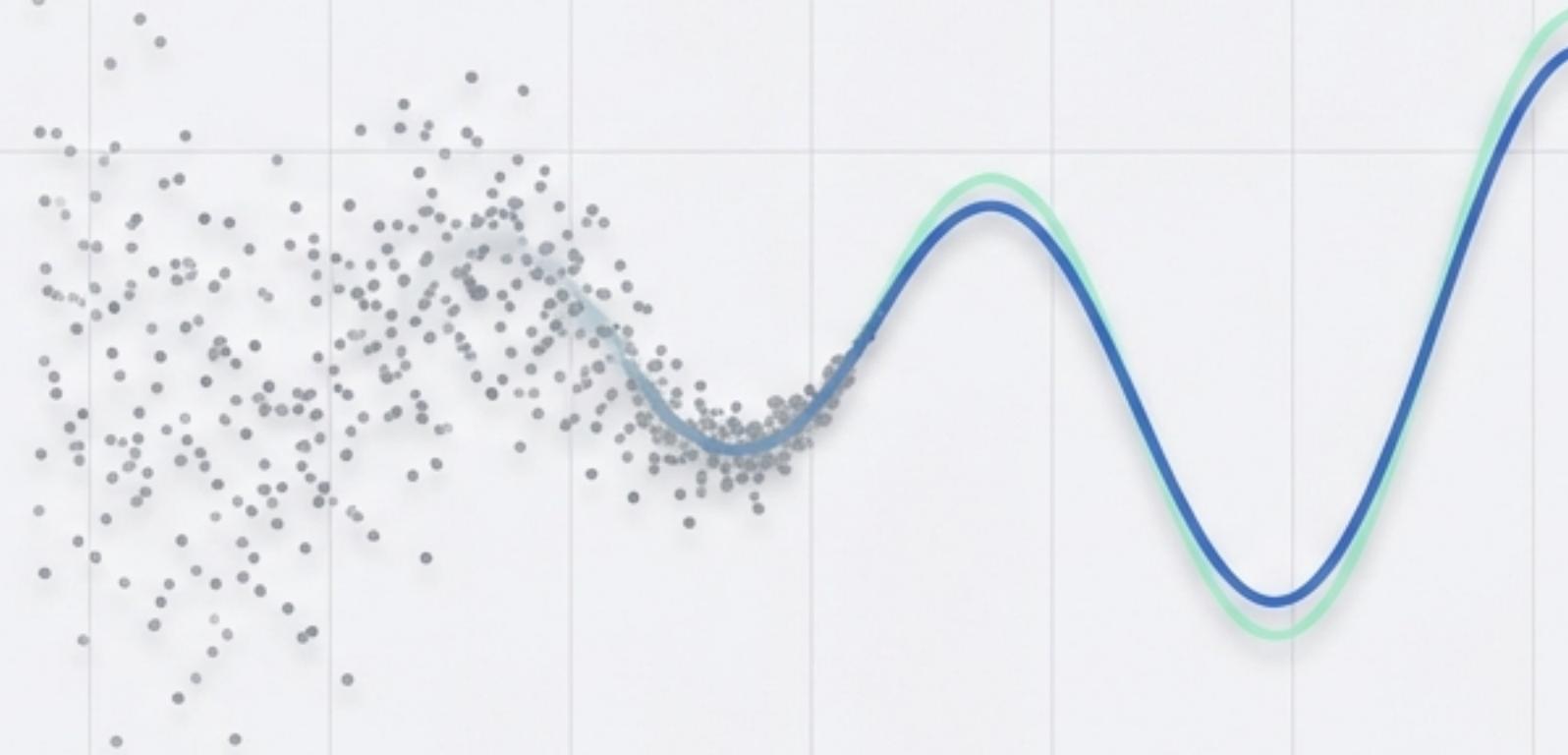


AI 활용 프로그래밍 Week 11: 데이터 시각화 핸드북

Matplotlib와 LLM으로 데이터의 패턴 발견하기

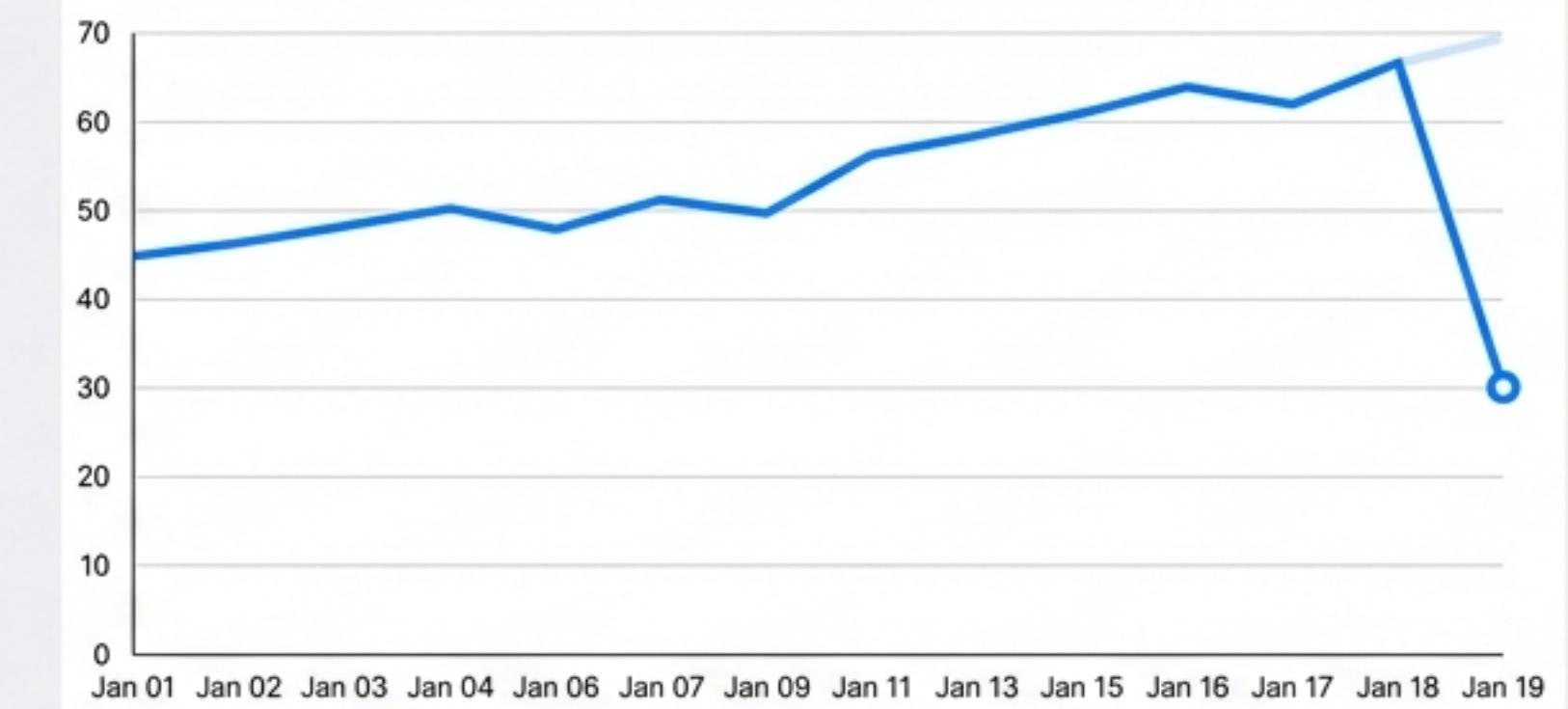


AI Utilization Programming | Python Matplotlib & AI Collaboration

숫자의 뒤에 숨겨진 패턴을 보는 법

| Date | Value |
|------------|-------|
| 2023-01-01 | 45.2 |
| 2023-01-02 | 46.8 |
| 2023-01-03 | 48.5 |
| 2023-01-04 | 50.1 |
| 2023-01-05 | 47.9 |
| 2022-01-06 | 51.3 |
| 2023-01-07 | 49.7 |
| 2023-01-08 | 52.4 |
| 2023-01-09 | 54.1 |
| 2023-01-10 | 56.7 |
| 2023-01-11 | 58.2 |
| 2022-01-12 | 59.9 |
| 2023-01-13 | 61.5 |
| 2023-01-14 | 63.8 |
| 2023-01-15 | 62.1 |
| 2023-01-16 | 65.4 |
| 2023-01-17 | 55.2 |
| 2023-01-18 | 38.2 |
| 2023-01-19 | 67.1 |
| 2023-01-20 | 69.3 |

표(Table)는 정확하지만, 패턴을 한눈에 파악하기 어렵습니다.



그래프(Graph)는 추세, 분포, 그리고 이상치(Outlier)를 즉각적으로 보여줍니다.

- 질문이 명확해야 합니다 (무엇을 보여주는가?)
- 축, 단위, 범례가 필수적입니다.
- 과장 없이 읽기 쉬워야 합니다.

Matplotlib의 3단계 기본 워크플로우



**Step 1: 준비
(Import & Data)**

```
import matplotlib.pyplot as plt
```



**Step 2: 그리기
(Plot)**

```
plt.plot(x, y)
```



**Step 3: 꾸미기 & 출력
(Style & Show)**

```
plt.title(...) → plt.show()
```

```
import matplotlib.pyplot as plt

x = [1, 2, 3]
y = [10, 20, 15]

plt.plot(x, y)
plt.show()
```



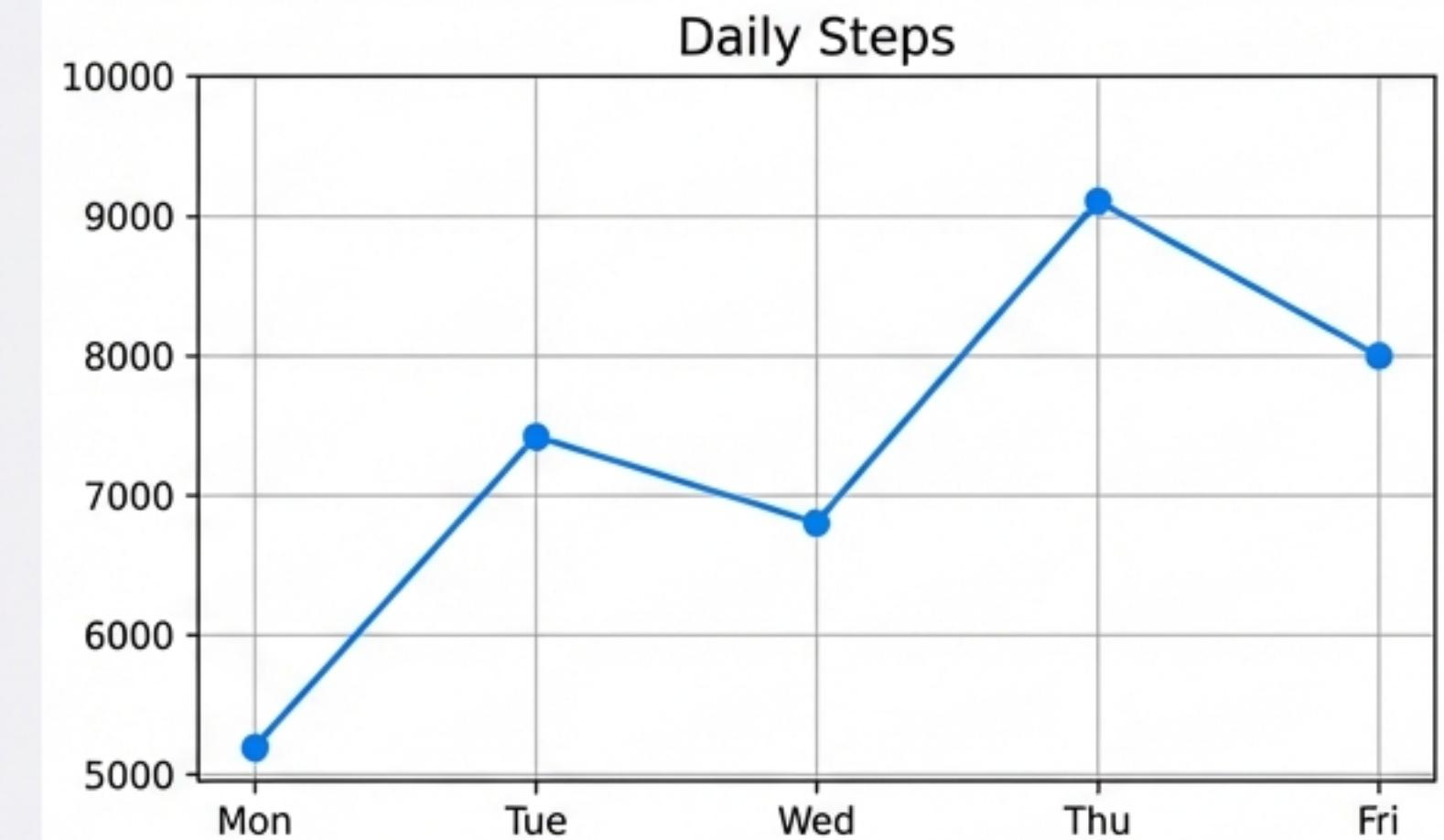
Success Check

선그래프 1개가 뜨고, 제목
과 축 라벨이 보이면 성공.

선그래프 (Line Chart) – 시간의 흐름과 변화

```
days = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri']  
steps = [5200, 7400, 6800, 9100, 8000]
```

```
# marker='o'는 데이터 포인트를 점으로 표시  
plt.plot(days, steps, marker='o')  
plt.title('Daily Steps')  
plt.grid(True) # 격자 추가  
plt.show()
```



언제 사용하는가?

시계열 데이터(날짜, 시간)의 추세를 확인할 때.
데이터가 많으면 marker를 생략하세요.

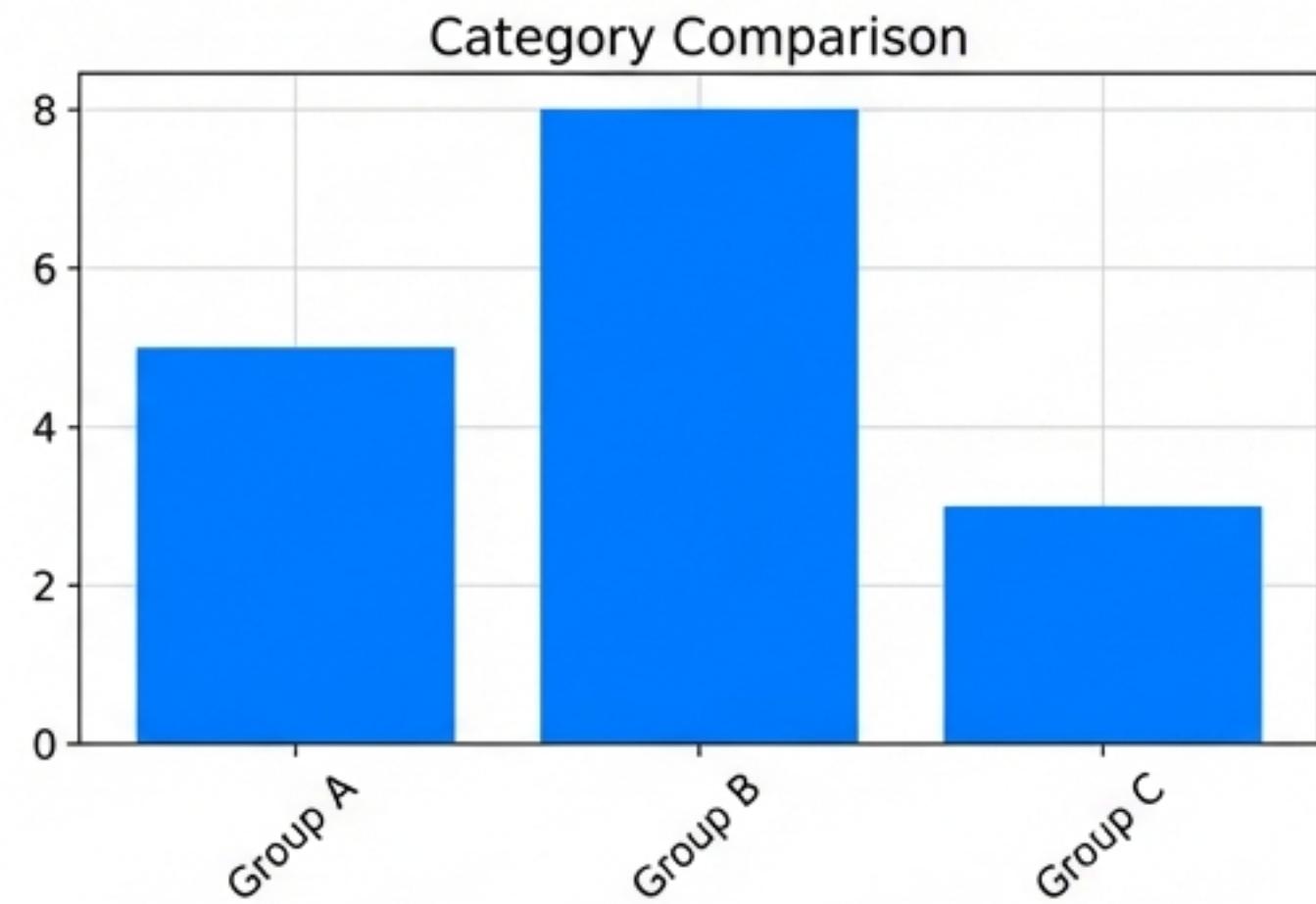
✓ Success Check

요일별 걸음 수 추세를 선으로 읽을 수 있다.

막대그래프 (Bar Chart) — 범주 간의 크기 비교

```
labels = ['Group A', 'Group B', 'Group C']
values = [5, 8, 3]

plt.bar(labels, values)
plt.title('Category Comparison')
# 라벨이 길다면 회전(rotation) 고려
plt.xticks(rotation=45)
plt.show()
```



Key Insight

언제 사용하는가? 그룹 간의 순위나 크기를 명확히 비교할 때. Y축이 0부터 시작해야 왜곡이 없습니다.

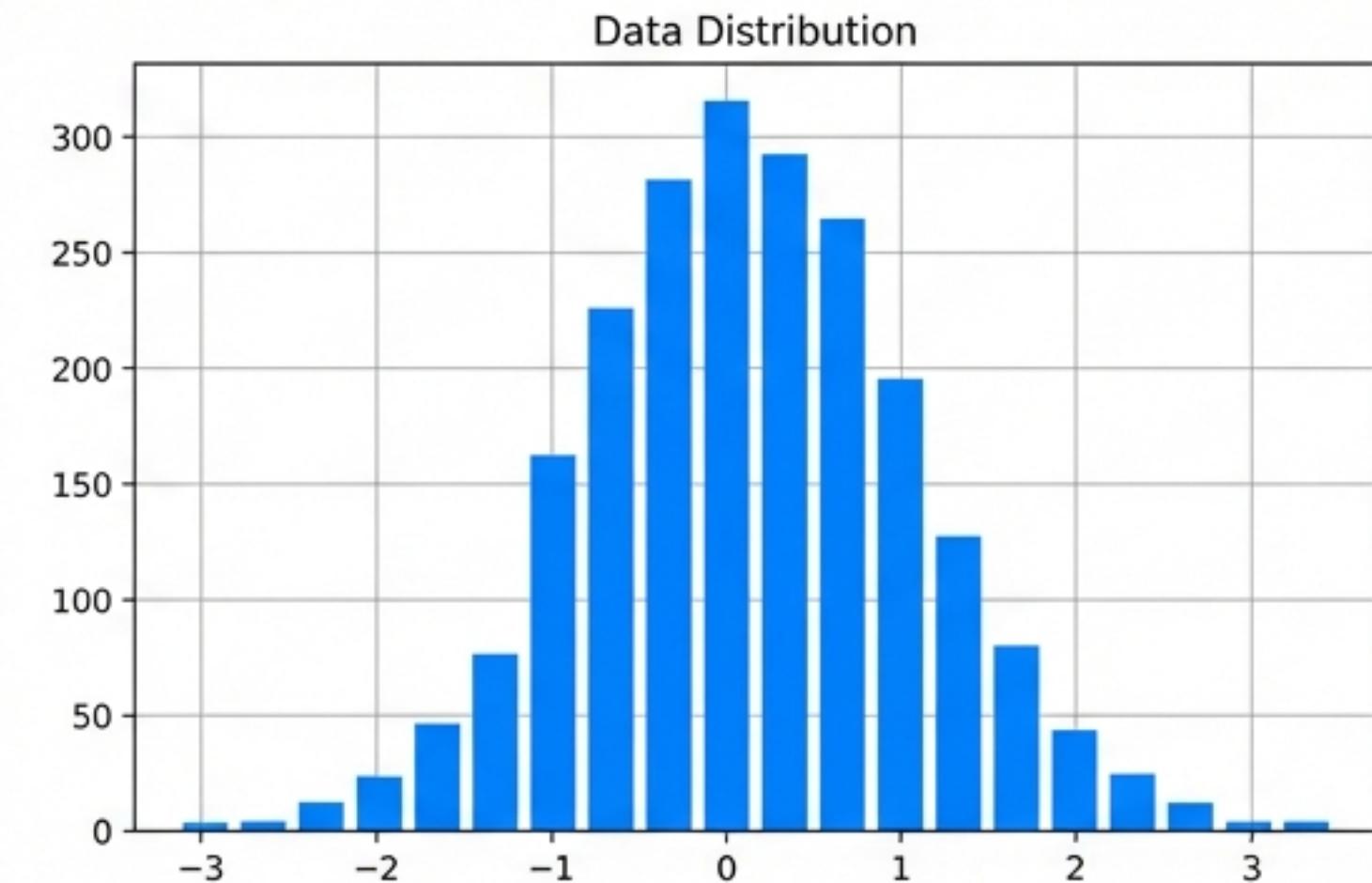


Success Check

A/B/C 막대가 나타나고, B가 가장 큰 것이 확인된다.

히스토그램 (Histogram) — 데이터의 분포와 모양

```
import numpy as np  
data = np.random.randn(1000)  
  
# bins: 데이터를 나누는 구간의 개수  
plt.hist(data, bins=20)  
plt.title('Data Distribution')  
plt.show()
```



Key Insight

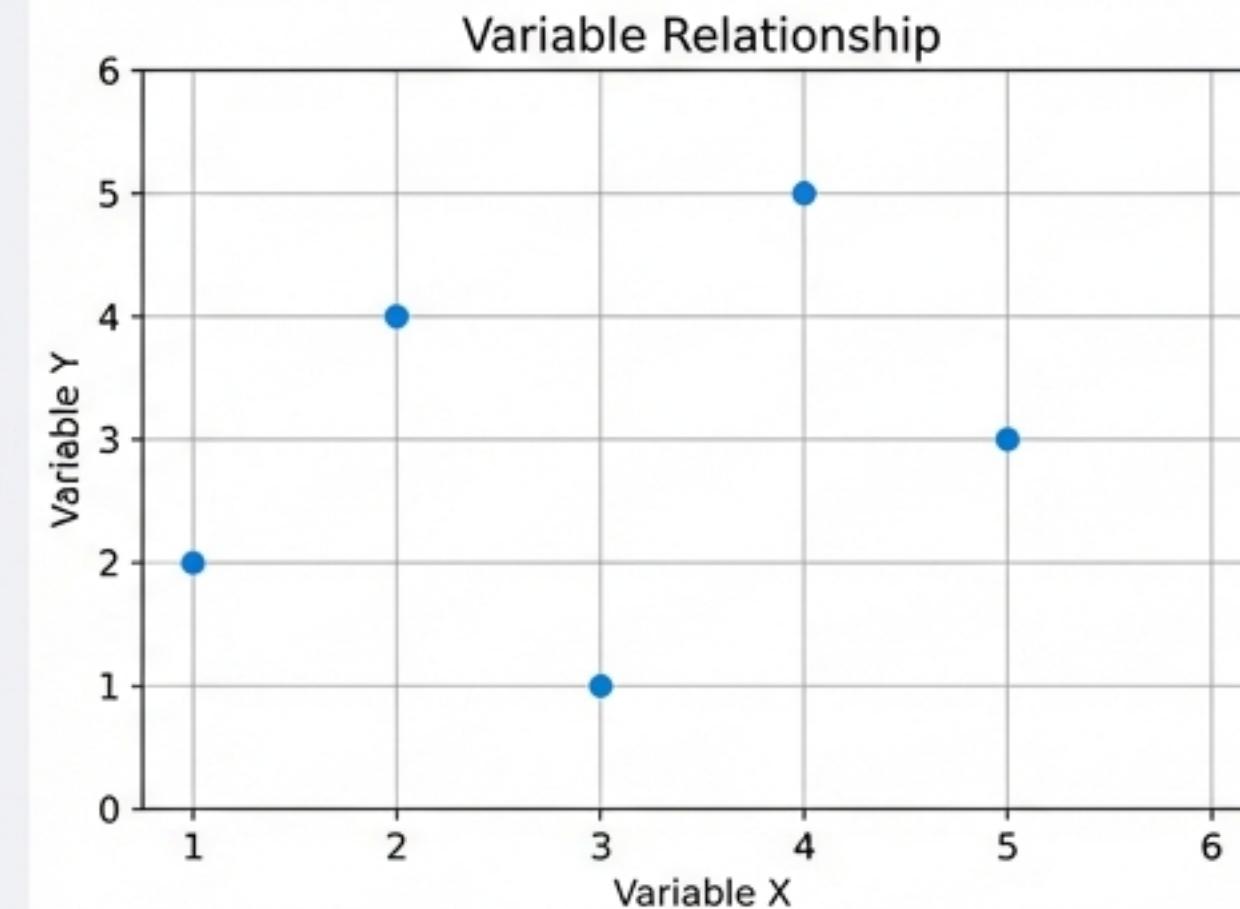
언제 사용하는가? 데이터가 특정 구간에 몰려있는지(치우침), 혹은 전반적인 모양(정규분포 등)을 볼 때.

Success Check

막대가 연속된 구간으로 나오며, 데이터의 뭉침(모양)이 보인다.

산점도 (Scatter Plot) — 두 변수의 상관관계

```
x = [1, 2, 3, 4, 5] # 변수 1  
y = [2, 4, 1, 5, 3] # 변수 2  
  
plt.scatter(x, y)  
plt.title('Variable Relationship')  
plt.xlabel('Variable X')  
plt.ylabel('Variable Y')  
plt.show()
```



Key Insight

언제 사용하는가? X가 증가할 때 Y가 어떻게 변하는지(양/음의 상관관계) 확인하거나, 튀는 점(Outlier)을 찾을 때.

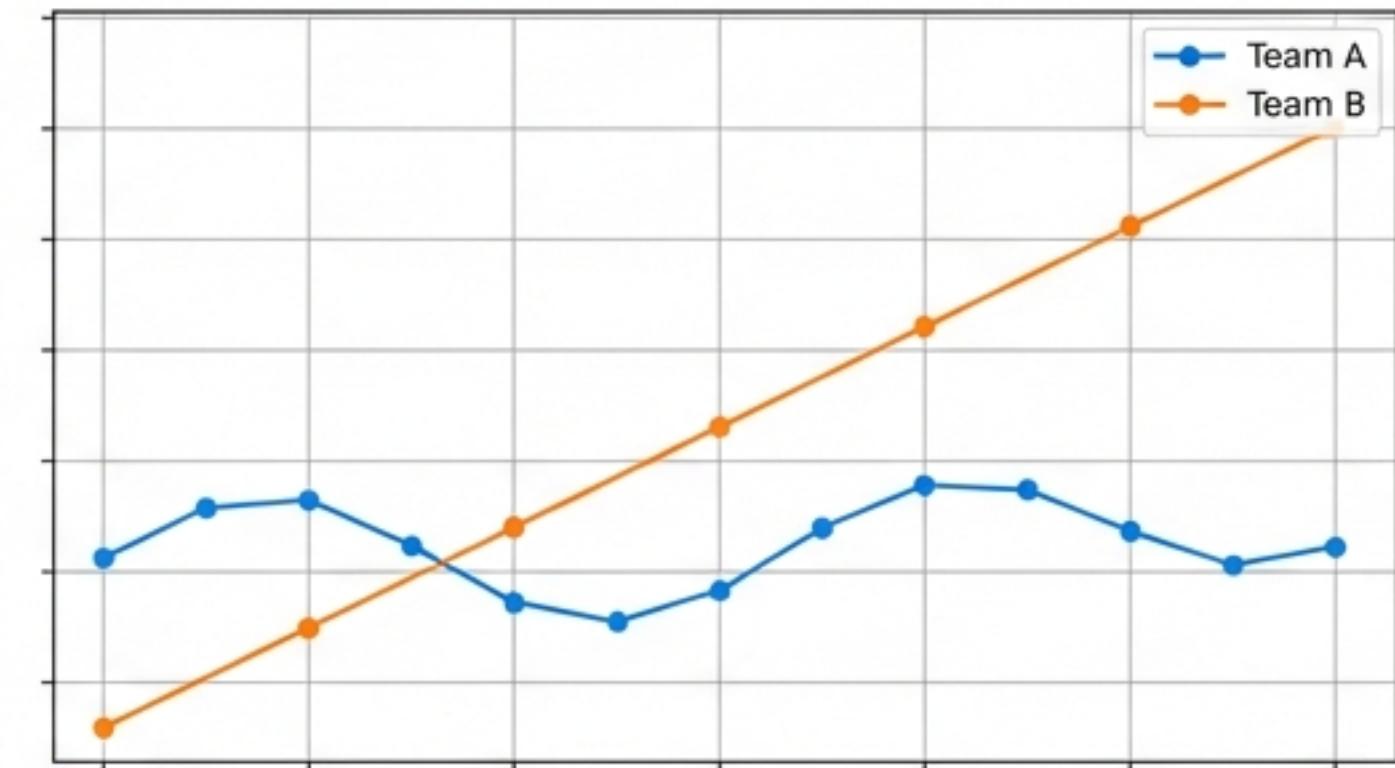
Success Check

점들이 좌표에 찍히고, 그 퍼짐을 통해 관계를 유추할 수 있다.

그래프의 문법 – 여러 데이터 비교하기 (Legend)

여러 선을 그릴 때는 반드시 `label`을 지정하고 `plt.legend()`를 호출해야 합니다.

```
# 데이터 2개 준비  
plt.plot(x, a, label='Team A') # 라벨 지정  
plt.plot(x, b, label='Team B') # 라벨 지정  
  
plt.legend() # 범례 표시 명령  
plt.show()
```



- ✓ 두 개의 선이 보이고, 범례(Legend)에 각 선의 이름이 표시된다.

Pandas Shortcut — 데이터프레임에서 바로 그리기

Concept

복잡한 설정 없이 Pandas 데이터프레임
뒤에 `.plot()`만 붙이면 됩니다.

Syntax

```
df['column'].plot(kind='...')
```

- kind='line' (기본값)
- kind='bar'
- kind='hist'
- kind='scatter'

Example

```
# 데이터 준비
df = pd.DataFrame({'score': [70, 85, 90,
60]})

# 히스토그램 그리기
df['score'].plot(kind='hist', bins=5)
plt.show()
```



Success Check

Pandas 데이터를
이용해 히스토그램이
바로 출력된다.

결과 저장하기 – 화면 밖으로 그래프 꺼내기

반드시 `plt.show()` 전에 `plt.savefig()`를 실행해야 합니다.
(`show()` 이후에는 그래프가 초기화되어 빈 파일이 저장됩니다.)

```
plt.plot([1, 2, 3], [2, 3, 5])  
plt.title('Final Report')  
  
# 파일명과 확장자(.png) 지정  
plt.savefig('my_graph.png') \  
plt.show()
```



Success Check



코드가 실행된 폴더(또는 Colab 파일 탭)에
'my_graph.png' 파일이 생성된다.

With AI — 데이터에 맞는 차트 추천받기

어떤 그래프를 그려야 할지 모를 때, AI에게 컬럼과 질문을 함께 던지세요.

With AI

너는 데이터 분석가야. 내가 가진 컬럼은 date, steps, sleep_hours야.
'수면 시간과 걸음 수의 관계'를 보고 싶은데 어떤 그래프가 좋을까?
이유와 Python 코드를 알려줘.

두 변수의 관계를 보기 위해서는 **산점도(Scatter Plot)**가 적합합니다.
X축에 수면 시간, Y축에 걸음 수를 배치하여 상관관계를 확인하세요.

```
```python
import matplotlib.pyplot as plt
plt.scatter(df['sleep_hours'], df['steps'])
plt.xlabel('수면 시간')
plt.ylabel('걸음 수')
plt.title('수면 시간과 걸음 수의 관계')
plt.show()
```
```

'왜(Why)'를 물어봐야 선택의 근거를 배울 수 있습니다.

With AI — 오류 해결 (Debugging)

‘x’와 ‘y’의 데이터 개수가 달라서 에러가 난 상황.

With AI

아래 코드에서 에러가 나. 원인과 해결책을 알려줘.

```
plt.plot(x, y)
```

ValueError: x and y must have same first dimension

x와 y의 리스트 길이가 다릅니다. `len(x)`와 `len(y)`를 출력해서 길이를 확인하고 데이터를 맞춰주세요.

에러 메시지 원문을 그대로 복사해서 붙여넣으세요.
데이터의 길이(shape)를 확인하는 코드를 요청하세요.

With AI — 결과 해석 및 리포트 작성

그래프만으로는 부족합니다. AI를 활용해 ‘2줄 인사이트’를 작성하세요.

With AI

이 그래프의 요약 통계를 바탕으로 3문장 해석을 써줘.

조건: 과장 금지(사실 기반), 이상치 언급 포함, 다음 분석 제안 1개.

- 수면 시간이 길수록 점수가 높은 양의 상관관계를 보입니다.
- 다만, 수면 시간이 짧은데도 고득점인 이상치가 하나 존재합니다.
- 향후 수면의 질 데이터도 함께 분석해보는 것을 제안합니다.



AI가 생성한 문장이 실제 그래프와 일치하는지 반드시 눈으로 대조해야 합니다.

자주 발생하는 오류 해결 (Troubleshooting)

-  **한글 깨짐:** 폰트 설정이 필요합니다. (OS별 설정 확인)
-  **그래프가 안 뜹:** `plt.show()`가 누락되지 않았는지 확인하세요.
-  **데이터 길이 불일치 (ValueError):** X와 Y의 개수가 같은지 `len()`으로 확인하세요.
-  **NaN (결측치) 존재:** 데이터에 빈 값이 있으면 그려지지 않을 수 있습니다.
-  **해석 오류:** 축의 단위를 확인하지 않으면 잘못된 결론을 내릴 수 있습니다.

미니 EDA 리포트 완성하기 (Checklist)

- Select:** CSV 데이터 선택 (학습시간, 걸음 수 등)
- Visualize:** 그래프 2종류 이상 그리기 (선, 막대, 히스토그램, 산점도 중 택)
- Label:** 제목, 축 라벨, 범례 포함하기
- Interpret:** 그래프를 보고 읽을 수 있는 결론 2~3줄 작성 (AI 활용)
- Save:** `savefig`로 이미지 저장 후 제출

데이터는 그리는 것(Plot)에서 끝나지 않습니다. 해석(Insight)해야 완성됩니다.