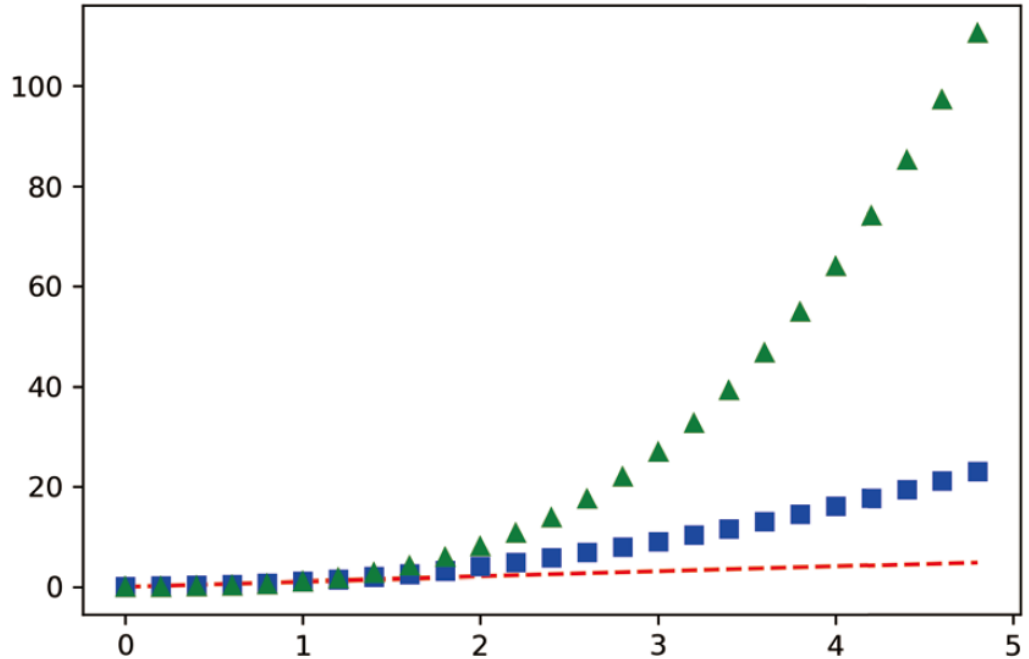


- numpy 라이브러리 알아보기
- numpy 라이브러리를 활용해 그래프 그리기
- numpy 를 활용한 인구분석

# numpy 라이브러리 그래프 그리기

```
import matplotlib.pyplot as plt
t = []
p2 = []
p3 = []
for i in range(0, 50, 2):
    t.append(i / 10)
    p2.append((i / 10) ** 2)
    p3.append((i / 10) ** 3)
plt.plot(t, t, 'r--', t, p2, 'bs', t, p3, 'g^')
plt.show()
```

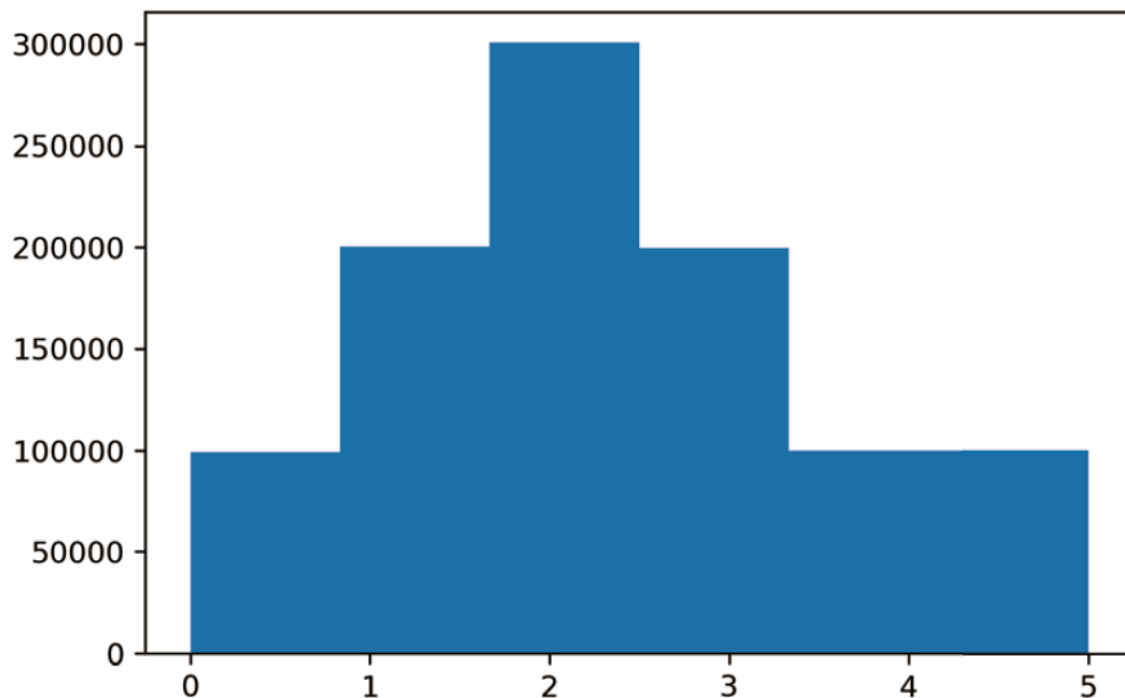


```
import matplotlib.pyplot as plt
import numpy as np
t = np.arange(0., 5., 0.2)
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```

# numpy 라이브러리 그래프 그리기

- 랜덤 확률이 설정된 히스토그램 그리기

```
import matplotlib.pyplot as plt
import numpy as np
dice = np.random.choice(6, 1000000, p=[0.1,0.2,0.3,0.2,0.1,0.1])
plt.hist(dice, bins=6) # 0, 1, 2, 3, 4, 5 중 랜덤으로 추출한 숫자를 히스토그램 표현
plt.show()
```



# numpy 라이브러리 그래프 그리기

- 버블 차트 그리기

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

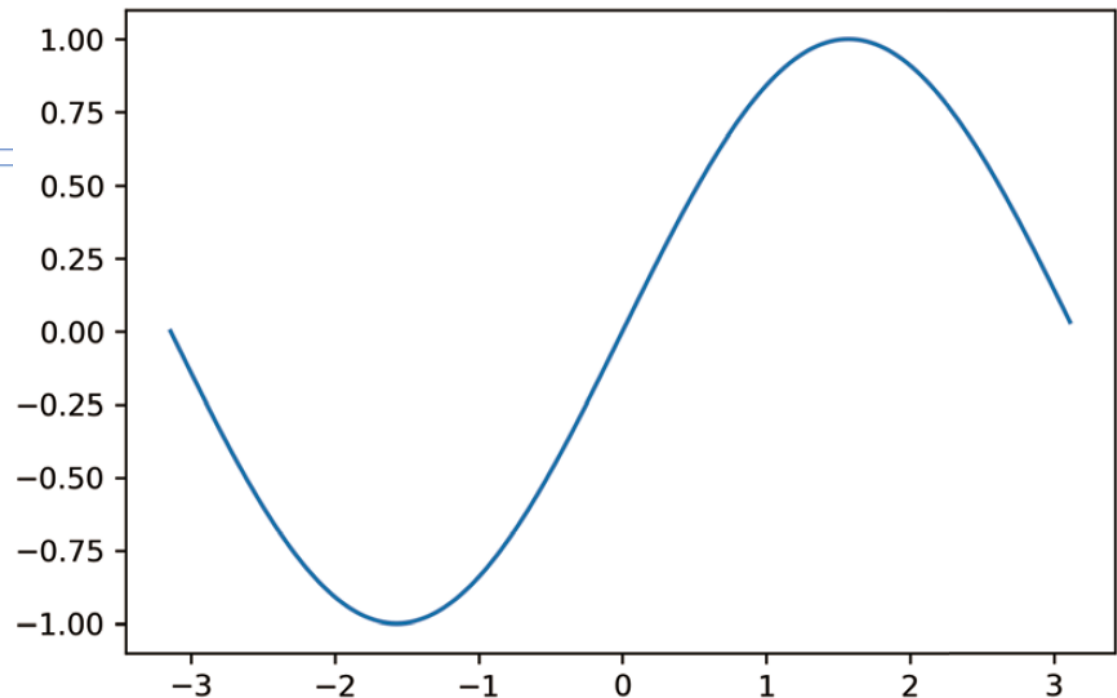
x = np.random.randint(10, 100, 200)
y = np.random.randint(10, 100, 200)
size = np.random.rand(100) * 100

plt.scatter(x, y, s=size, c=x, cmap='jet', alpha=0.7)
plt.colorbar()
plt.show()
```

# numpy 라이브러리 그래프 그리기

- numpy ndarray를 활용한 그래프 그리기

```
import matplotlib.pyplot as plt
import numpy as np
a = np.arange(-np.pi, np.pi, np.pi/100)
plt.plot(a, np.sin(a))
plt.show()
```



# numpy 라이브러리 그래프 그리기

- numpy 라이브러리를 사용하여 재미있는 버블 차트 그리기

```
import matplotlib.pyplot as plt
import numpy as np
x = np.random.randint(-100, 100, 1000) # 1000개의 랜덤 값 추출
y = np.random.randint(-100, 100, 1000) # 1000개의 랜덤 값 추출
size = np.random.rand(100) * 100
mask1 = abs(x) > 50 # x에 저장된 값 중 절댓값이 50보다 큰 값 걸러 냄
mask2 = abs(y) > 50 # y에 저장된 값 중 절댓값이 50보다 큰 값 걸러 냄
x = x[mask1+mask2] # mask1과 mask2 중 하나라도 만족하는 값 저장
y = y[mask1+mask2] # mask1과 mask2 중 하나라도 만족하는 값 저장
plt.scatter(x, y, s=size, c=x, cmap='jet', alpha=0.7)
plt.colorbar()
plt.show()
```

## 처리 목표

- 원하는 지역의 인구분포를 찾아 그래프로 표현
- 전국에서 찾은 지역과 인구분포가 가장 비슷한 지역찾아보기

# numpy 를 활용한 인구분석

```
import numpy as np
import csv
f = open('age.csv')
data = csv.reader(f)
next(data)
name = input('인구 구조가 알고 싶은 지역의 이름(읍면동 단위)을 입력해
주세요 : ')
for row in data :
    if name in row[0] :
        home = np.array(row[3:], dtype=int)
print(home)
```

```
import matplotlib.pyplot as plt
plt.rc('font', family='Malgun Gothic')
plt.title(name+' 지역의 인구 구조')
plt.plot(home)
plt.show()
```



# numpy 를 활용한 인구분석

- 내가 찾은 지역과 가장 비슷한 인구 구조를 가진 지역 찾기
  - ① 전국의 모든 지역 중 한 곳(B)을 선택한다.
  - ② 궁금한 지역 A의 0세 인구수에서 B의 0세 인구수를 뺀다.
  - ③ ②를 100세 이상 인구수에 해당하는 값까지 반복한 후 각각의 차이를 모두 더한다.
  - ④ 전국의 모든 지역에 대해 반복하며 그 차이가 가장 작은 지역을 찾는다.