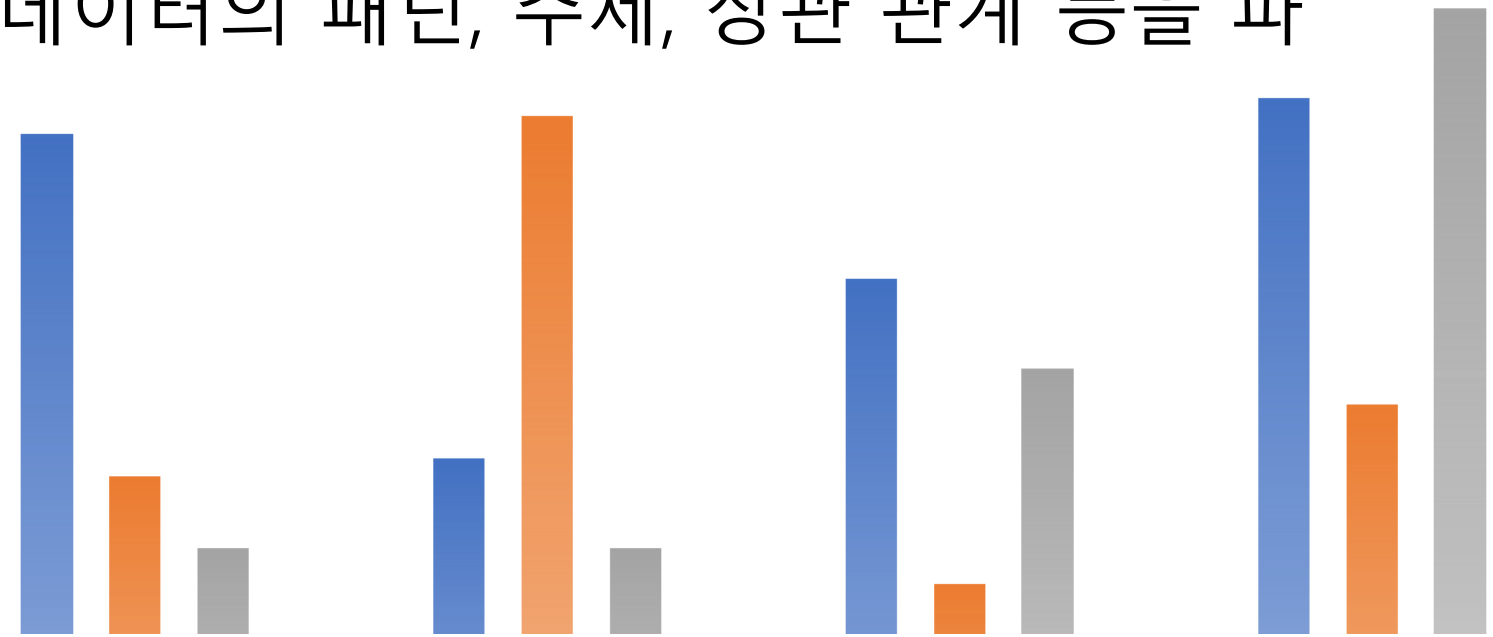
K-Digital Training 스마트 팩토리 3기

데이터 시각화

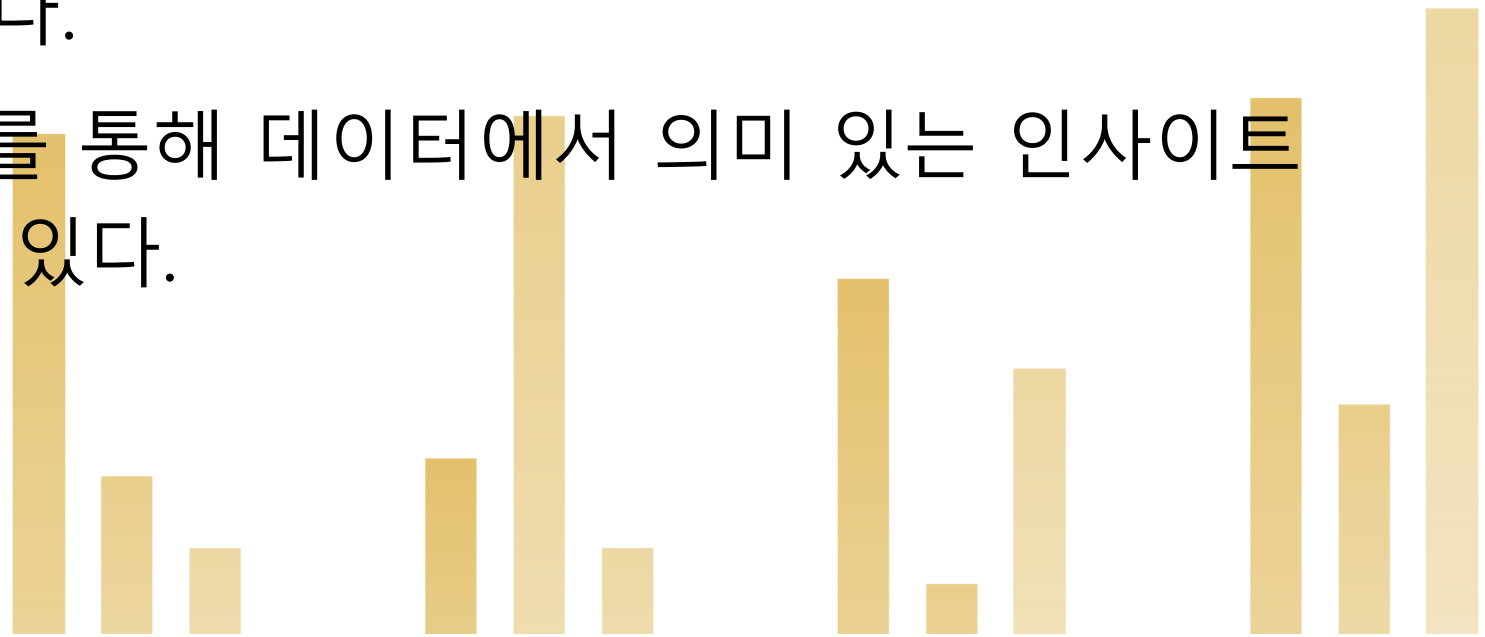
데이터 시각화

- 그래프, 차트, 다이어그램 등 다양한 시각화 도구를 사용하여 데이터를 시각적으로 표현한 것
- 데이터 시각화는 복잡한 데이터 집합을 직관적이고 이해하기 쉬운 형태로 변환하여 데이터의 패턴, 추세, 상관 관계 등을 파악하는데 유리하다.



데이터 시각화의 목적

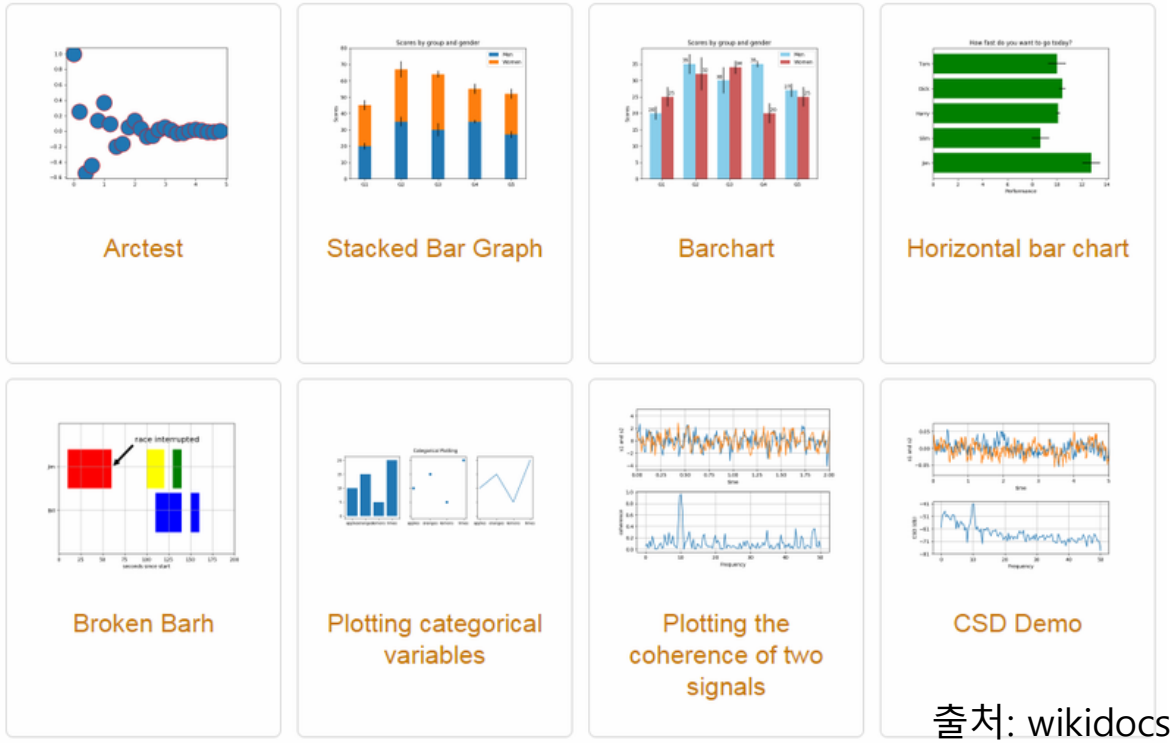
- 데이터의 패턴과 추세 파악: 시각화를 통해 데이터의 패턴, 추세, 이상치 등을 식별하고 분석할 수 있다.
- 데이터 간 관계 이해: 다양한 변수 간의 관계와 상관 관계를 시각적으로 이해할 수 있다.
- 인사이트 도출: 시각화를 통해 데이터에서 의미 있는 인사이트와 통찰력을 도출할 수 있다.



matplotlib

Matplotlib

- 파이썬에서 가장 널리 사용되는 데이터 시각화 라이브러리
- 그래프, 차트, 플롯 등 다양한 시각화 요소를 생성하고 데이터를 시각적으로 나타낼 수 있다.



출처: wikidocs

Google Colab에서 Matplotlib 사용하기

```
[1]  import matplotlib

     matplotlib.__version__
```

```
'3.7.1'
```

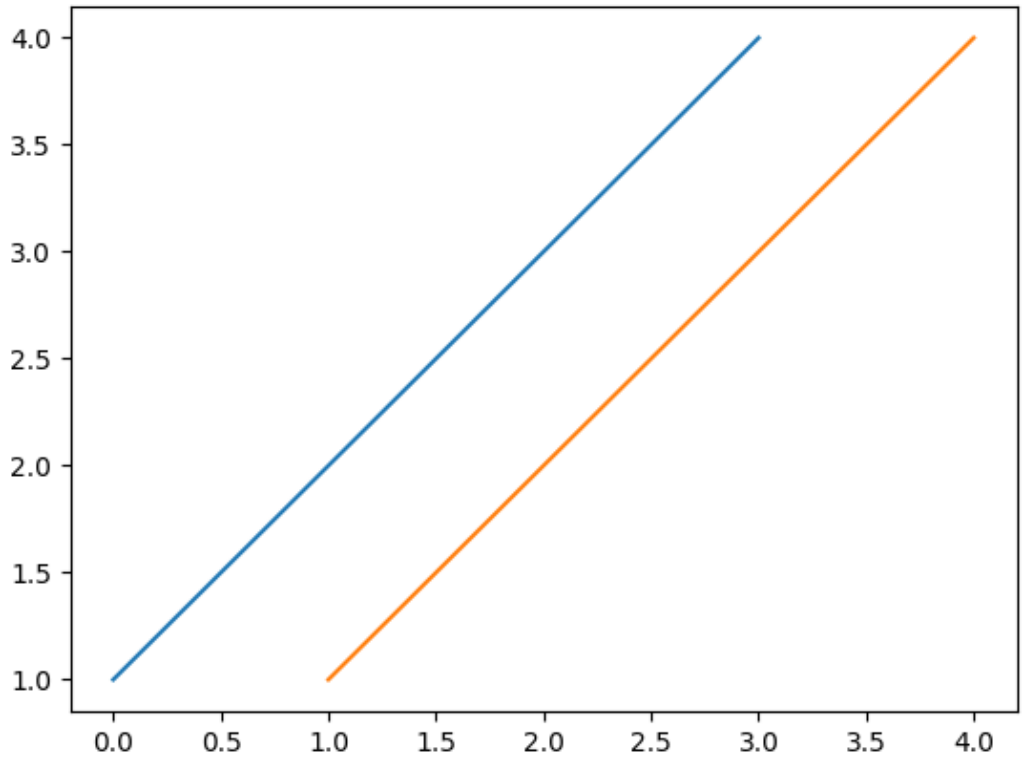
기본 그래프 그리기

- 그래프 그리기
  - plt.plot([x좌표], [y좌표], '포맷 문자열')
- x축, y축 범위 지정
  - plt.axis([xmin, xmax, ymin, ymax])
- 그래프 표시
  - plt.show()

```
import matplotlib.pyplot as plt

# x 좌표를 생략하면 [0, 1, 2, 3, ...]이 자동으로 들어간다.
plt.plot([1, 2, 3, 4])

# (0, 1), (1, 2), (2, 3), (3, 4)
plt.plot([0, 1, 2, 3], [1, 2, 3, 4])
plt.show()
```



Colors

character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

Line Styles

character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style

Markers

character	description
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker

기본 그래프 그리기

- 축 레이블 설정하기
  - plt.xlabel('x축 레이블', [labelpad, loc, fontdict])
  - plt.ylabel('y축 레이블', [labelpad, loc, fontdict])

```
font1 = {'family': 'fantasy', 'color': 'deeppink', 'weight': 'normal', 'size': 'xx-large'}

plt.plot([1, 2, 3, 4], [1, 4, 9, 16])

# labelpad: lable padding(여백), loc: x - left/center/right, y - top/center/bottom, fontdict: 폰트
plt.xlabel('x', labelpad=30, loc='left', fontdict=font1)
plt.ylabel('y', loc='top')
plt.show()
```

기본 그래프 그리기

- 범례 표시하기
  - plt.plot([1, 2, 3, 4], [1, 2, 3, 4], label=‘’)
  - plt.legend(loc=() ncol=열의 개수, fontsize, frameon, shadow)
    - loc=(0, 0) : 왼쪽 아래
    - loc=(1, 1) : 오른쪽 위
    - loc=‘best’ : 겹침이 최소화되는 최적의 위치
    - loc=‘lower right’: 오른쪽 아래
    - 등등 (아래 링크 참고)
- https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.legend.html

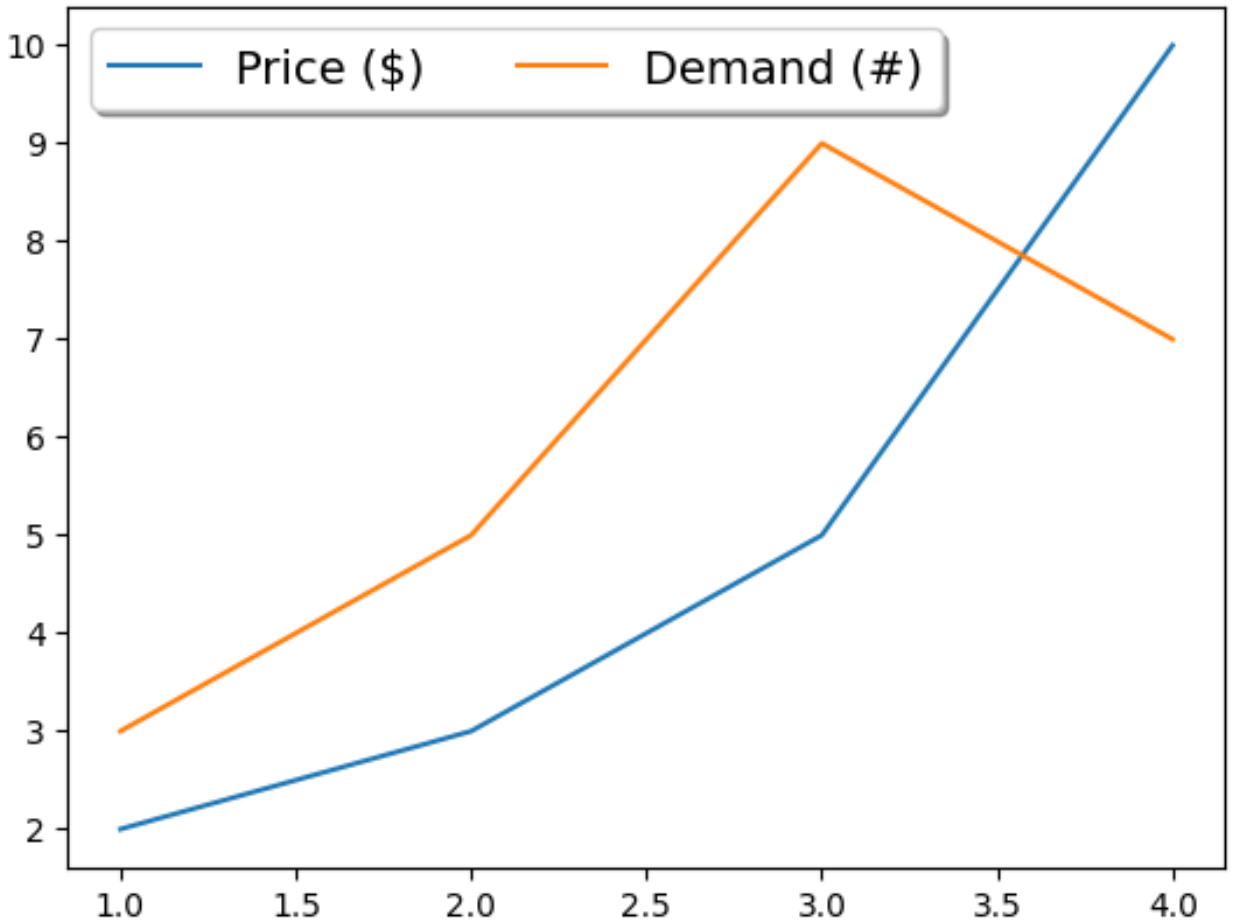
기본 그래프 그리

- 범례 표시하기

```
# 범례 표시하기

plt.plot([1, 2, 3, 4], [2, 3, 5, 10], label='Price ($)')
plt.plot([1, 2, 3, 4], [3, 5, 9, 7], label='Demand (#)')
plt.legend(loc='best', ncol=2, fontsize=14, frameon=True, shadow=True)

plt.show()
```



기본 그래프 그리기

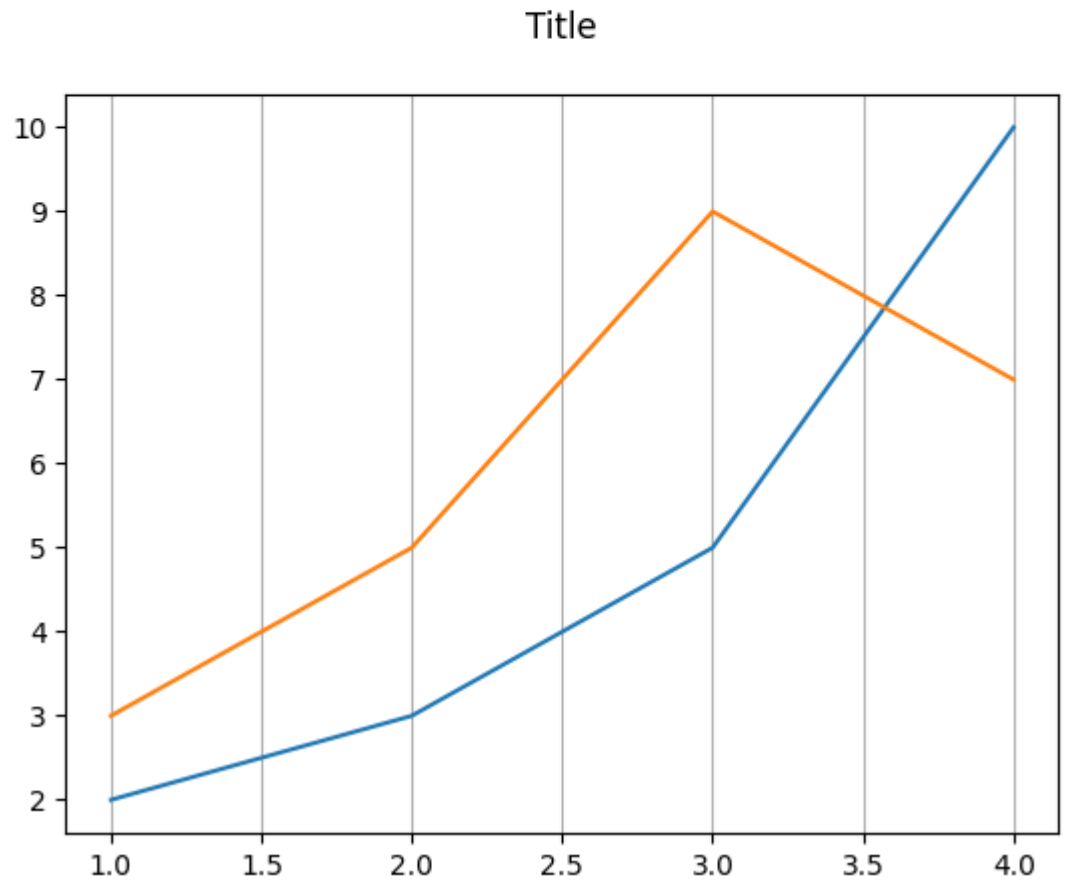
- 타이틀 설정하기
  - plt.title('타이틀', [loc, pad])
- 그리드 설정하기
  - plt.grid(True, axis)
    - axis = 'x' or 'y' or 'both'

```
[91]  # grid, title

      plt.plot([1, 2, 3, 4], [2, 3, 5, 10])
      plt.plot([1, 2, 3, 4], [3, 5, 9, 7])

      plt.title('Title', loc='center', pad=20)

      # axis = {'both', 'x', 'y'}
      plt.grid(True, axis='x')
      plt.show()
```



막대 그래프 그리기

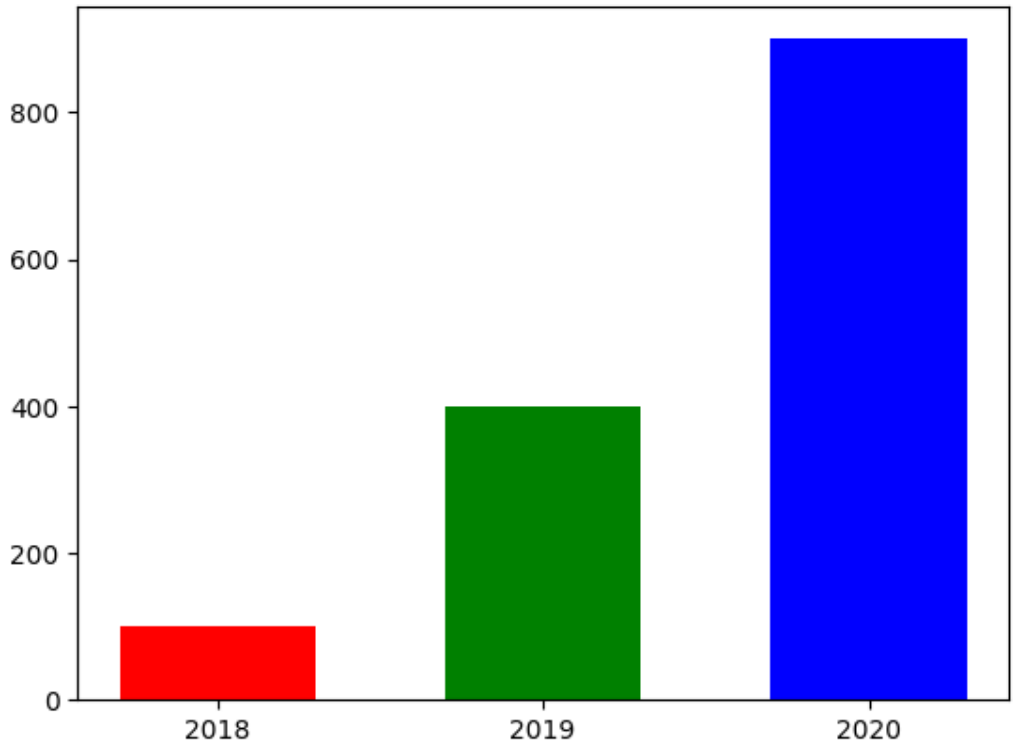
- 그래프 그리기
  - plt.bar([x값], [y값], color, width)
- 눈금 표시하기
  - plt.xticks([x값], [눈금])
  - plt.yticks([y값], [눈금])

```
[102] import matplotlib.pyplot as plt
      import numpy as np

      x = np.arange(3)

      # 막대 그래프 그리기
      plt.bar(x, [100, 400, 900], color=['r', 'g', 'b'], width=0.6)

      # 눈금 표시
      plt.xticks(x, ['2018', '2019', '2020'])
      plt.show()
```



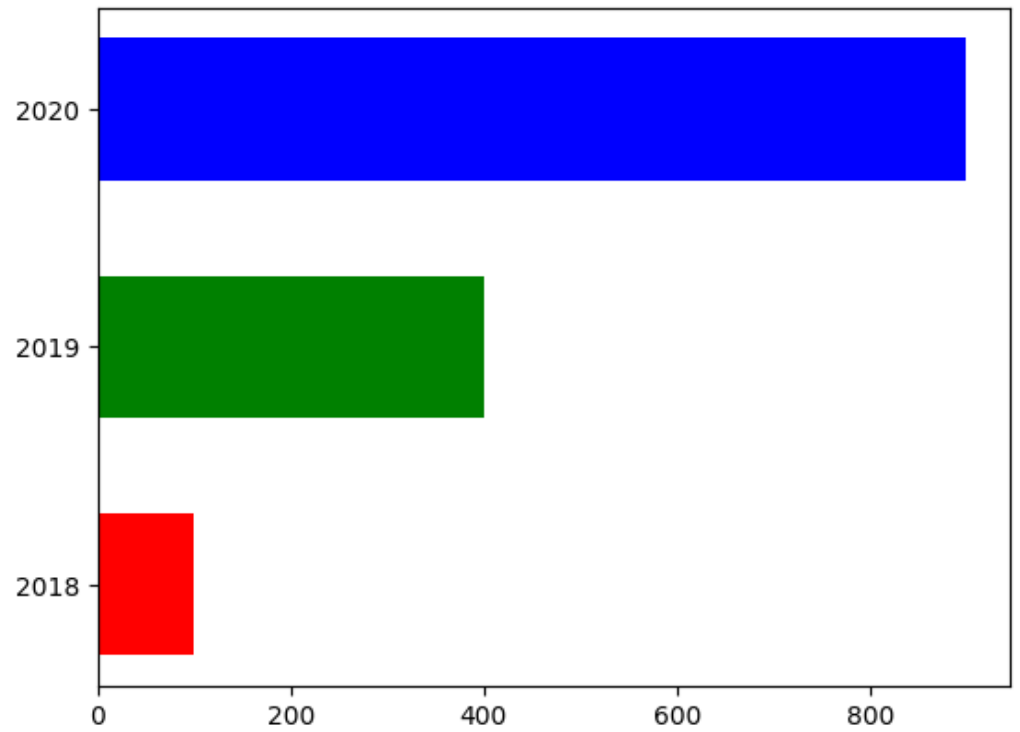
수평 막대 그래프 그리기

- 그래프 그리기
  - plt.barh([y값], [x값], color, height)

```
[104] y = np.arange(3)

      # 막대 그래프 그리기
      plt.barh(y, [100, 400, 900], color=['r', 'g', 'b'], height=0.6)

      # 눈금 표시
      plt.yticks(y, ['2018', '2019', '2020'])
      plt.show()
```



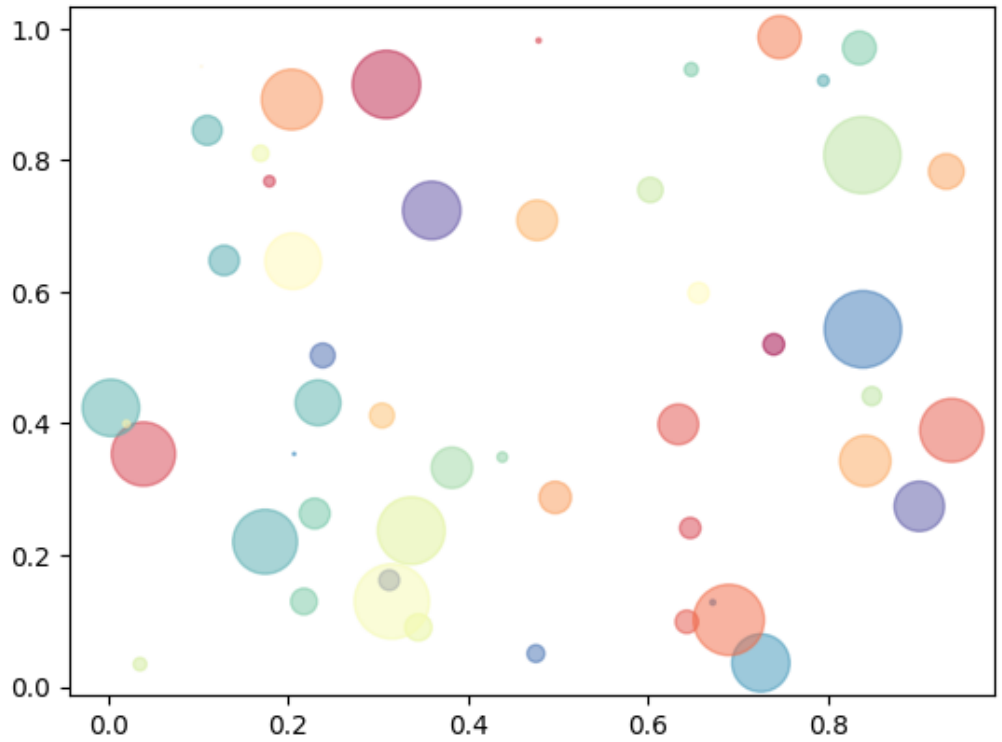
산점도 그리기

```
[141] n = 50
      x = np.random.rand(n)
      y = np.random.rand(n)

      area = (30 * np.random.rand(n)) ** 2 # 0과 30사이 난수
      colors = np.random.rand(n)

      # cmap : viridis(default), Spectral, plasma, inferno, magma, cividis, ...
      plt.scatter(x, y, s=area, c=colors, cmap='Spectral', alpha=0.5)
      plt.show()
```

- 그래프 그리기
  - plt.scatter([x값], [y값], s, c, cmap, alpha)
    - s: 마커의 크기
    - c: 마커의 색상
    - cmap: 컬러맵
    - alpha: 투명도



파이차트 그리기

- 그래프 그리기
  - plt.pie([비율], labels, autopct, explode, colors, wedgeprops)
    - labels : 레이블
    - autopct : 소수점 자리
    - explode : 부채꼴이 파이 차트의 중심에서 벗어나는 정도
    - colors : 색
    - wedgeprops : 부채꼴 스타일

파이차트 그리기

```
ratio = [34, 32, 16, 18]
labels = ['Apple', 'Banana', 'Melon', 'Grapes']
wedgeprops={'width': 0.7, 'edgecolor': 'k', 'linewidth': 1}

plt.pie(ratio, labels=labels, autopct='%.1f%%', explode=[0, 0.1, 0, 0.1], colors=['red', 'yellow', 'green', 'purple'], wedgeprops=wedgeprops)
plt.show()
```