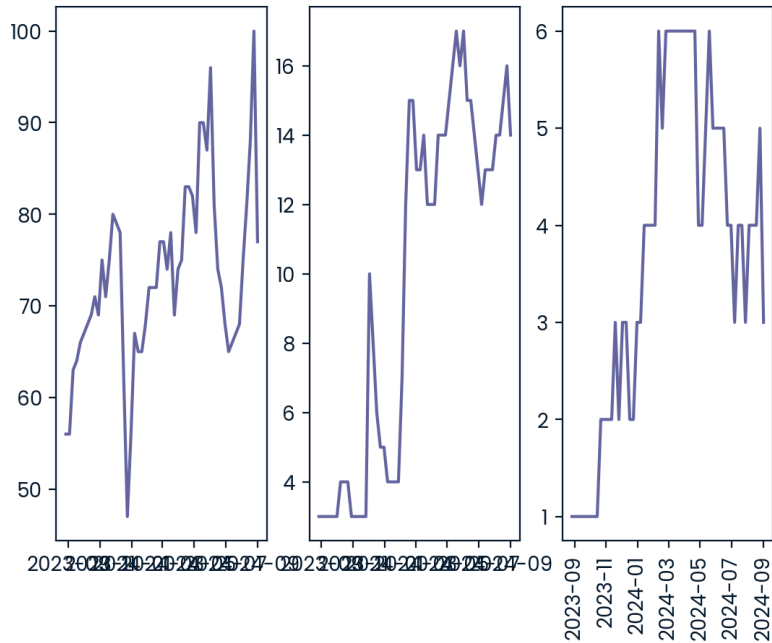
# Inspect the data
trends.head()
```

index	...	↑↓	week	...	↑↓	chatgpt	...	↑↓	gemini	...	↑↓	microsoft_copilot	...
		0	2023-08-27			56			3				
		1	2023-09-03			56			3				
		2	2023-09-10			63			3				
		3	2023-09-17			64			3				
		4	2023-09-24			66			3				

Rows: 5

↗ Expand

```
fig, axs = plt.subplots(1, 3)
trends['week'] = pd.to_datetime(trends['week'])
trends = trends.set_index('week')
chatgpt = trends['chatgpt']
gemini = trends['gemini']
copilot = trends['microsoft_copilot']
axs[0].plot(chatgpt)
axs[1].plot(gemini)
axs[2].plot(copilot)
most_consistent_tool = 'chatgpt'
plt.xticks(rotation=90);
```



```
fig2, ax2 = plt.subplots()
ax2.plot(chatgpt)
plt.xticks(rotation=90);
gpt_dip = "January 2024"
```



```
trends['month']=trends.index.month
monthly_average=trends.groupby('month')[['chatgpt', 'gemini', 'microsoft_copilot']].mean()
monthly_average['per_month']=monthly_average.mean(axis="columns")
best_month='May'
monthly_average
```

...	↑↓	...	↑↓	...	↑↓	microsoft_co...	...	↑↓	per_m...	...	↑↓	
1		66.25		4				3.75	24.6666666667			
2		73.25		12.25				5.25	30.25			
3		74.4		12.8				6	31.0666666667			
4		80.75		13.5				5.5	33.25			
5		86.25		16				5	35.75			
6		78.2		14.8				4.6	32.5333333333			
7		66.5		12.75				3.5	27.5833333333			
8		80		12.4				3.6	32			
9		65.2		5.2				1.4	23.9333333333			
10		68.8		3.6				1.4	24.6			
11		75.25		3				2.25	26.8333333333			
12		64.4		6.8				2.6	24.6			