

# 데이터프로그래밍 기초 9일차

---

2026-1 DS Bootcamp

부산대학교  
데이터사이언스전문대학원  
석사과정 박민서

---

# CONTENTS

---

1 Matplotlib & Pyplot

2 Seaborn & Plotly

3 Practice (Data EDA)

## ▪ 맷플롯립 (Matplotlib)

- 데이터 시각화 + 2D 그래프를 그릴 때 사용하는 Python 라이브러리
- <https://matplotlib.org/stable/gallery/index.html>



## ▪ 파이플롯 (Pyplot)

- MATLAB 스타일의 함수 호출 방식으로 Python에서 그래프를 그릴 수 있게 하는 모듈

## ▪ 맷플롯립 (Matplotlib)

- 데이터 시각화 + 2D 그래프를 그릴 때 사용하는 Python 라이브러리
- <https://matplotlib.org/stable/gallery/index.html>
- `pip install matplotlib`

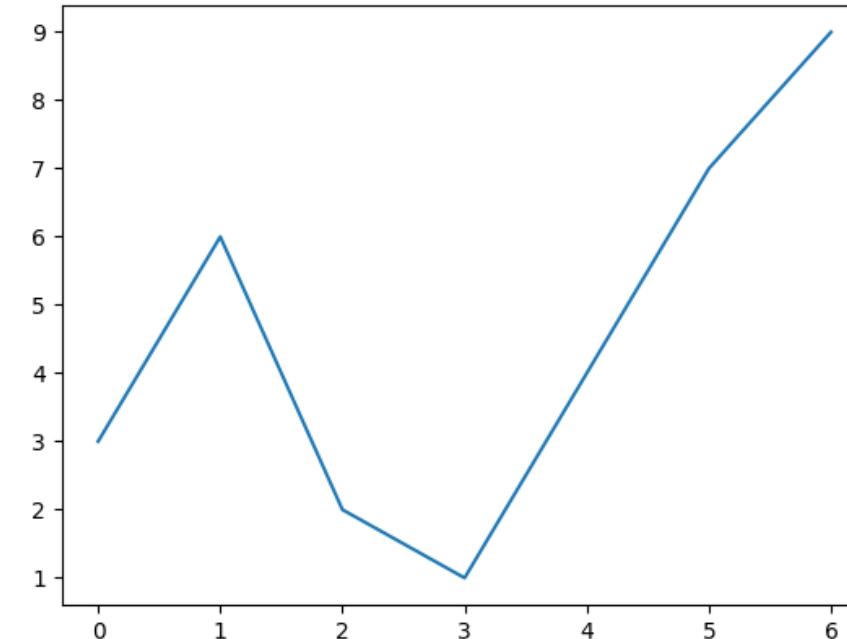


## ▪ 파이플롯 (Pyplot)

- MATLAB 스타일의 함수 호출 방식으로 Python에서 그래프를 그릴 수 있게 하는 모듈

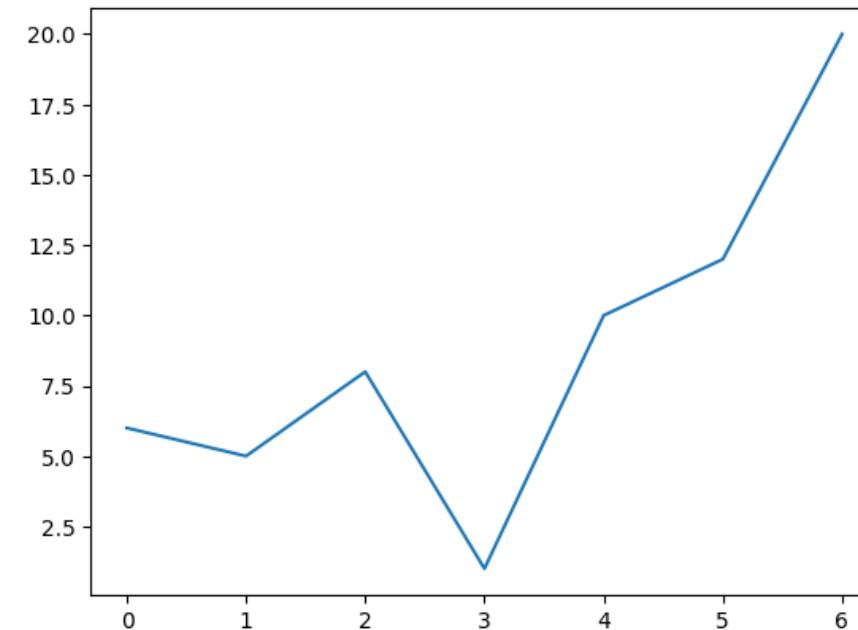
## ▪ 기초 시작화

- .plot(데이터): 데이터를 2차원 line plot으로 표현
- 데이터의 종류: (요소가 전부 숫자인) list, array (numpy), DataFrame의 열



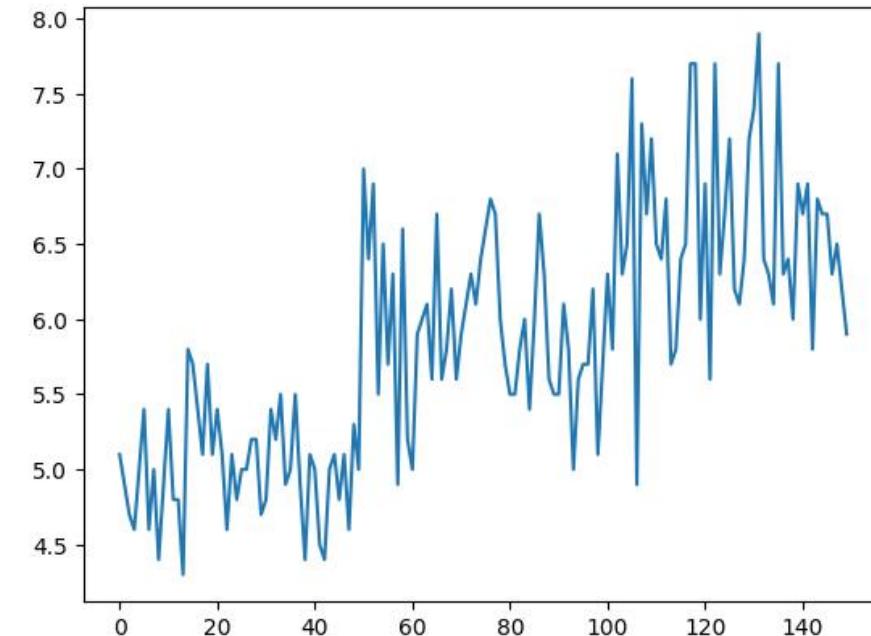
## ■ 기초 시작화

- .plot(데이터): 데이터를 2차원 line plot으로 표현
- 데이터의 종류: (요소가 전부 숫자인) list, array (numpy), DataFrame의 열



## ▪ 기초 시작화

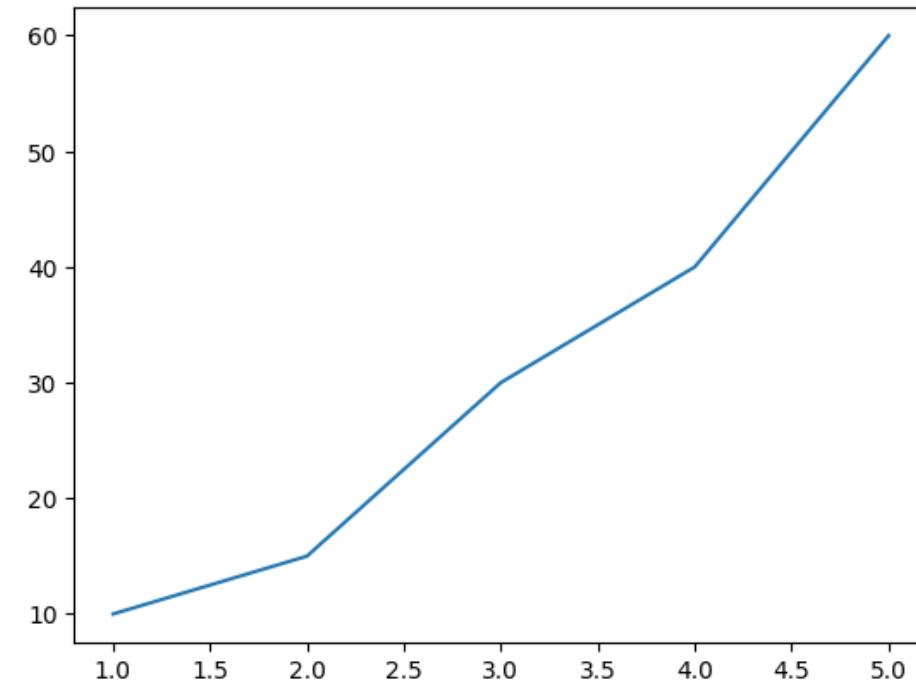
- .plot(데이터): 데이터를 2차원 line plot으로 표현
- 데이터의 종류: (요소가 전부 숫자인) list, array (numpy), DataFrame의 열



# Matplotlib & Pyplot

## ▪ 기초 시각화

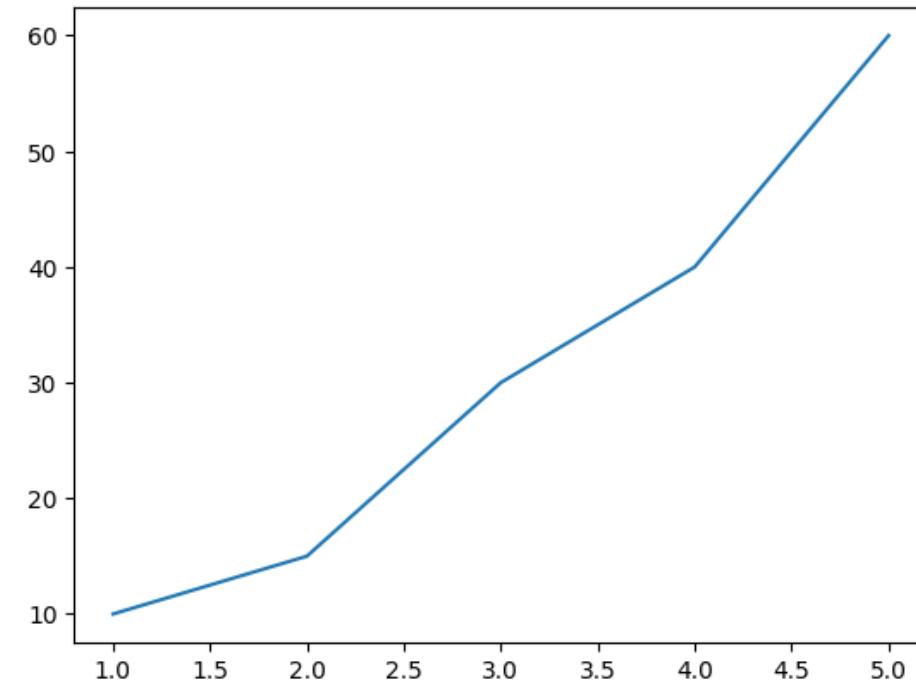
- `.plot(x값, y값)`: x, y값을 동시에 plot



# Matplotlib & Pyplot

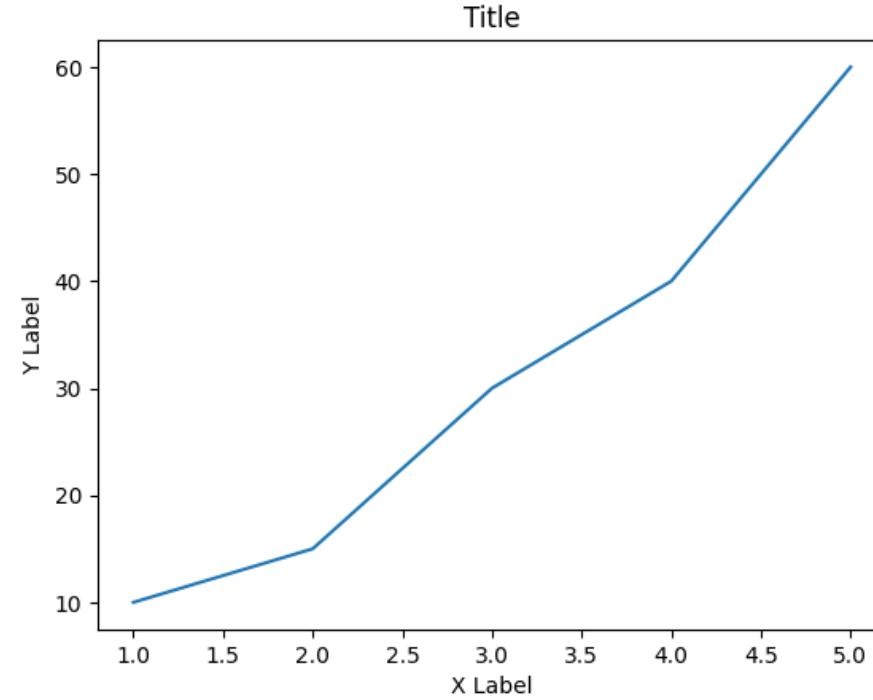
## ▪ 기초 시각화

- `.plot(x값, y값)`: x, y값을 동시에 plot



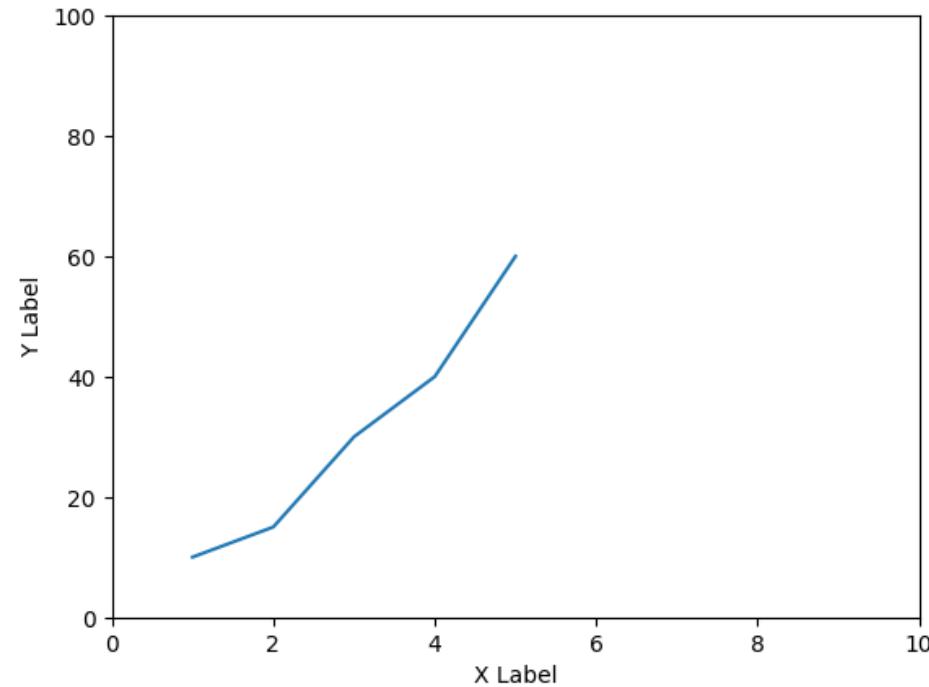
## ▪ 기초 시작화

- .xlabel("라벨 입력"), .ylabel("라벨 입력"): x축, y축에 쓸 축 이름 생성
- .title("제목 입력"): 그래프 제목 생성



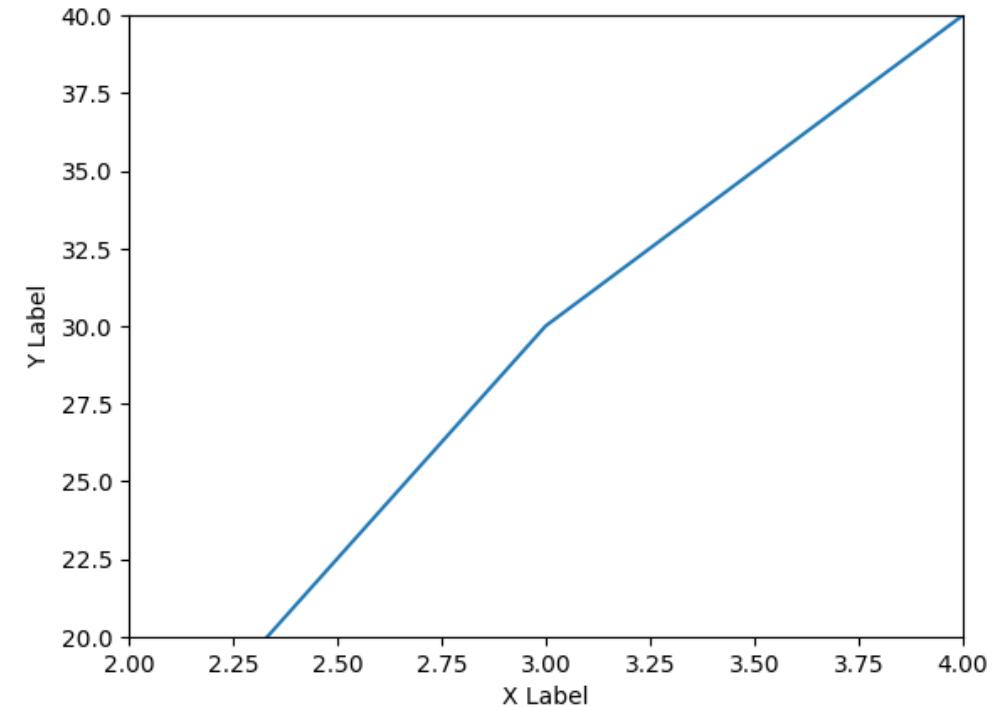
## ▪ 기초 시각화

- `.xlim([시작, 끝])`, `.ylim([시작, 끝])`: x값과 y값의 범위를 제한



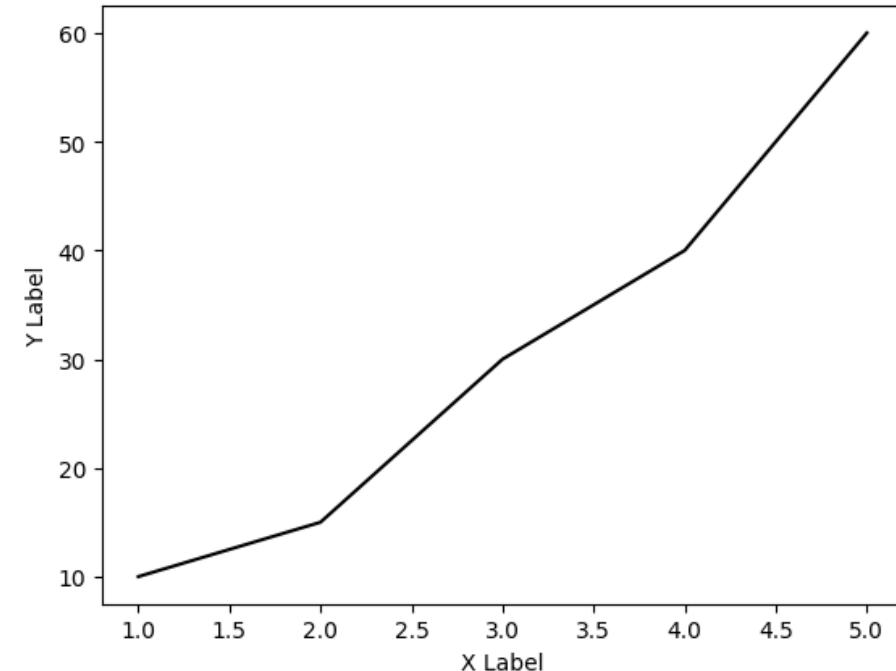
## ▪ 기초 시각화

- `.xlim([시작, 끝])`, `.ylim([시작, 끝])`: x값과 y값의 범위를 제한



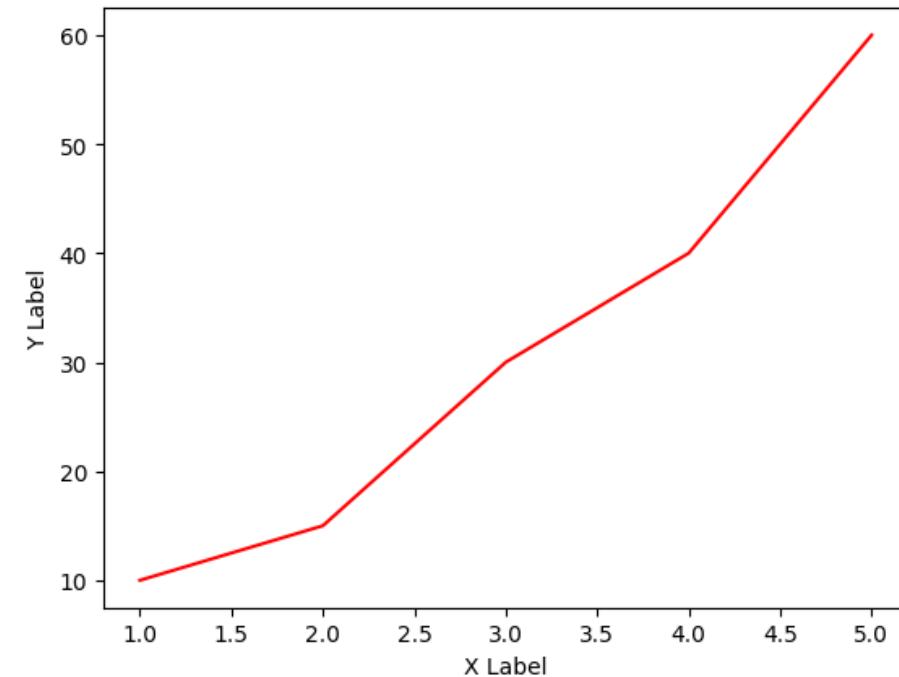
## ■ 기초 시각화

- .plt(color = “색깔”): 그래프의 색을 지정
- ‘b’: 파랑, ‘g’: 초록, ‘r’: 빨강, ‘c’: 청록 (cyan),
- ‘m’: 마젠타 (magenta), ‘y’: 노랑, ‘k’:검정(?), ‘w’:흰색, <https://wikidocs.net/92085>



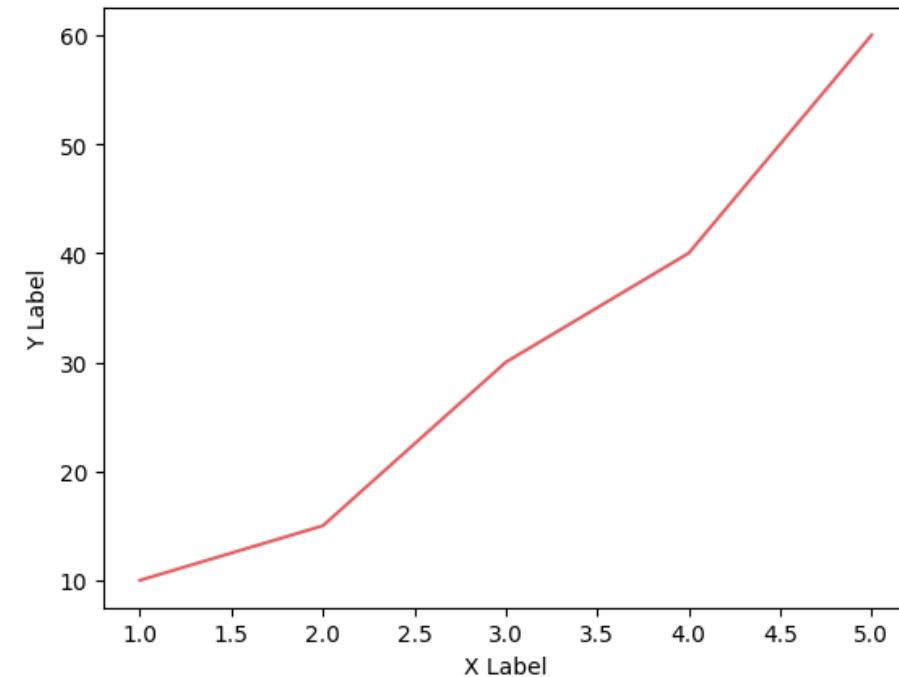
## ■ 기초 시각화

- .plt(color = “색깔”): 그래프의 색을 지정
- ‘b’: 파랑, ‘g’: 초록, ‘r’: 빨강, ‘c’: 청록 (cyan),
- ‘m’: 마젠타 (magenta), ‘y’: 노랑, ‘k’:검정(?), ‘w’:흰색, <https://wikidocs.net/92085>



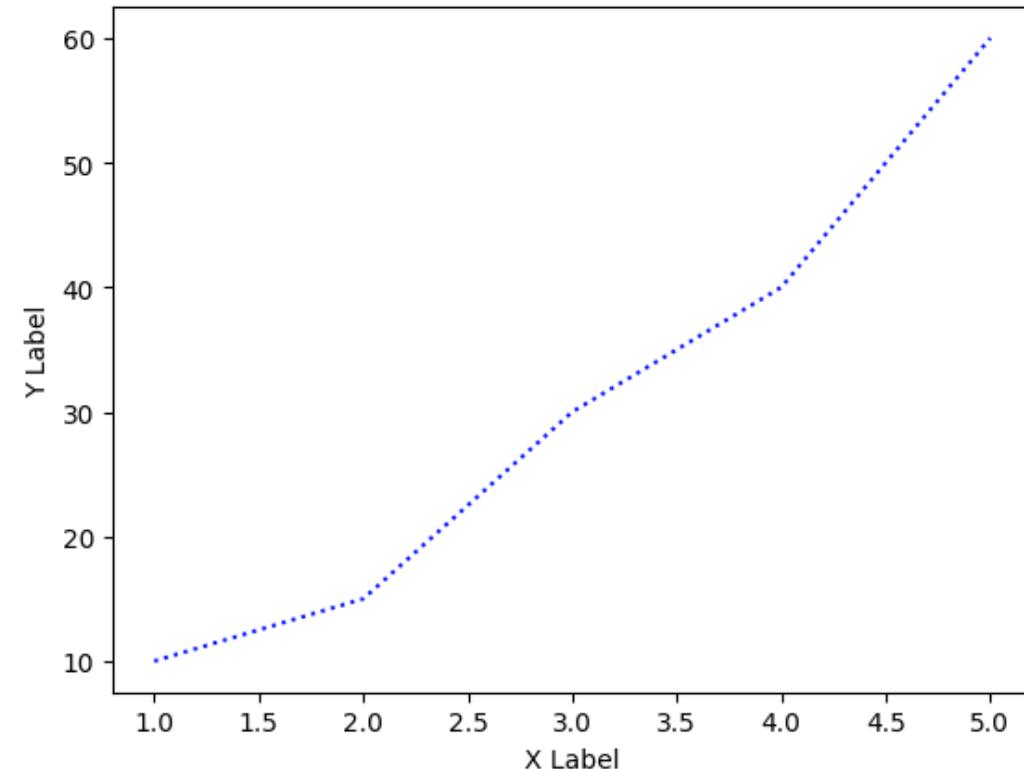
## ■ 기초 시각화

- .plt(color = “색깔”): 그래프의 색을 지정
- ‘b’: 파랑, ‘g’: 초록, ‘r’: 빨강, ‘c’: 청록 (cyan),
- ‘m’: 마젠타 (magenta), ‘y’: 노랑, ‘k’:검정(?), ‘w’:흰색, <https://wikidocs.net/92085>



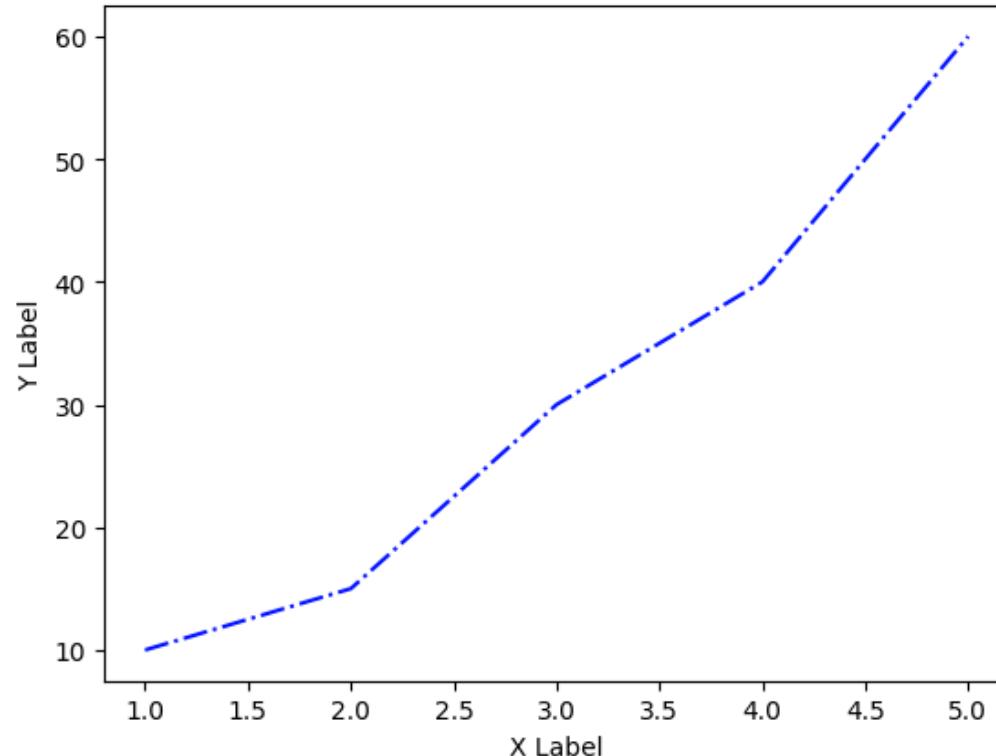
## ■ 기초 시각화

- `.plt(style = “스타일”)`: 선 그래프의 스타일을 지정
- “-”: solid (실선, 기본값), “--”: dashed (파선), “-.”: dashdot (일점쇄선), “:”: dotted(점선)



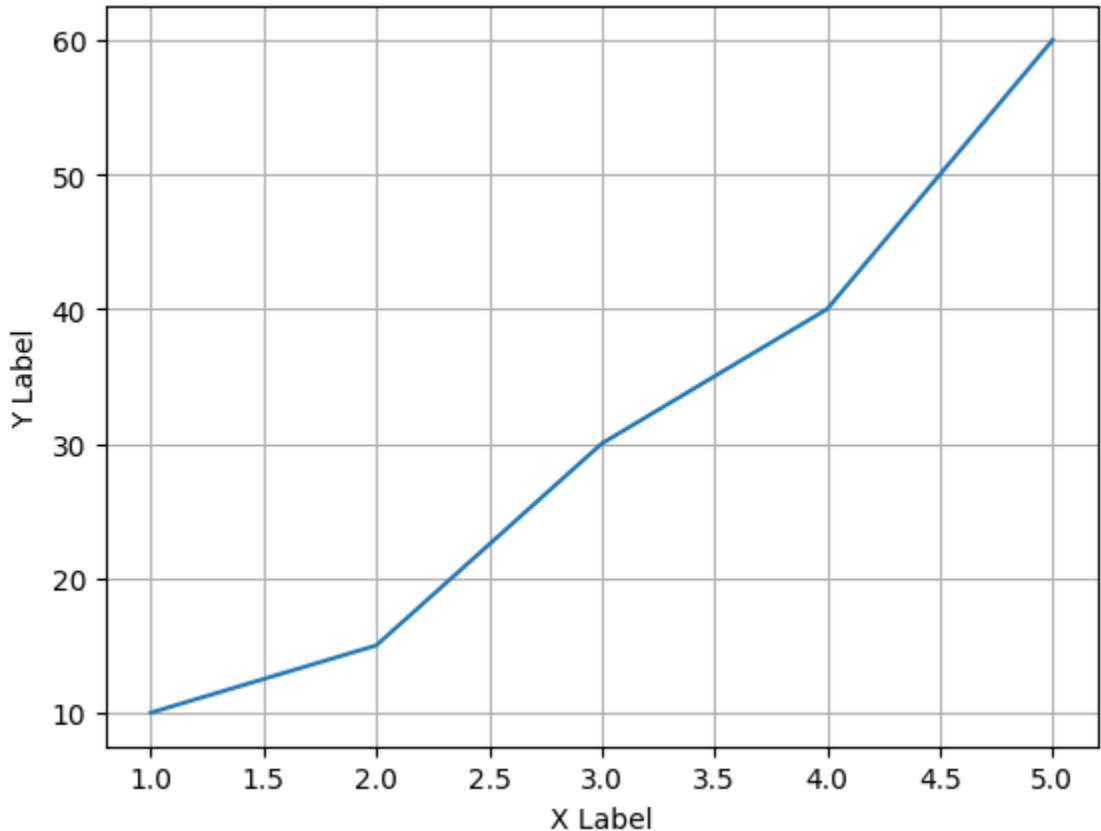
## ■ 기초 시각화

- `.plt(style = “스타일”)`: 선 그래프의 스타일을 지정
- “-”: solid (실선, 기본값), “--”: dashed (파선), “-.”: dashdot (일점쇄선), “:”: dotted(점선)



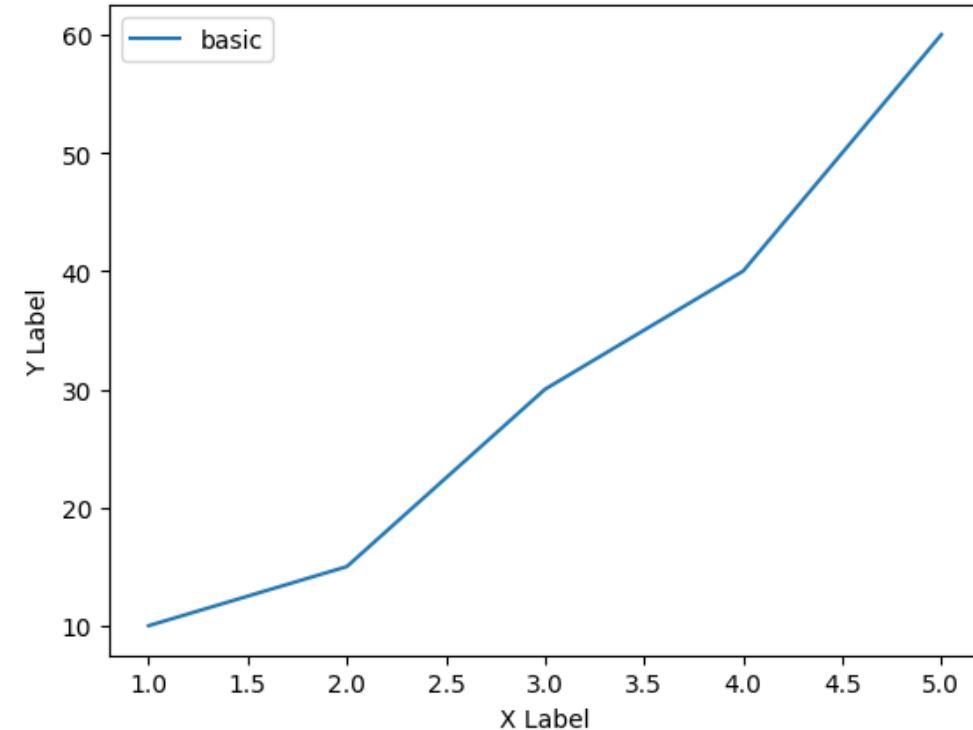
## ■ 기초 시작화

- .grid( ): 그래프의 격자 그리는 설정 (기본값 True)



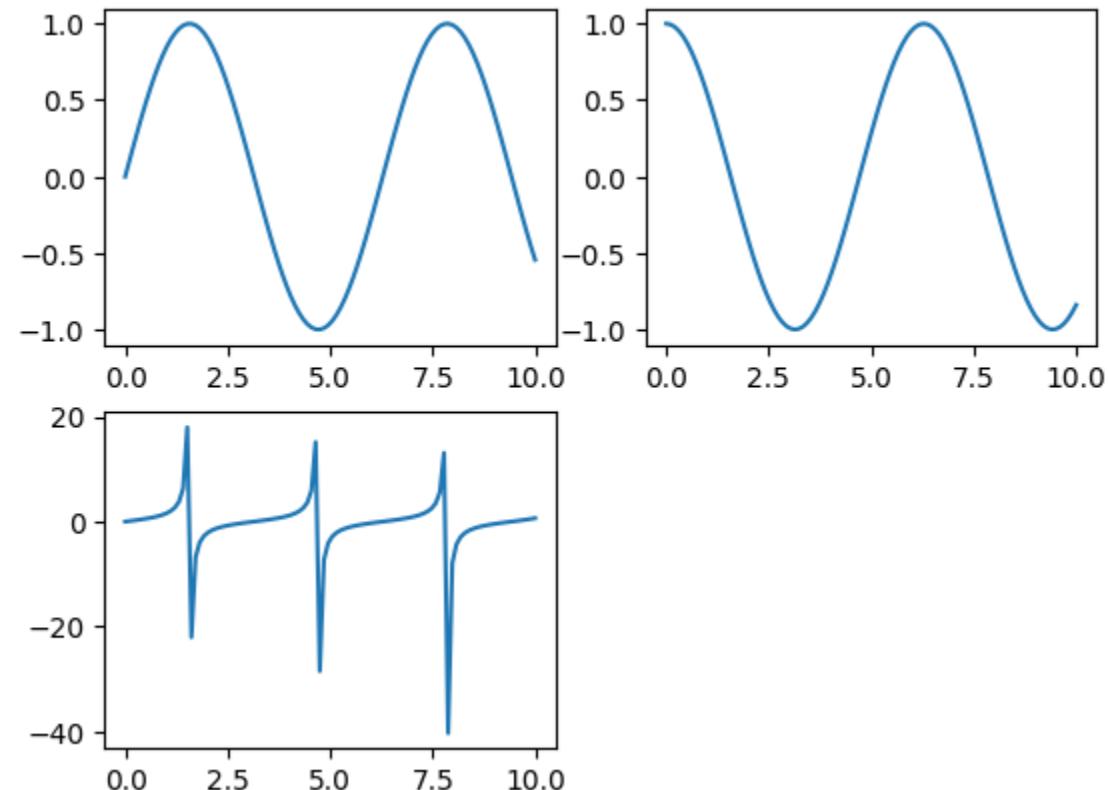
## ▪ 기초 시작화

- `.legend( )`: 그래프의 격자 그리는 설정
- `loc = [옵션]`: 범례 위치 설정 (기본은 “best”)



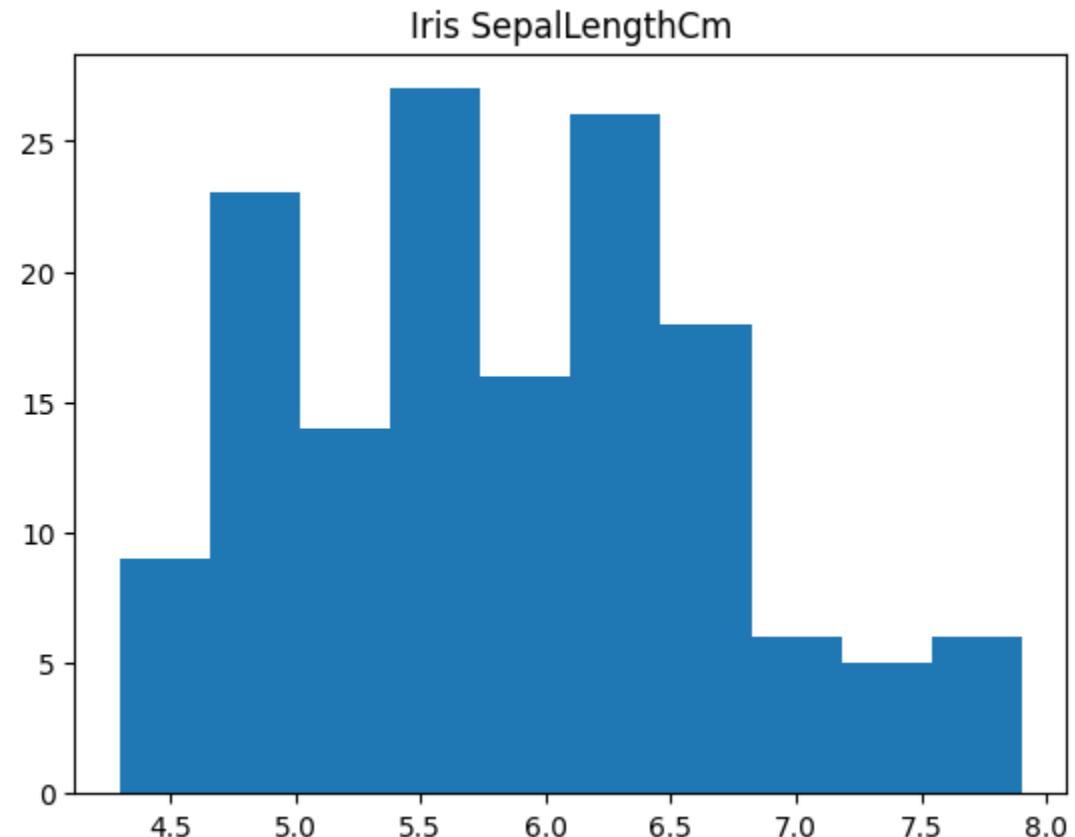
## ▪ 기초 시각화

- `.subplot(행 개수, 열 개수, index[1~])`: 여러 개의 그래프 동시에 그리는 설정



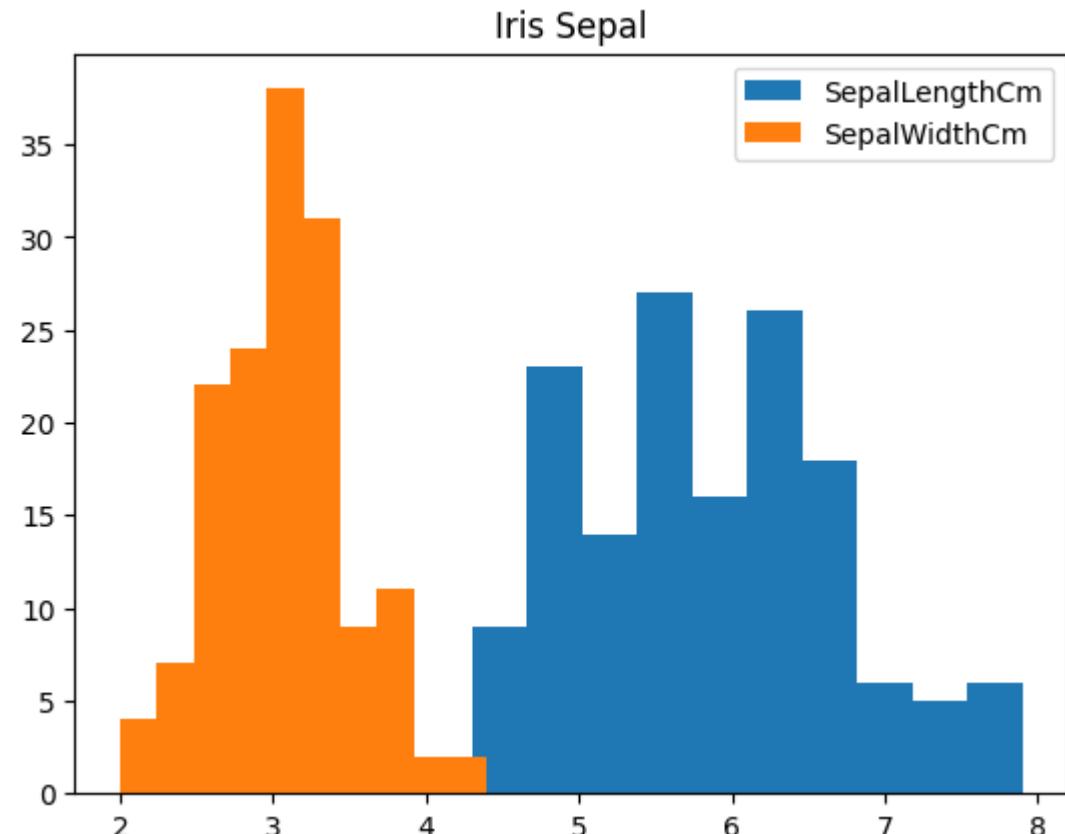
## ▪ 기초 시각화

- .hist(데이터): 히스토그램 생성



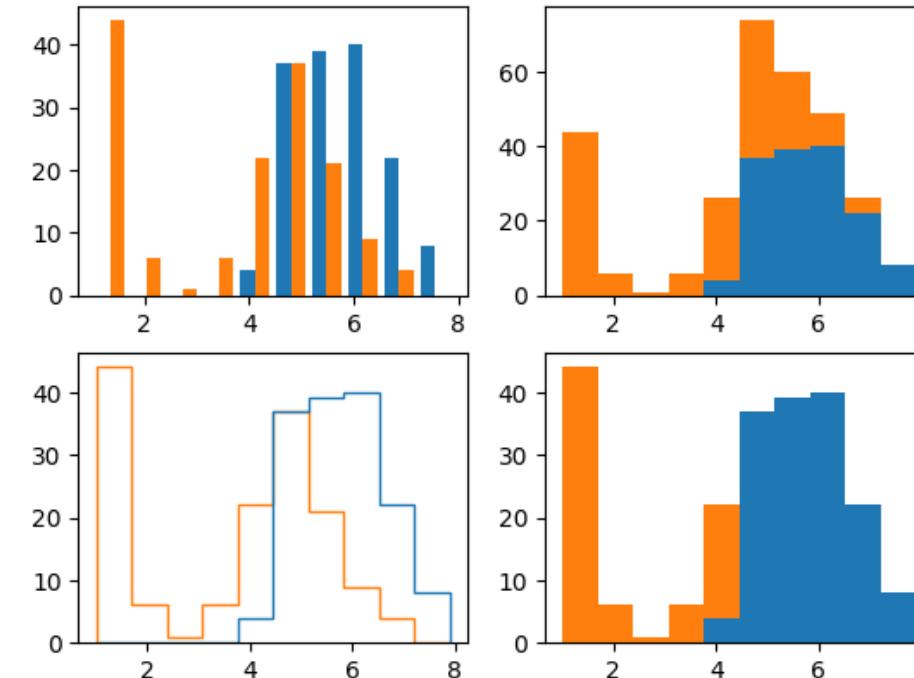
## ▪ 기초 시각화

- .hist(데이터): 히스토그램 생성



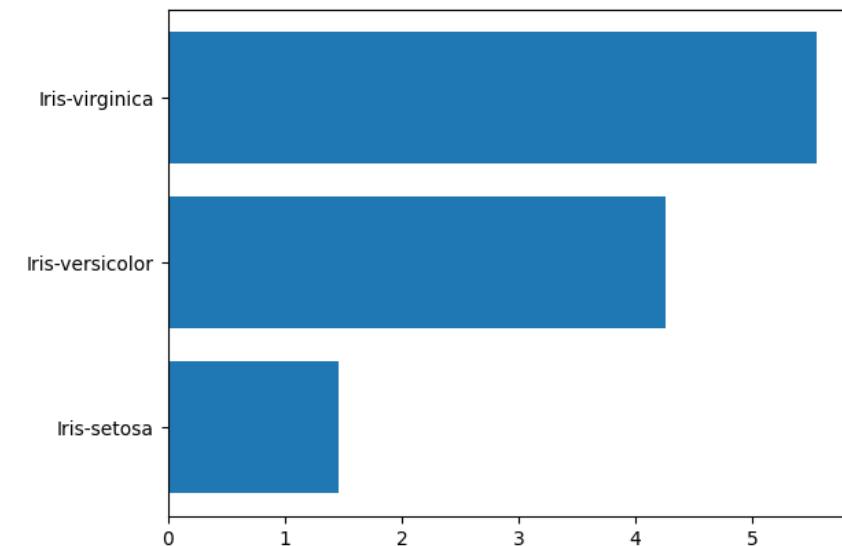
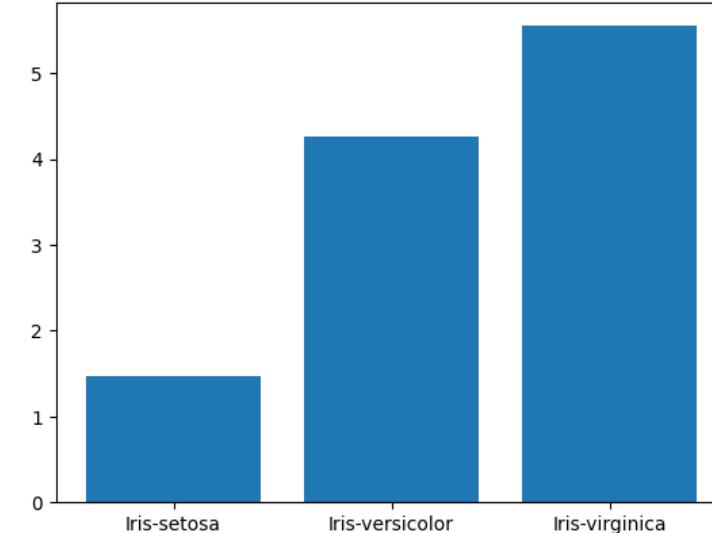
## ■ 기초 시각화

- .hist(데이터): 히스토그램 생성
  - histtype 옵션 지정:
    - “bar” – 막대그래프, “barstacked” – 막대 + 위로 쌓기, “step” – 계단식 (윤곽선만), “stepfilled” – 계단식 (색)



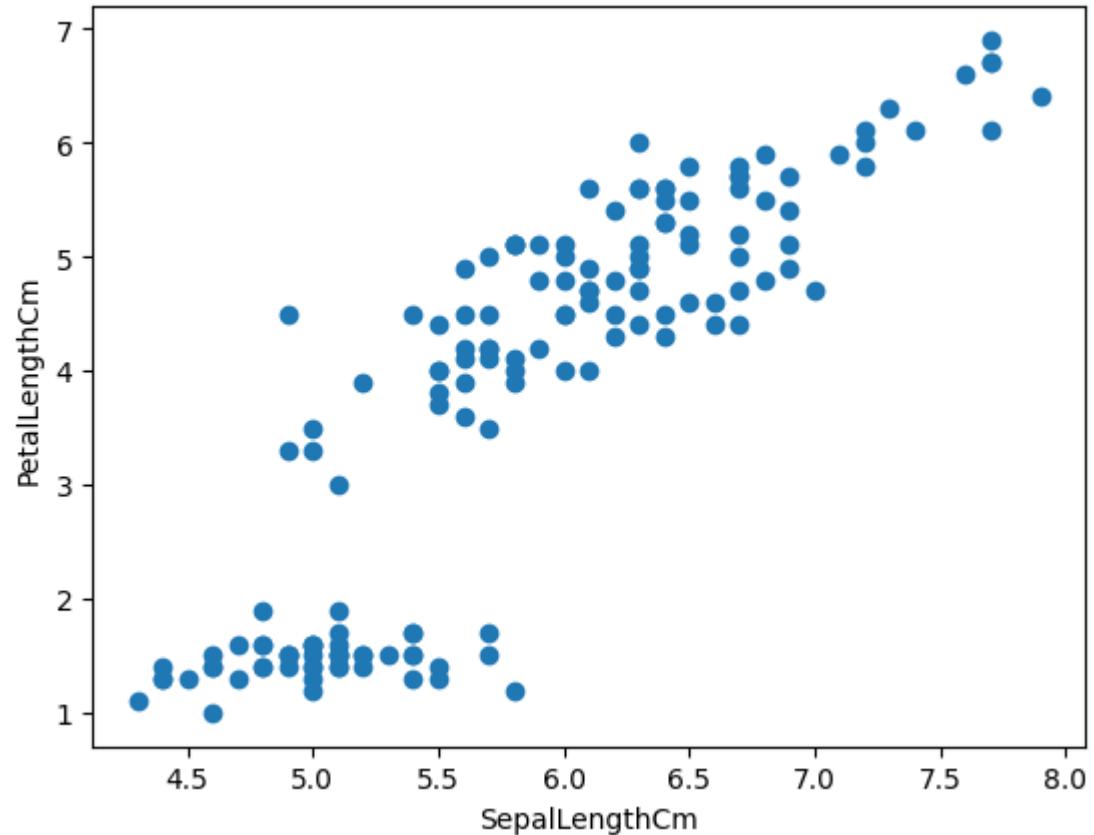
## ■ 기초 시각화

- `.bar(x, y)`: 막대그래프 생성
- `.barh(x, y)`: 수평 방향의 막대그래프 생성



## ▪ 기초 시각화

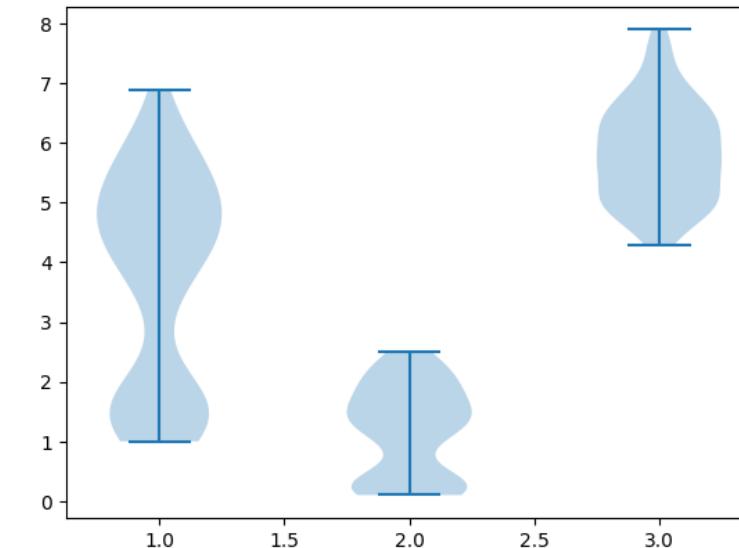
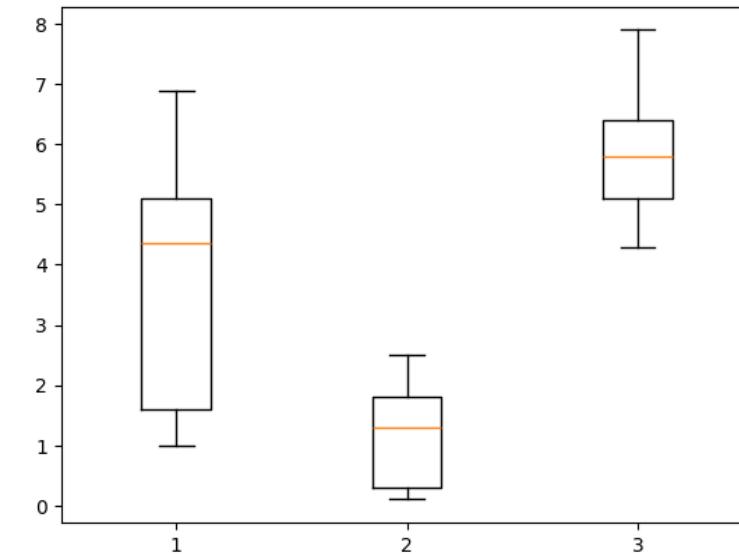
- `.scatter(x, y)`: 산점도 생성



# Matplotlib & Pyplot

## ▪ 기초 시각화

- `.boxplot(데이터)`: 박스플롯 생성
- `.violinplot(데이터)`: 바이올린 플롯 생성



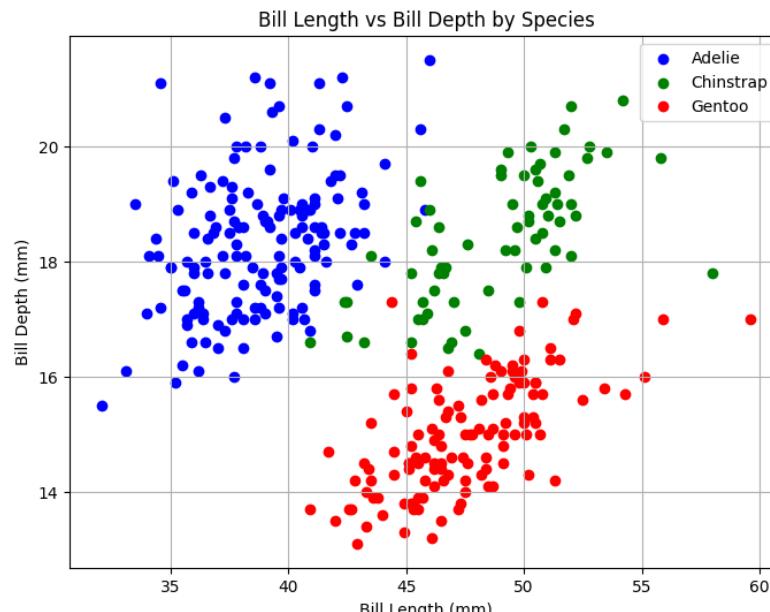
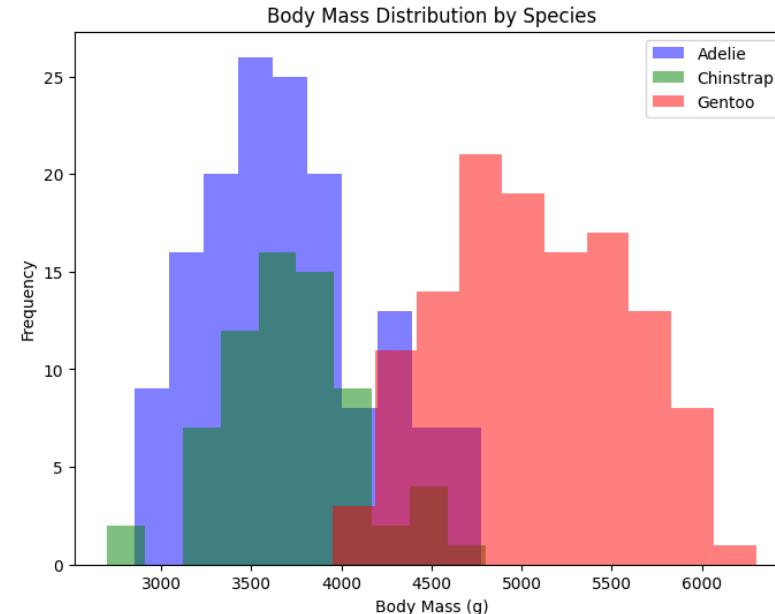
---

# Q&A

---

## ■ 실습 1

- “penguins\_size.csv” 파일에 대해 다음 작업을 수행하세요.
  - species별 ‘body\_mass\_g’를 한 그래프에 겹쳐서 히스토그램으로 그리시오.
    - alpha (투명도) = 0.5로, 서로 다른 색상으로 설정하세요.
    - 범례, 제목, x, y 라벨 설정 (범례의 label은 각 species 이름으로 설정)
  - ‘culmen\_length\_mm’와 ‘culmen\_depth\_mm’에 대한 산점도를 그리시오.
    - 각 species 별로 색을 다르게 해서 표현
    - 범례, 제목, x, y 라벨 설정 (범례의 label은 각 species 이름으로 설정)

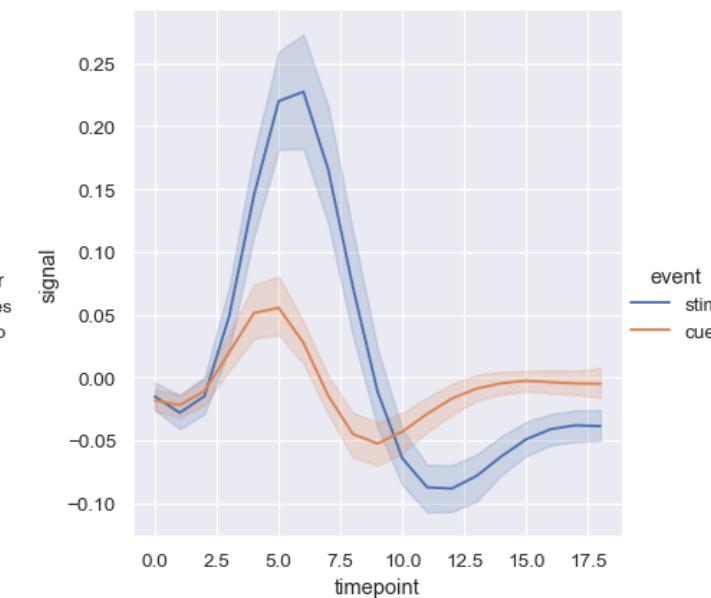
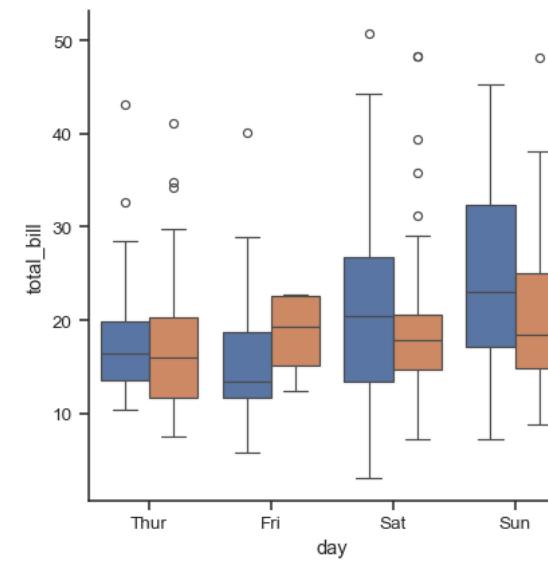
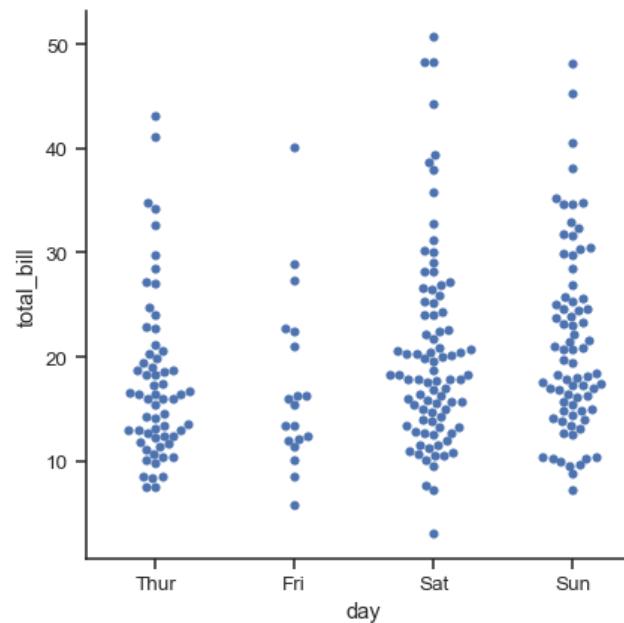


## ▪ 시본 (Seaborn)

- Matplotlib 기반의 데이터 시각화 라이브러리
- pyplot보다 좀 더 고차원의 인터페이스 + 간결한 코드
- Python 3.8 이상 지원
- <https://seaborn.pydata.org/api.html>

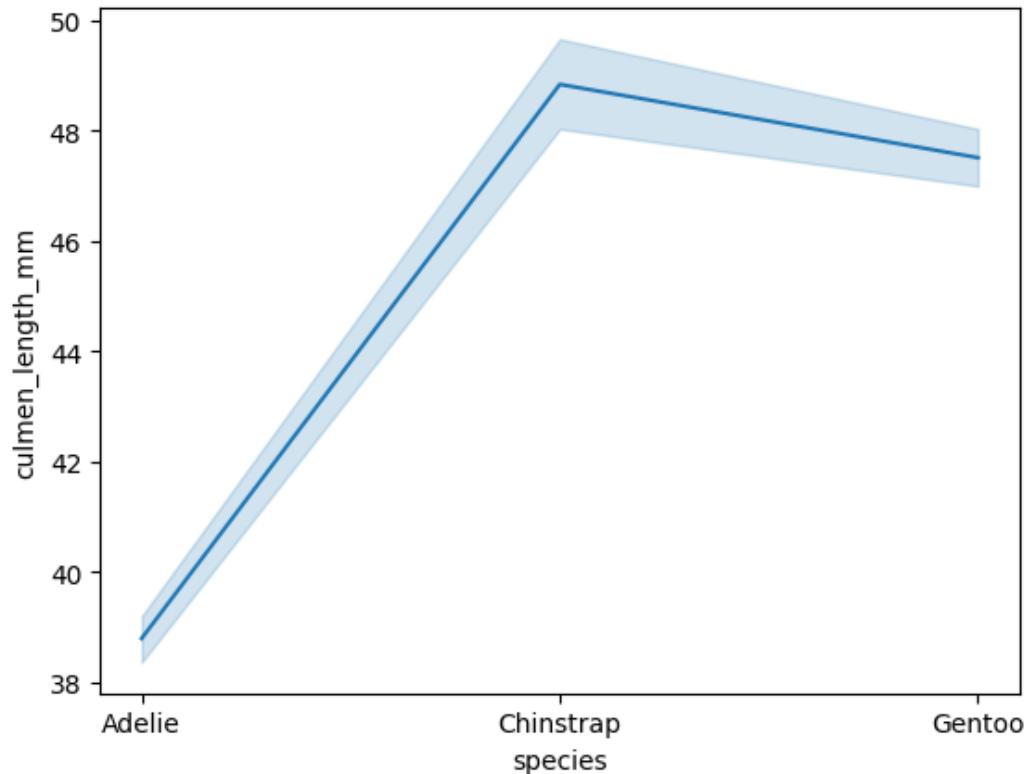


seaborn



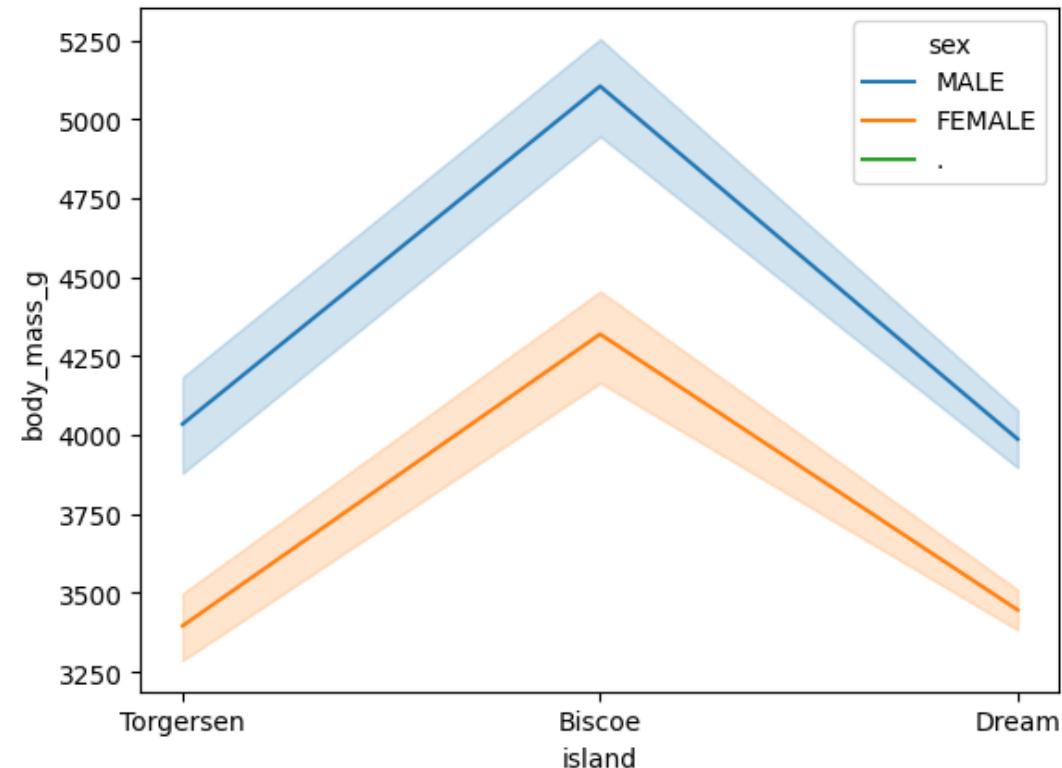
## ▪ Seaborn 시각화

- `.lineplot(data, x, y)`: 선 그래프 생성
- 같은 x값들에 대한 y값이 집계되어 평균 계산 + 95% 신뢰구간 표시



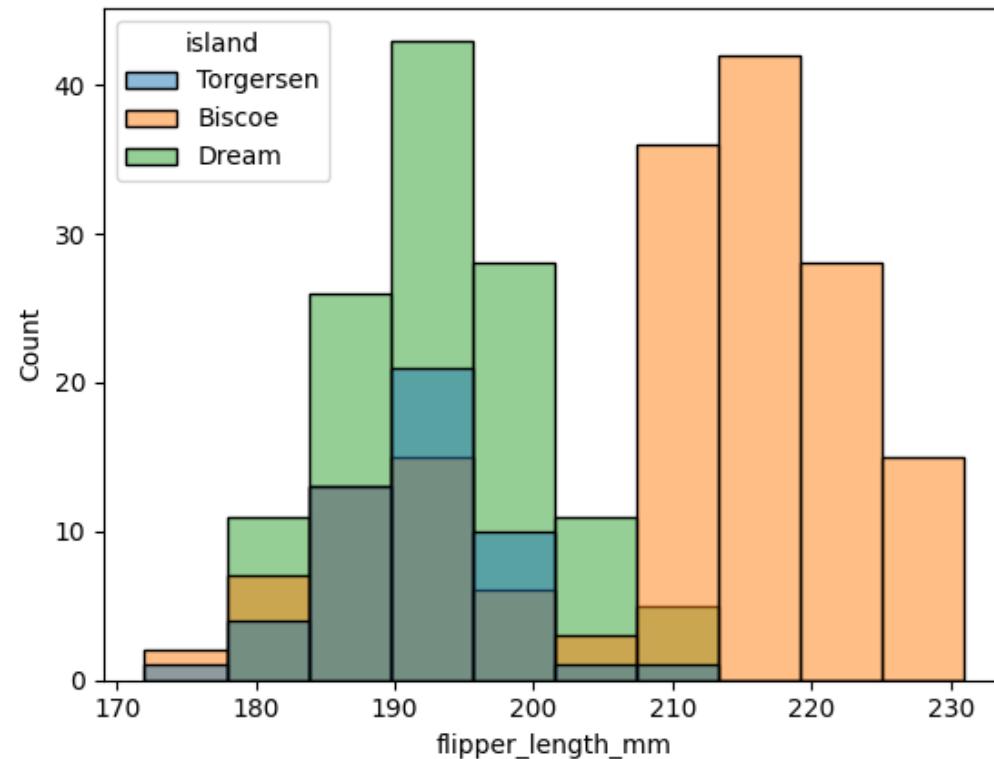
## ▪ Seaborn 시각화

- `.lineplot(data, x, y)`: 선 그래프 생성
- `hue = “범주형 변수”`: 범주에 따른 색 자동 지정



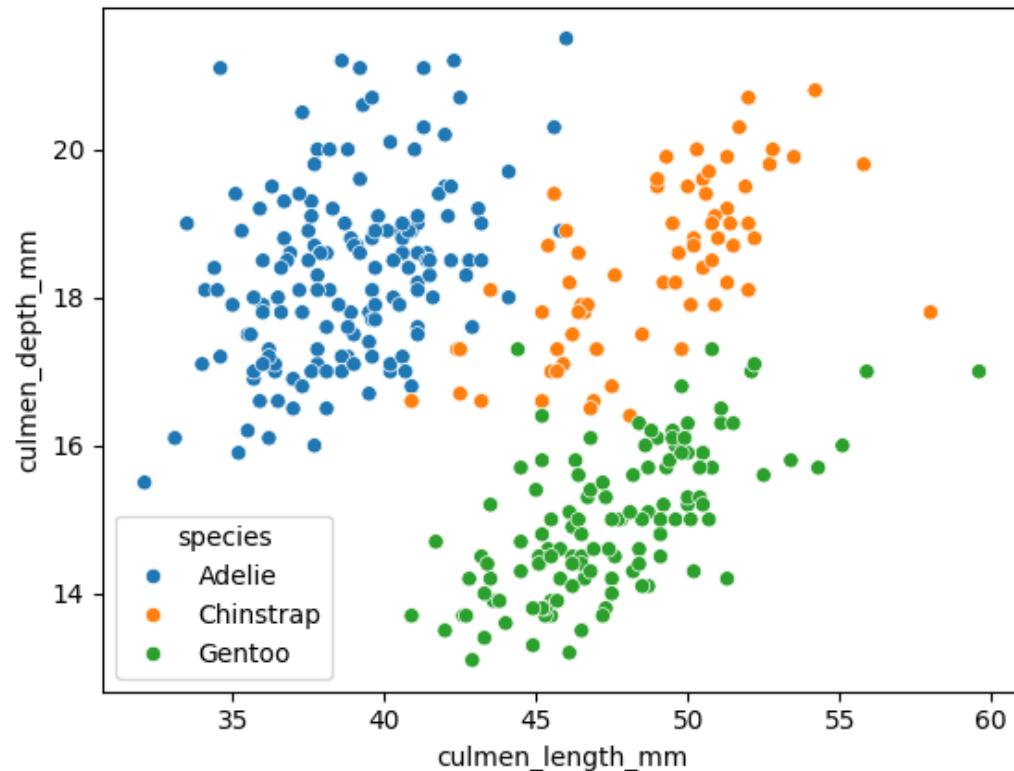
## ▪ Seaborn 시각화

- `.histplot(data, x, [bins])`: 히스토그램 생성
- `hue = “범주형 변수”`: 범주에 따른 색 자동 지정



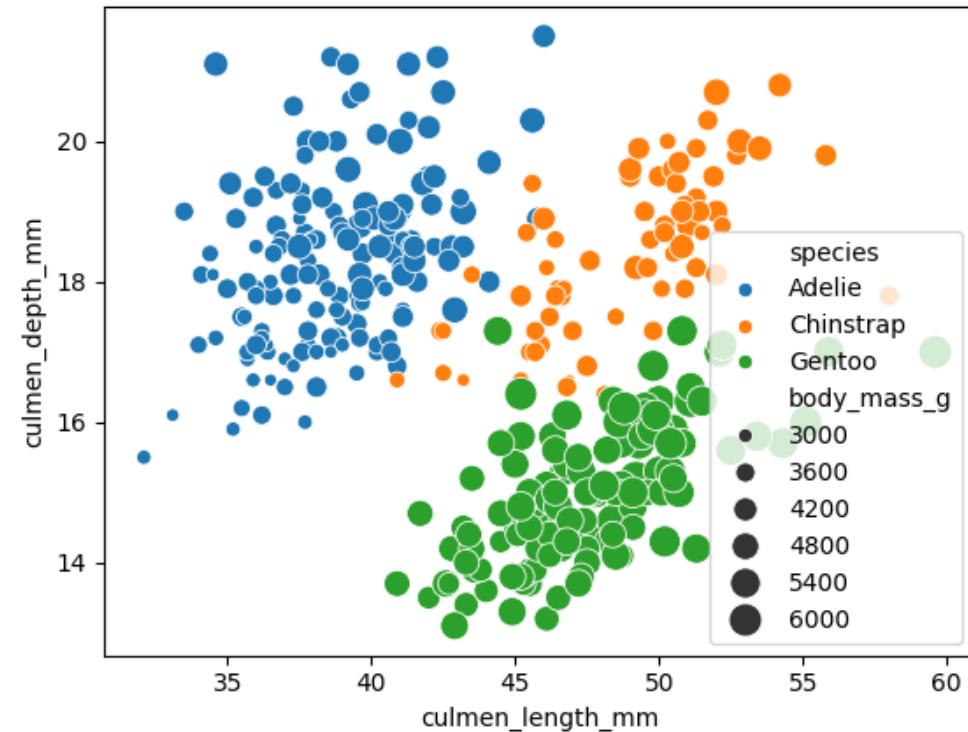
## ▪ Seaborn 시각화

- `.scatter(data, x, y)`: 산점도 그래프 생성
- `hue = “범주형 변수”`: 범주에 따른 색 자동 지정



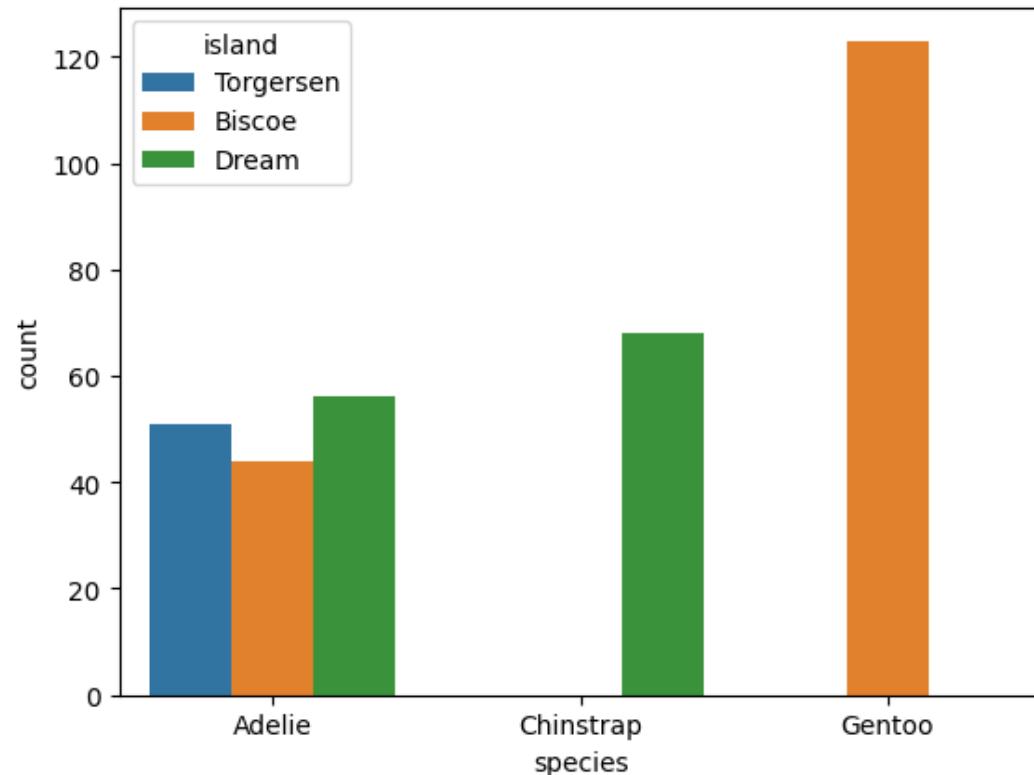
## ▪ Seaborn 시각화

- `.scatter(data, x, y)`: 산점도 그래프 생성
- `hue = “범주형 변수”`: 범주에 따른 색 자동 지정
- `size = “수치형 변수”`: 수치에 따른 사이즈 조정, `sizes = (최소, 최대)`



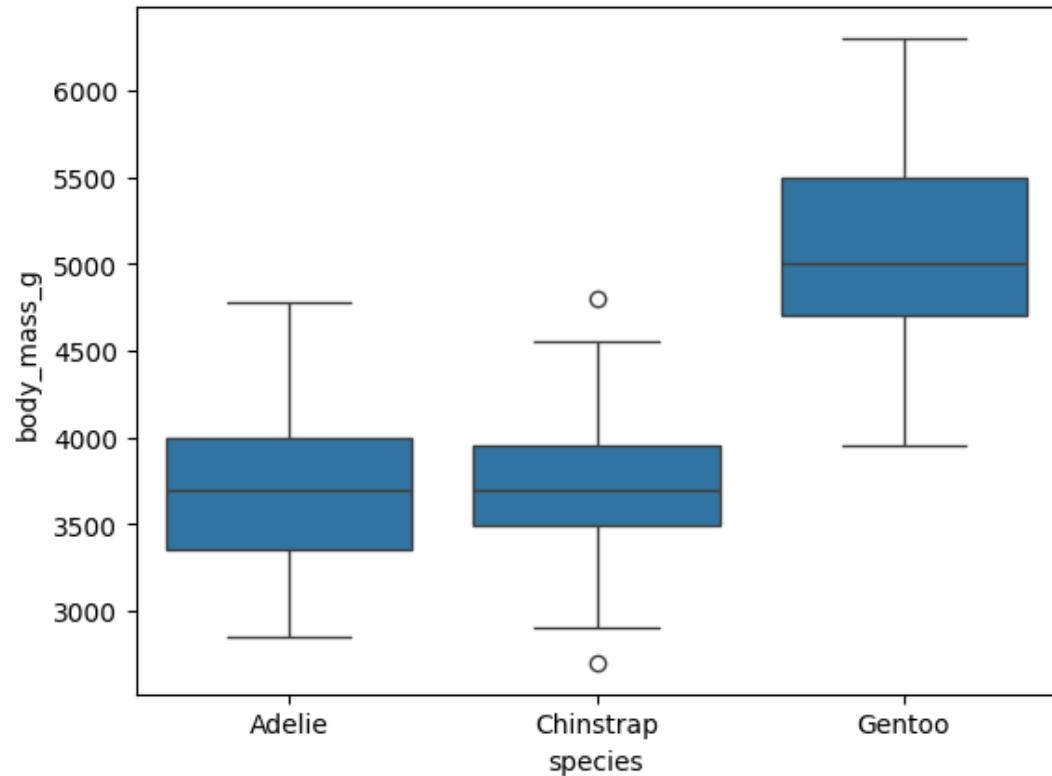
## ▪ Seaborn 시각화

- `.countplot(data, x)`: 범주형 자료의 개수 확인
- `hue = “범주형 변수”`: 범주에 따른 색 자동 지정



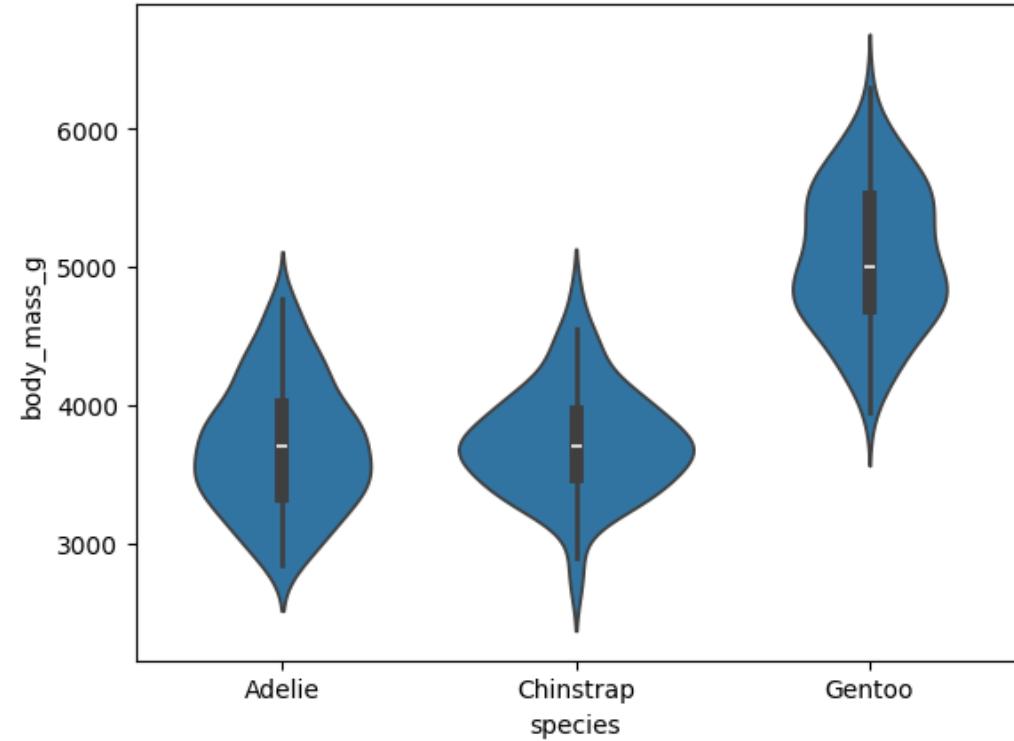
## ▪ Seaborn 시각화

- `.boxplot(data, x, y)`: 박스플롯 생성
- `hue = “범주형 변수”`: 범주에 따른 색 자동 지정



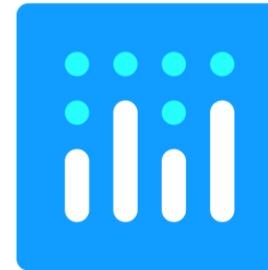
## ▪ Seaborn 시각화

- `.violinplot(data, x, y)`: 바이올린플롯 생성
- `hue = “범주형 변수”`: 범주에 따른 색 자동 지정

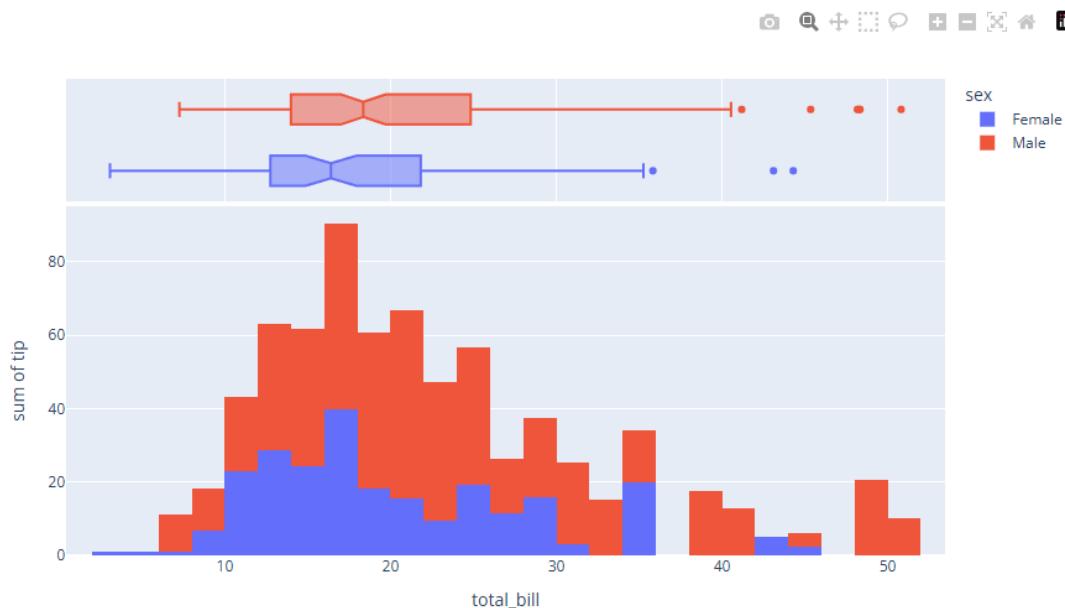


## ▪ 플롯트리 (Plotly) 소개

- Matplotlib / Seaborn과 다르게 Interactive한 시각화 가능
- Matplotlib 대비 간결한 코드
- 연습 코드: <https://wikidocs.net/185024>
- <https://plotly.com/python/>



plotly



---

# Q&A

---