



# 시각화

최석재

lingua@naver.com

# Matplotlib 소개

*Introducing the Matplotlib*

- Matplotlib은 데이터의 시각화를 위한 패키지이다
- 파이썬의 여러 시각화 패키지 중 가장 잘 알려진 기본 패키지이다
- Matplotlib 패키지는 아나콘다를 설치했다면 기본적으로 포함되어 있다
- 파이썬을 이용하는 모든 플랫폼에서 잘 동작하는 것이 큰 장점이다

# Matplotlib 불러오기

- 별명으로는 보통 plt 를 사용한다

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

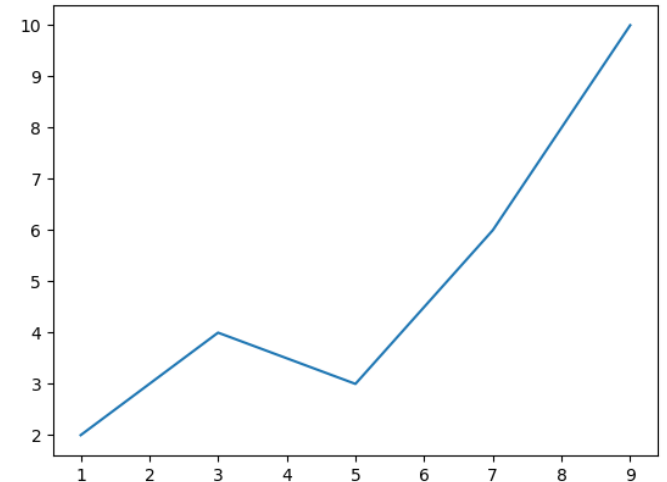
```
x = [1, 3, 5, 7, 9]
```

```
y = [2, 4, 3, 6, 10]
```

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

```
plt.show()
```

*# 마지막에 한 번만 사용한다*



# 스타일

- 간단히 스타일을 변경할 수 있다
- <https://python-graph-gallery.com/199-matplotlib-style-sheets/>

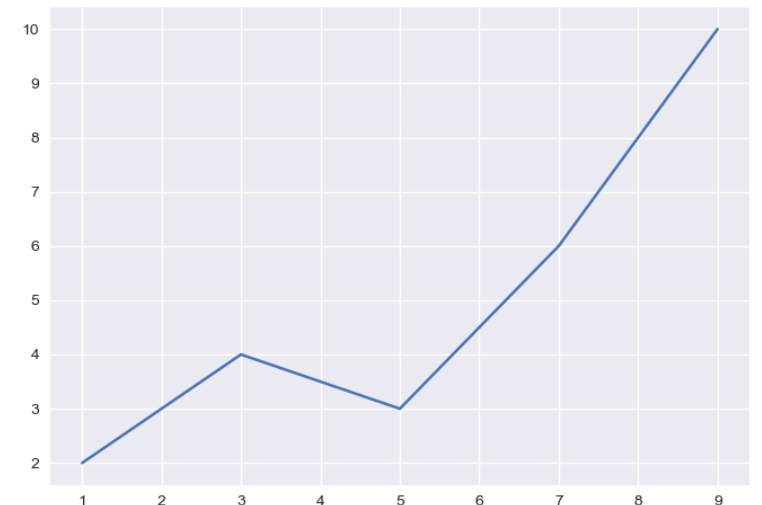
"plt.style.use" 로 검색

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

```
plt.style.use("seaborn")
```

```
x = [1, 3, 5, 7, 9]  
y = [2, 4, 3, 6, 10]
```

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



# 그림 저장

- 그림을 저장하는 방법을 본다
- 캡처 도구를 사용하는 것이 더 편리할 수 있다 ;)

```
import os
```

```
print(os.getcwd())  
os.chdir(os.getcwd()+"/image/")
```

*# 현재 경로를 출력해본다*

*# 이미 현재 경로 아래에 image 라는 폴더가 존재해야 한다*

```
import matplotlib.pyplot as plt  
plt.style.use("seaborn")
```

```
x = [1, 3, 5, 7, 9]  
y = [2, 4, 3, 6, 10]
```

```
fig = plt.figure()  
plt.plot(x, y)  
fig.savefig("mypic.png")
```

*# figure 객체를 만든다*

*# 그림을 그리고,*

*# figure 객체를 이용해 현재의 그림을 저장한다 (jpg도 가능)*

# 그림 겹치기

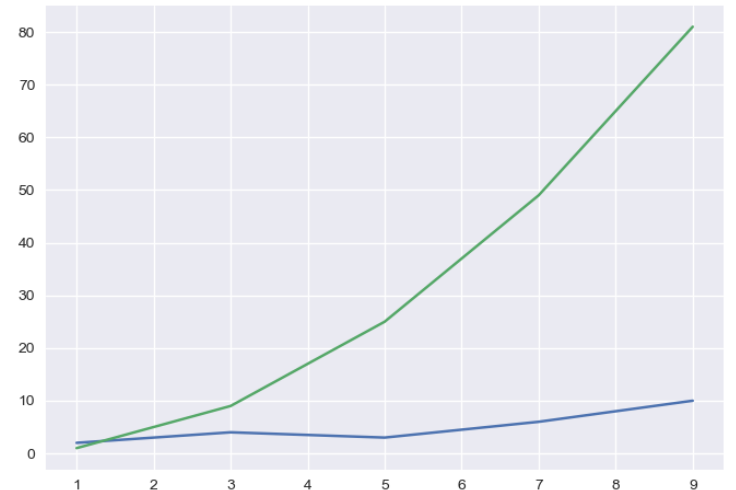
- 두 개 이상의 그림을 한 화면에 그릴 수 있다

```
import matplotlib.pyplot as plt
plt.style.use("seaborn")
```

```
x = [1, 3, 5, 7, 9]
y = [2, 4, 3, 6, 10]
y2 = [1, 9, 25, 49, 81]
```

```
plt.plot(x, y)
plt.plot(x, y2)
```

```
plt.show()
```



# 선 색상 수정하기

```
import matplotlib.pyplot as plt
plt.style.use("seaborn")
```

```
def mycal(num):
    num2 = num ** 2
    return num2
```

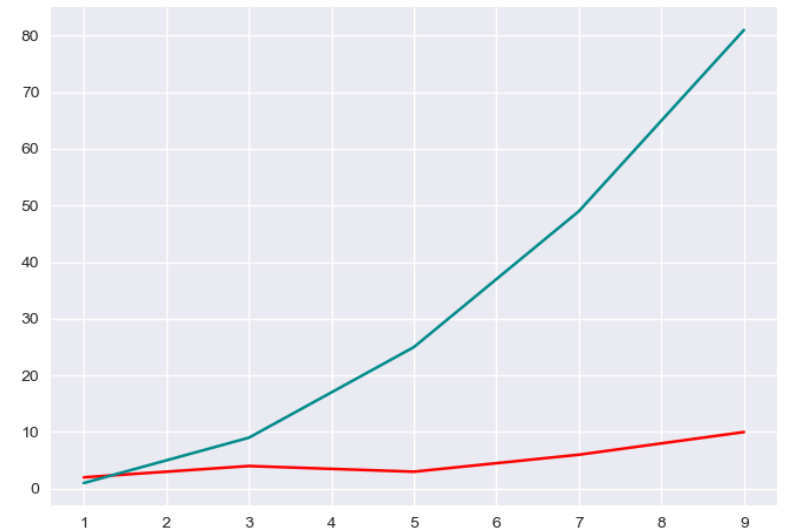
- color 속성을 이용하여 선의 색을 바꿀 수 있다
- [https://matplotlib.org/3.1.0/gallery/color/named\\_colors.html](https://matplotlib.org/3.1.0/gallery/color/named_colors.html)  
(matplotlib color)

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

```
for x_each in x:                # 함수로 y2 생성
    y2.append(mycal(x_each))
```

```
plt.plot(x, y, color='red')
plt.plot(x, y2, color='darkcyan')
```

```
plt.show()
```



# 선 스타일 수정하기

```
import matplotlib.pyplot as plt
plt.style.use("seaborn")
```

```
def mycal(num, i):          # 복수의 매개변수를 갖는 함수
    num2 = num ** i
    return num2
```

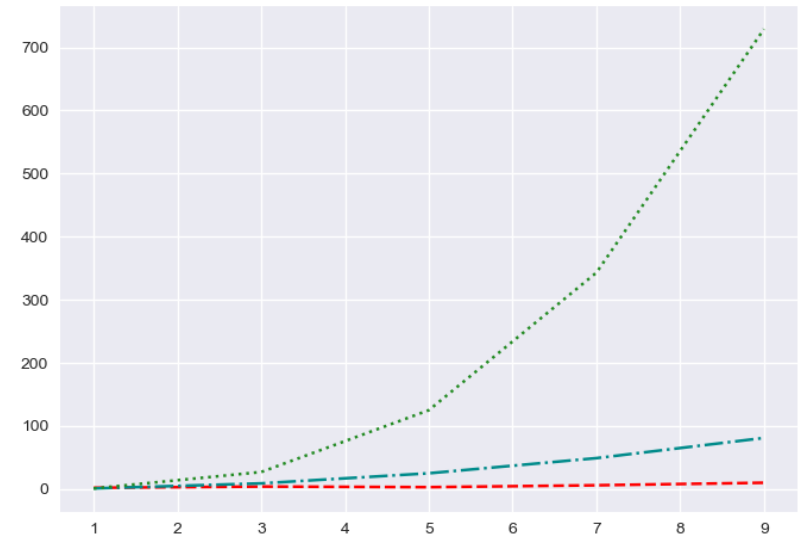
```
x = [1, 3, 5, 7, 9]
y = [2, 4, 3, 6, 10]
y2 = [] ; y3 = []
```

*# 다음 줄에 쓸 것을 세미콜론으로 엮을 수 있다*

```
for x_each in x:
    y2.append(mycal(x_each, 2))
    y3.append(mycal(x_each, 3))
```

```
plt.plot(x, y, color='red', linestyle='dashed')
plt.plot(x, y2, color='darkcyan', linestyle='dashdot')
plt.plot(x, y3, color='forestgreen', linestyle='dotted')
```

```
plt.show()
```





# 레이블 붙이기

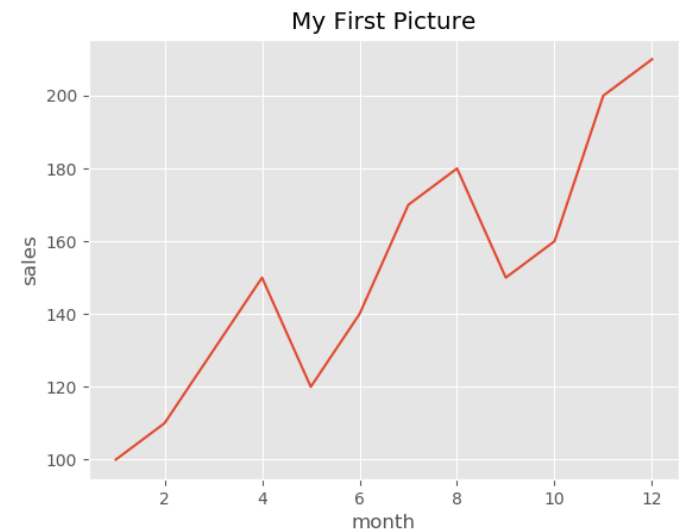
```
import matplotlib.pyplot as plt  
plt.style.use("ggplot")
```

```
x = range(1, 13)                                # range(include:exclude)  
y = [100, 110, 130, 150, 120, 140, 170, 180, 150, 160, 200, 210]
```

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

```
plt.title("My First Picture")  
plt.xlabel("month")  
plt.ylabel("sales")
```

```
plt.show()
```



# 한글 폰트 사용하기

- 컴퓨터가 가지고 있는 한글 폰트 경로를 알려주어 사용하게 한다
- C:/Windows/Fonts 에 있는 폰트의 속성을 통해 폰트명을 찾는다
- 일부 폰트는 적용되지 않는 경우도 있다

```
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm
```

```
font_location = 'C:/Windows/Fonts/전주완판본 순R.ttf' #NanumGothic, HMFMPYUN, HMKMMAG
font_name = fm.FontProperties(fname = font_location).get_name()
matplotlib.rc('font', family = font_name)
```

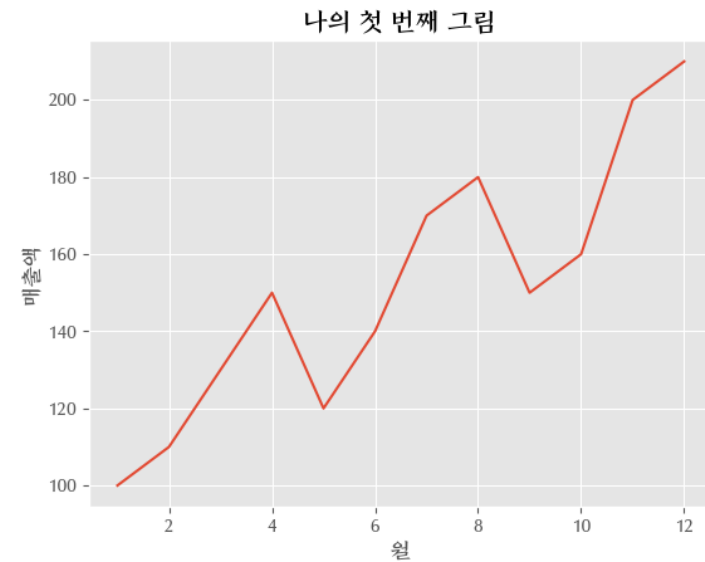
```
plt.style.use("ggplot")
```

```
x = range(1, 13) # range() 함수는 마지막 부분이 제외된다
y = [100, 110, 130, 150, 120, 140, 170, 180, 150, 160, 200, 210]
```

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

```
plt.title("나의 첫 번째 그림")
plt.xlabel("월")
plt.ylabel("매출액")
```

```
plt.show()
```



# 범례 붙이기

```
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm
```

```
font_location = 'C:/Windows/Fonts/NanumGothic.ttf'
font_name = fm.FontProperties(fname = font_location).get_name()
matplotlib.rc('font', family = font_name)
```

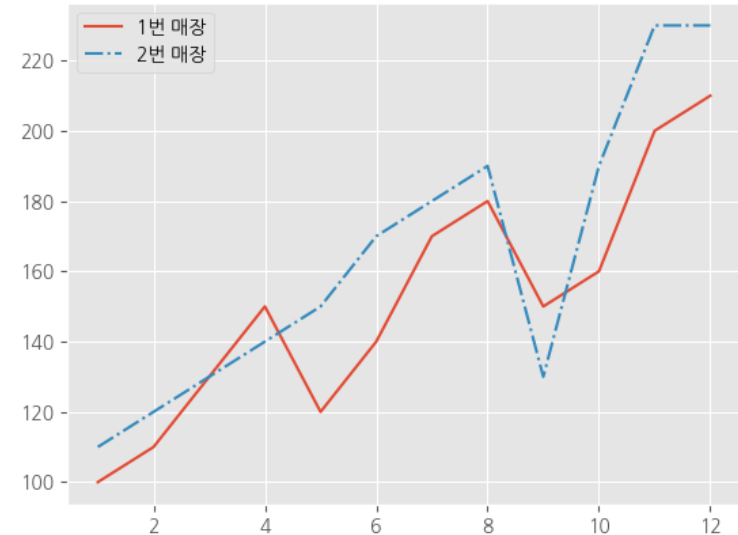
```
plt.style.use("ggplot")
```

```
x = range(1, 13)
y = [100, 110, 130, 150, 120, 140, 170, 180, 150, 160, 200, 210]
y2 = [110, 120, 130, 140, 150, 170, 180, 190, 130, 190, 230, 230]
```

```
plt.plot(x, y, label="1번 매장")
plt.plot(x, y2, linestyle='dashdot', label="2번 매장")
```

```
plt.legend()
plt.show()
```

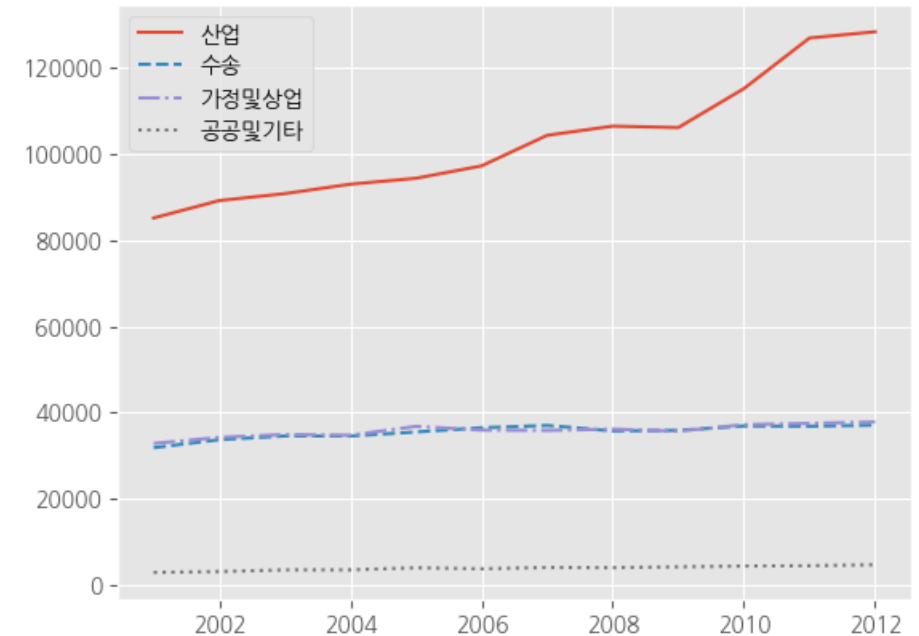
# 범례 붙이기



# 연습문제 1

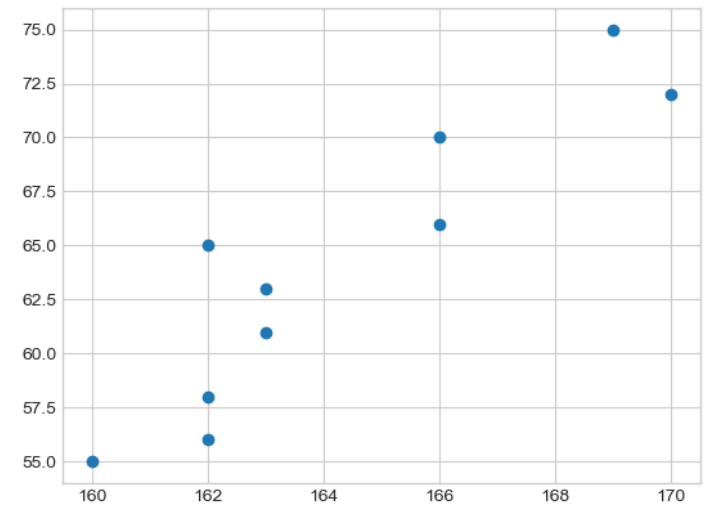
- electricity.csv 파일은 2001~2012년 전기의 사용량을 산업, 수송, 가정및상업, 공공및기타의 4가지로 분류한 연도별 데이터이다
- 4개 분야의 전기 사용량이 연도별로 어떻게 달라졌는지를 그래프로 표현하시오

- hint:
  - Pandas를 이용하시오
  - 산업 = data[data.Part=='산업' ]
  - 산업\_usage = 산업["Usage"]



# 산점도(散点圖, Scatter Plot)

- 선이 아니라 점을 찍어서 그래프를 표현할 수도 있다
- 일련의 흐름이 아니라면 산점도를 그린다



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

```
plt.style.use("seaborn-whitegrid")
```

```
height = [160, 162, 162, 163, 163, 162, 166, 166, 170, 169]
```

```
weight = [55, 56, 58, 61, 63, 65, 66, 70, 72, 75]
```

```
plt.scatter(height, weight)
```

```
plt.show()
```

# 산점도 색상과 크기

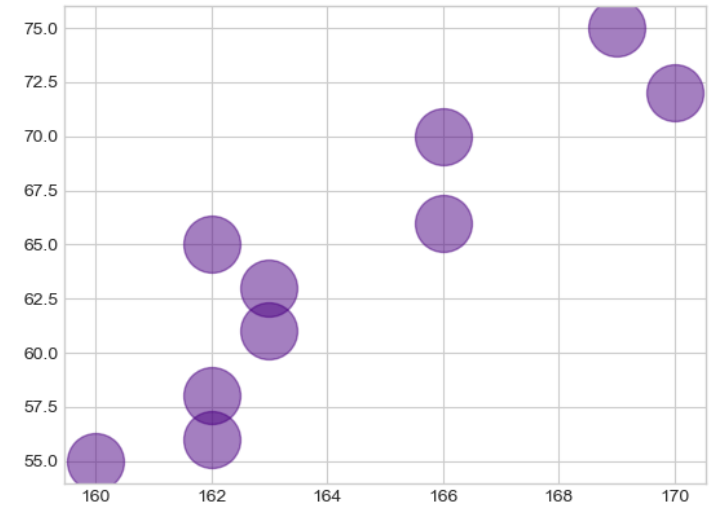
- 산점도의 색, 투명도, 크기를 바꿀 수 있다

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

```
plt.style.use("seaborn-whitegrid")
```

```
height = [160, 162, 162, 163, 163, 162, 166, 166, 170, 169]
weight = [55, 56, 58, 61, 63, 65, 66, 70, 72, 75]
```

```
plt.scatter(height, weight, c="indigo", alpha=0.5, s=1000)
plt.show()
```



# 산점도 겹쳐 그리기

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

```
plt.style.use("seaborn-whitegrid")
```

```
height = [160, 162, 162, 163, 163, 162, 166, 166, 170, 169]
```

```
weight_man = [55, 56, 58, 61, 63, 65, 66, 70, 72, 75]
```

```
weight_woman = [53, 55, 56, 60, 61, 62, 62, 65, 65, 66]
```

```
plt.scatter(height, weight_man, alpha=0.5, s=300, label="male")
```

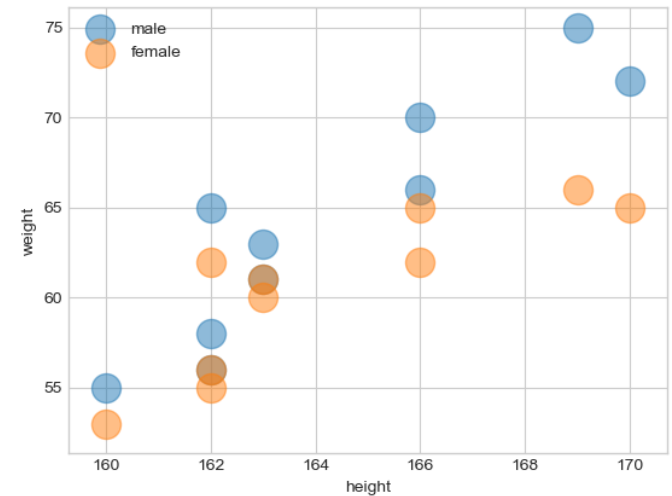
```
plt.scatter(height, weight_woman, alpha=0.5, s=300, label="female")
```

```
plt.xlabel("height")
```

```
plt.ylabel("weight")
```

```
plt.legend()
```

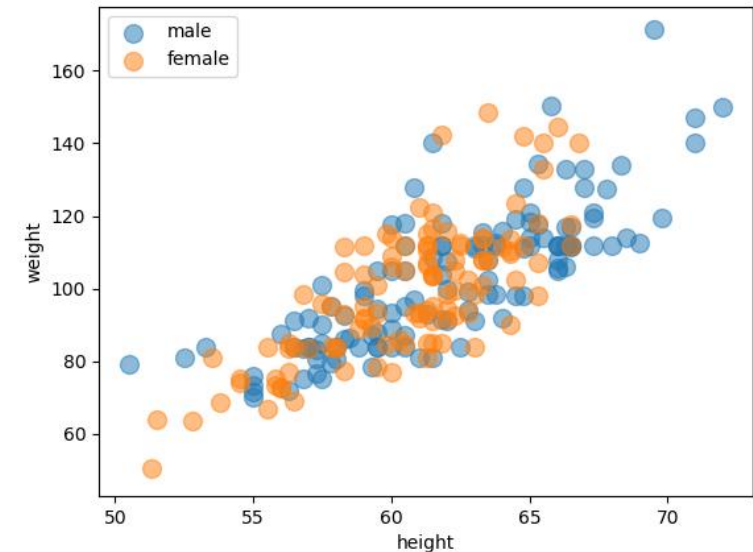
```
plt.show()
```





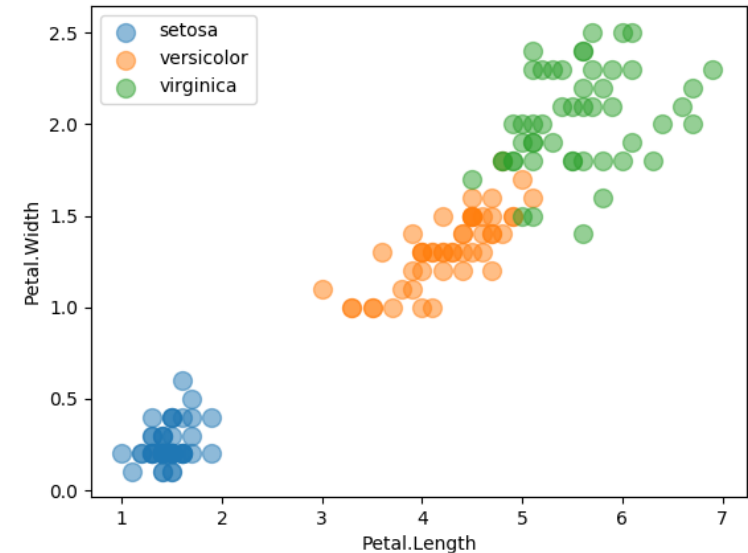
# 연습문제 2

- 미국 남녀 학생의 연령별 키와 몸무게 차이를 기록한 heightweight.csv 파일을 읽고, 남녀의 키(heightIn) 차이를 산점도로 표현하세요



# 연습문제 3

- iris.csv 파일은 붓꽃의 하위 품종인 setosa, versicolor, virginica 의 특징을 기록한 파일입니다
- 품종의 특징이 잘 드러나도록 두 개 컬럼을 선택하여 산점도를 그리세요



# 연습문제 4

- `plt.hist()` 함수는 분포별 빈도를 보여주는 히스토그램을 그린다
- `presidents_heights.csv`를 이용하여 미국 대통령의 신장 분포를 그래프로 표현하되, `plt.style.use()` 함수와 `color`, `alpha`, `edgecolor` 등의 속성을 이용하세요

※ hint: `plt.hist` 를 검색하여 아규먼트 확인  
`plt.hist(np.array(data['height']), ...`

