

기본 플로팅 (Basic Plotting)

Matplotlib의 `pyplot` 모듈을 사용하여 데이터의 특성에 맞는 기초적인 그래프를 생성합니다.

라인 플롯 (Line Plot)

연속적인 값의 변화를 보여줄 때 유용합니다.

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

```
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
```

```
plt.plot(x, y)
plt.title("간단한 라인 플롯 예시")
plt.xlabel("X축 레이블")
plt.ylabel("Y축 레이블")
plt.show()
```

산점도 (Scatter Plot)

두 변수 간의 상관관계를 점으로 표시하여 시각화합니다.

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

```
x = [1, 2, 3, 4, 5]
y = [2, 5, 3, 7, 4]
```

```
plt.scatter(x, y)
plt.title("산점도 시각화")
plt.xlabel("X값")
plt.ylabel("Y값")
plt.show()
```

막대 그래프 (Bar Chart)

범주형 데이터의 빈도나 수치를 비교할 때 주로 사용됩니다.

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

```
categories = ['A', 'B', 'C']
values = [10, 20, 15]
```

```
plt.bar(categories, values)
plt.title("카테고리별 수치 비교")
plt.xlabel("항목")
plt.ylabel("측정값")
plt.show()
```

스타일 사용자 정의 (Customization)

그래프의 가독성을 높이기 위해 색상, 마커, 선 스타일 및 범례를 설정합니다.

색상, 마커, 선 스타일 설정

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

```
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
```

```
# color: 색상, linestyle: 선 종류,
marker: 점 모양, markersize: 마커 크기
plt.plot(x, y, color='red',
linestyle='--', marker='o',
markersize=8)
plt.title("스타일이 적용된 플롯")
plt.grid(True) # 격자 표시
plt.show()
```

범례 (Legend) 활용

여러 데이터를 한 플롯에 그릴 때 각 선이 무엇을 의미하는지 표시합니다.

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

```
x = [1, 2, 3, 4, 5]
y1 = [2, 4, 6, 8, 10]
y2 = [1, 3, 5, 7, 9]
```

```
plt.plot(x, y1, label='그룹 A')
plt.plot(x, y2, label='그룹 B')
plt.title("범례가 포함된 그래프")
plt.legend(loc='upper left') # 범례 위치 지정
plt.show()
```

다중 그래프 구성 (Subplots)

한 화면에 여러 개의 그래프를 배치하여 비교 분석할 수 있습니다.

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

```
x = np.linspace(0, 2 * np.pi, 100)
y_sin = np.sin(x)
y_cos = np.cos(x)
```

```
# 1행 2열 구조의 서브플롯 생성
```

```
fig, axes = plt.subplots(1, 2,
figsize=(12, 4))
```

```
# 첫 번째 그래프
axes[0].plot(x, y_sin, color='blue')
axes[0].set_title("사인 함수 (Sine)")
axes[0].set_xlabel("x")
axes[0].set_ylabel("sin(x)")
```

```
# 두 번째 그래프
axes[1].plot(x, y_cos, color='orange')
axes[1].set_title("코사인 함수 (Cosine)")
axes[1].set_xlabel("x")
axes[1].set_ylabel("cos(x)")
```

```
plt.tight_layout() # 그래프 간 간격 자동 조절
plt.show()
```

결과물 저장 (Saving Figures)

생성한 그래프를 고해상도 이미지 파일로 저장합니다.

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

```
plt.plot([1, 2, 3], [4, 5, 6])
plt.title("파일 저장 예시")
```

```
# dpi: 해상도 설정, bbox_inches: 여백 최소화
```

```
plt.savefig("my_visualization.png",
dpi=300, bbox_inches='tight')
print("시각화 결과가 my_visualization.png
파일로 저장되었습니다.")
```