
파이썬 기반
빅데이터 처리
및 분석기술
Crayon

빅데이터 시각화

2 선그림과 점그림 작성

- 기본 선그림 작성
- 플롯 조정 : 색상과 스타일
- 기본 점그림 작성
- 플롯 조정 : 마커 색상과 스타일

1. Matplotlib 사용법

1) 선그림(Line plot)

- $y=f(x)$ 시각화

- 변수 x의 값이 연속적인 경우
- x의 변화에 대한 y의 변화, 추세를 시각적으로 표시

1. Matplotlib 사용법

1) 선그림(Line plot)

간단한 선그림

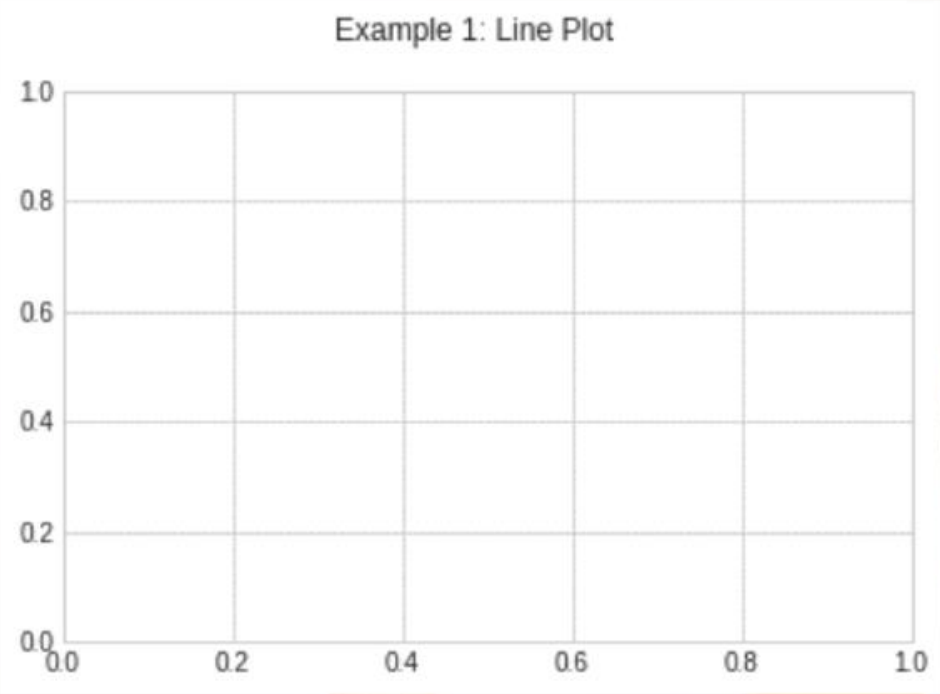
- %matplotlib inline
- import matplotlib.pyplot as plt
- plt.style.use('seaborn-whitegrid')
- import numpy as np

1. Matplotlib 사용법

1) 선그림(Line plot)

◆ 간단한 선그림

- fig = plt.figure()
- fig.suptitle("Example 1: Line Plot")



1. Matplotlib 사용법

1) 선그림(Line plot)

◆ 간단한 선그림

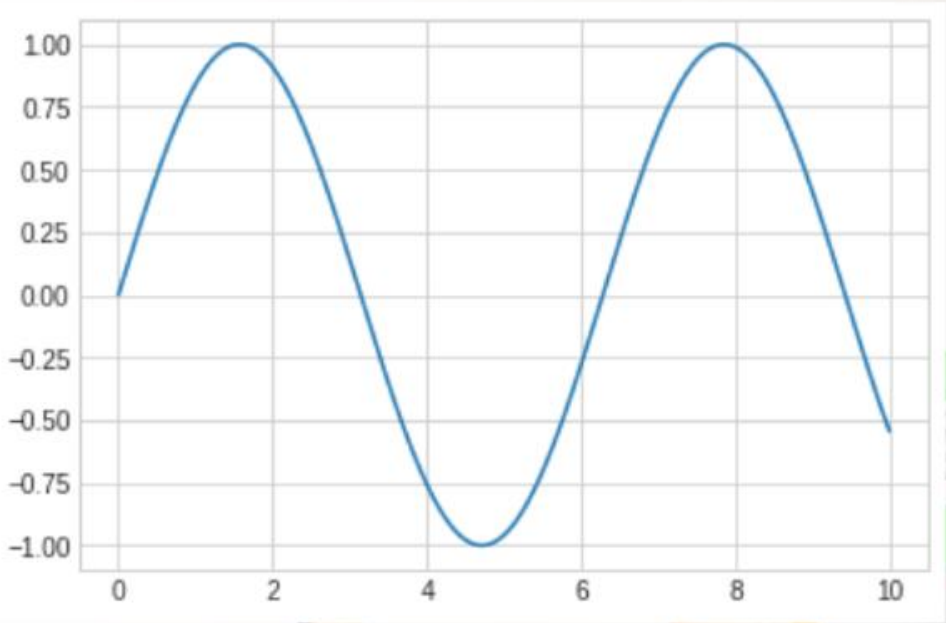
- fig = plt.figure()
- fig.suptitle("Example 1: Line Plot")
- ax = plt.axes()

1. Matplotlib 사용법

1) 선그림(Line plot)

◆ 간단한 선그림

- fig = plt.figure()
- ax = plt.axes()

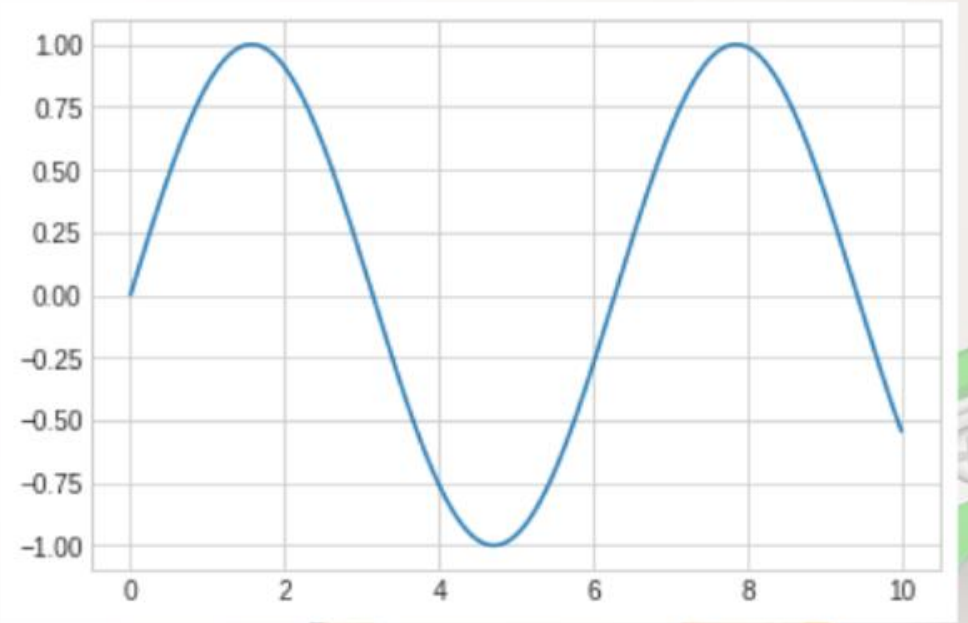


1. Matplotlib 사용법

1) 선그림(Line plot)

◆ 간단한 선그림

- plt.plot(x, y);
- ax = plt.axes()

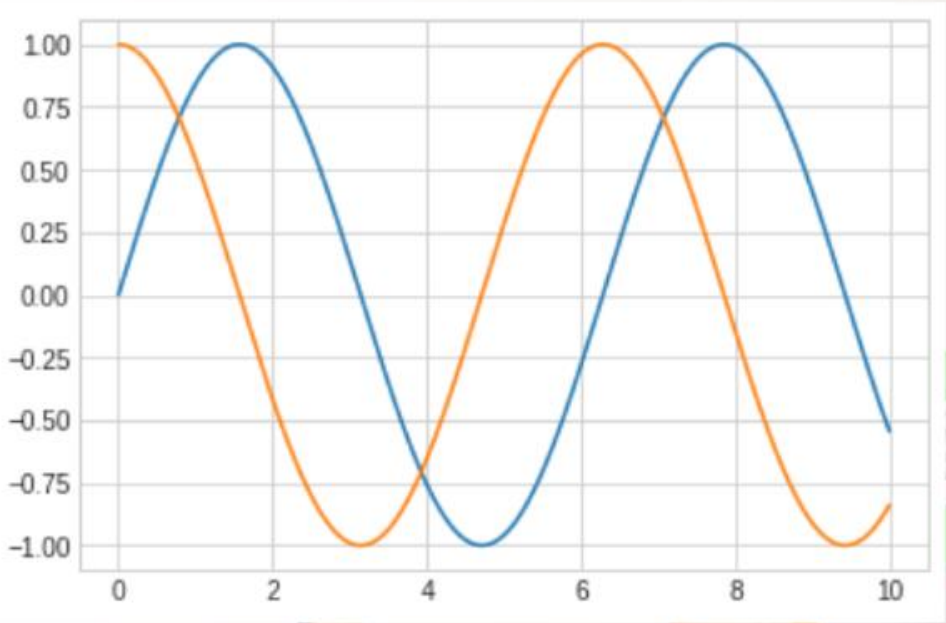


1. Matplotlib 사용법

1) 선그림(Line plot)

◆ 간단한 선그림

- plt.plot(x, np.sin(x))
- plt.plot(x, np.cos(x));



1. Matplotlib 사용법

2) 플롯 조정

◆ 선색과 스타일

plt.plot(x, np.sin(x - 0), color='blue')	색상을 이름으로 지정
plt.plot(x, np.sin(x - 1), color='g')	짧은 색상 코드 (rgb, cmyk)
plt.plot(x, np.sin(x - 2), color='0.75')	회색조 지정, 0과 1사이 값
plt.plot(x, np.sin(x - 3), color='#FFDD44')	16진수 코드(RRGGBB, 00 ~ FF 사이)
plt.plot(x, np.sin(x - 4), color=(1.0,0.2,0.3))	RGB 튜플, 0과 1 사이 값
plt.plot(x, np.sin(x - 5), color='chartreuse');	모든 HTML 색상 이름을 지원

1. Matplotlib 사용법

2) 플롯 조정

◆ 선색과 스타일

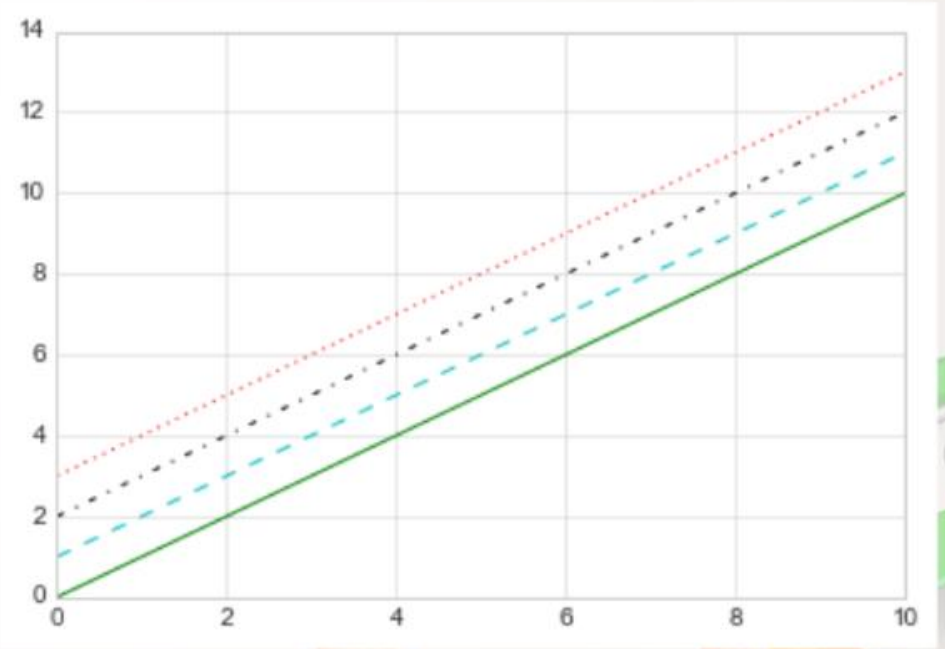
plt.plot(x, x + 0, linestyle='solid')	키워드 사용
plt.plot(x, x + 1, linestyle='dashed')	
plt.plot(x, x + 2, linestyle='dashdot')	
plt.plot(x, x + 3, linestyle='dotted');	
plt.plot(x, x + 4, linestyle='-')	solid와 동일한 단축 기호
plt.plot(x, x + 5, linestyle='--')	dashed와 동일한 단축 기호
plt.plot(x, x + 6, linestyle='-.')	dashdot와 동일한 단축 기호
plt.plot(x, x + 7, linestyle=':');	dotted와 동일한 단축 기호

1. Matplotlib 사용법

2) 플롯 조정

◆ 선색과 스타일

- plt.plot(x, x + 0, '-g') → solid green
- plt.plot(x, x + 1, '--c') → dashed cyan
- plt.plot(x, x + 2, '-.k') → dashdot black
- plt.plot(x, x + 3, ':r'); → dotted red

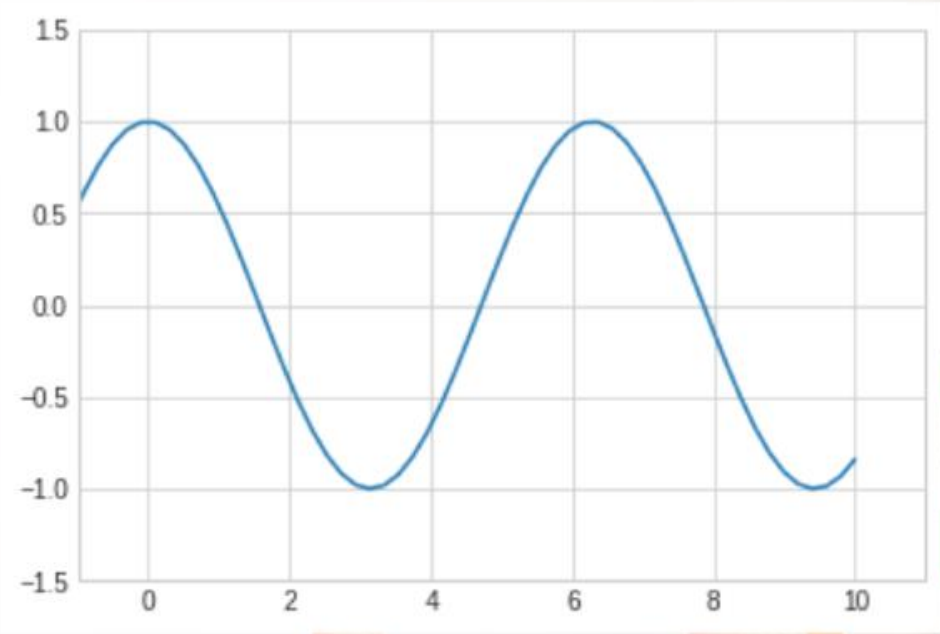


1. Matplotlib 사용법

2) 플롯 조정

◆ 축의 범위 지정

- plt.plot(x, np.cos(x))
- plt.xlim(-1, 11)
- plt.ylim(-1.5, 1.5);

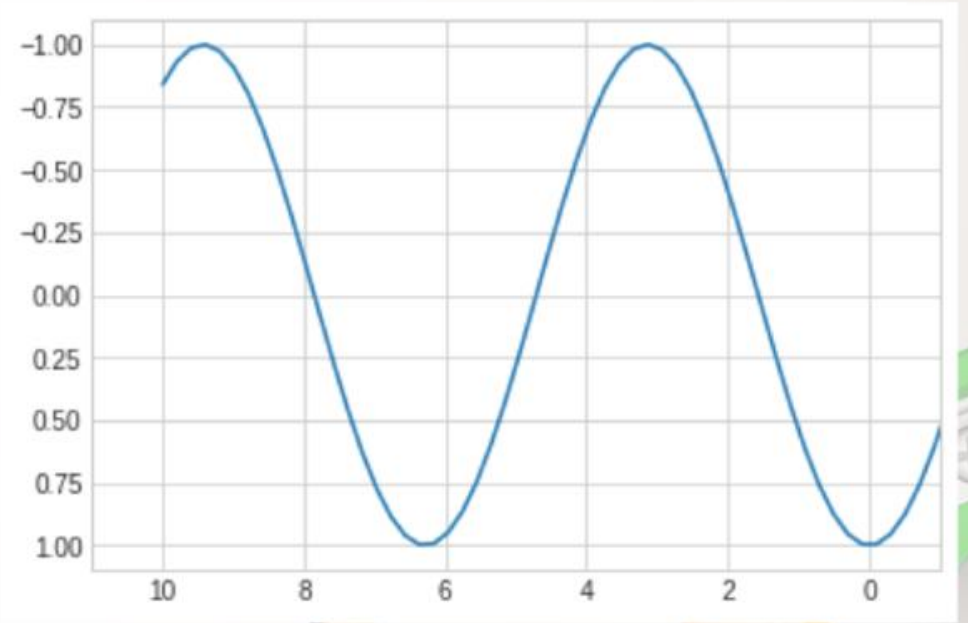


1. Matplotlib 사용법

2) 플롯 조정

◆ 축의 범위 지정

- plt.plot(x, np.cos(x))
- plt.xlim(11, -1)
- plt.ylim(1.1, -1.1);

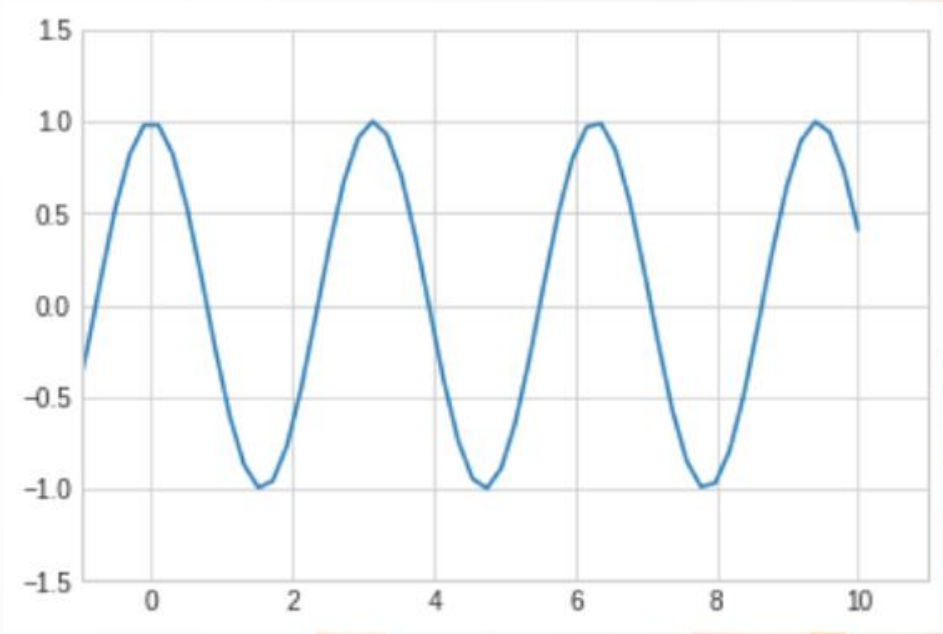


1. Matplotlib 사용법

2) 플롯 조정

◆ 축의 범위 지정

- plt.plot(x, np.cos(x*2))
- plt.axis([-1, 11, -1.5, 1.5]);

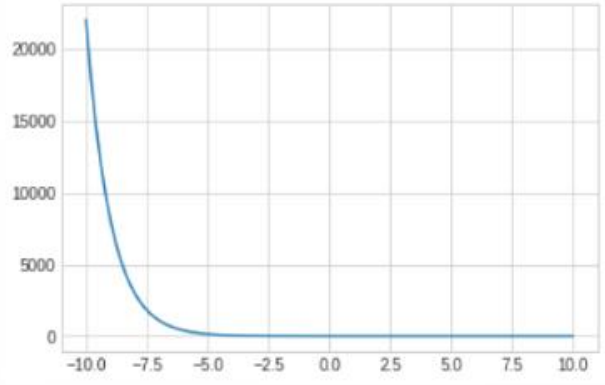


1. Matplotlib 사용법

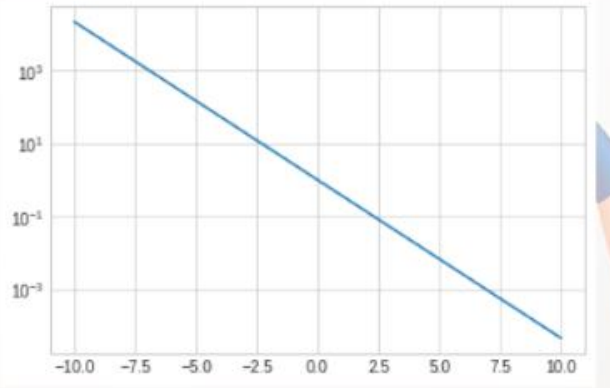
2) 플롯 조정

◆ 축의 스케일 변경

- plt.plot(x, np.exp(-x));



- plt.plot(x, np.exp(-x))
- plt.yscale("log");

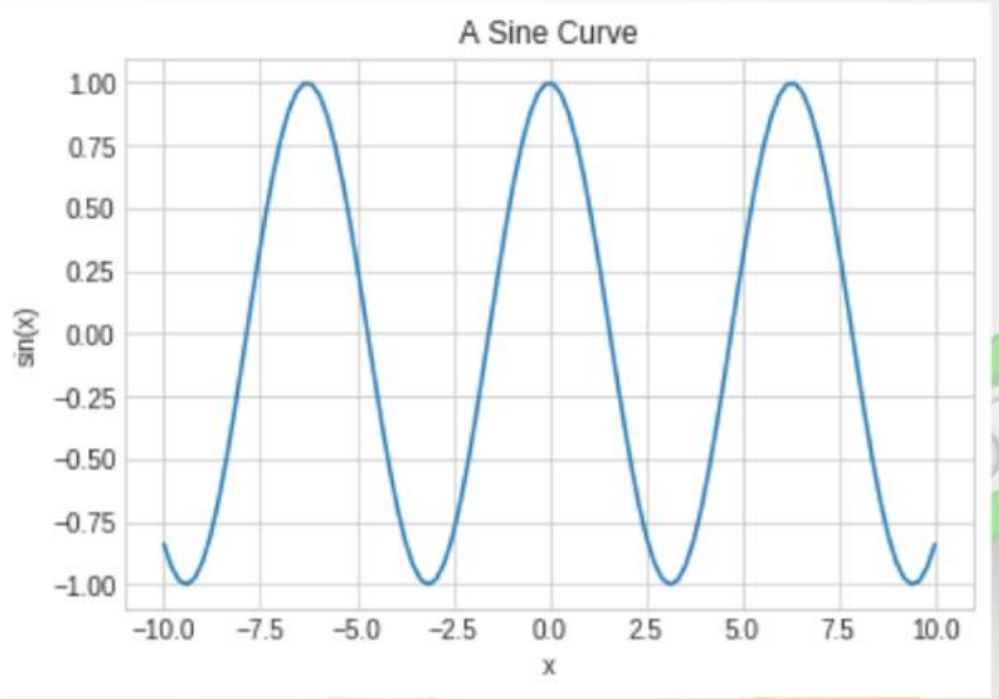


1. Matplotlib 사용법

2) 플롯 조정

◆ 축의 레이블 추가

- plt.plot(x, np.cos(x))
- plt.title("A Cosine Curve")
- plt.xlabel("x")
- plt.ylabel("sin(x)");

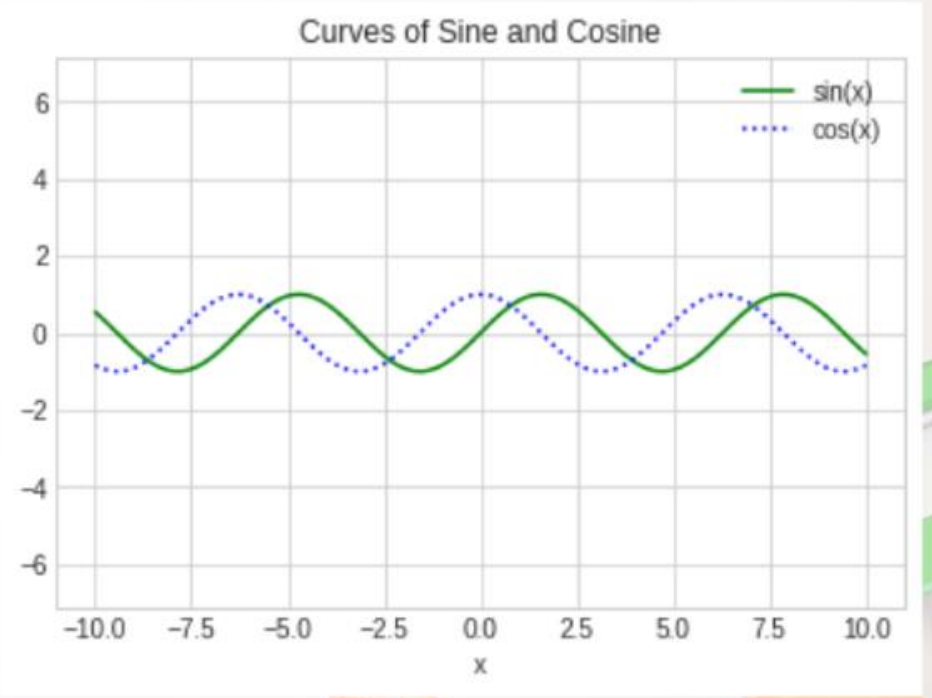


1. Matplotlib 사용법

2) 플롯 조정

◆ 범례(Legend) 추가

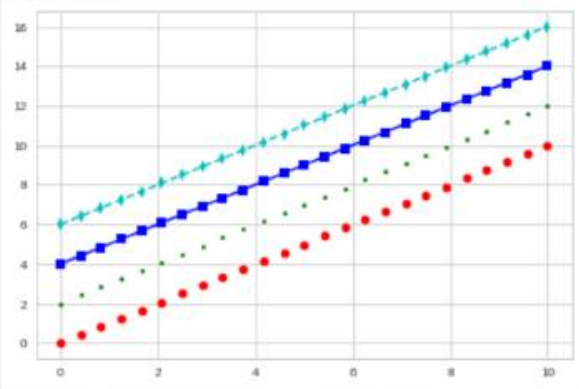
- plt.plot(x, np.sin(x), '-g', label='sin(x)')
- plt.plot(x, np.cos(x), ':b', label='cos(x)')
- plt.axis('equal')
- plt.title('Curves of Sine and Cosine')
- plt.xlabel('x')
- plt.legend();



1. Matplotlib 사용법

3) 점그림(Scatter plot)

◆ 간단한 점그림

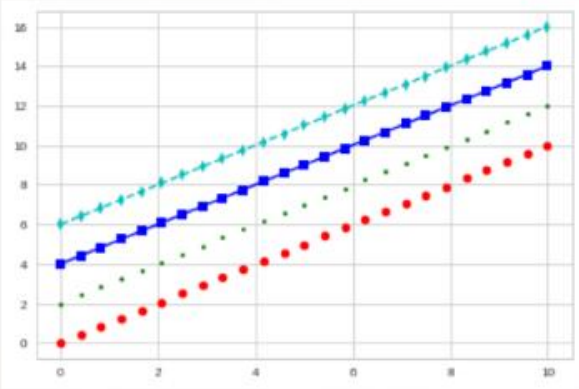


- x = np.linspace(0, 10, 25)
- plt.plot(x, x, 'o', color='r')
- plt.plot(x, x+2, '.', color='g')
- plt.plot(x, x+4, 's-', color='b')
- plt.plot(x, x+6, 'd--', color='c');

1. Matplotlib 사용법

3) 점그림(Scatter plot)

◆ 간단한 점그림



- x = np.linspace(0, 10, 25)
- plt.plot(x, x, 'o', color='r')
- plt.plot(x, x+2, '.', color='g')
- plt.plot(x, x+4, 's-', color='b')
- plt.plot(x, x+6, 'd--', color='c');

1. Matplotlib 사용법

3) 점그림(Scatter plot)

◆ 간단한 점그림

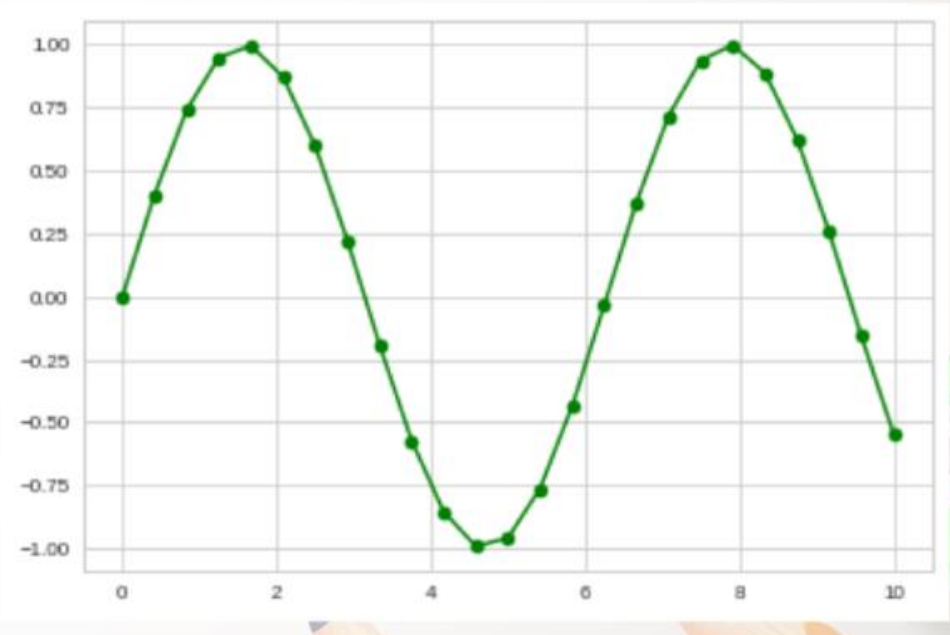
- plt.plot(x, np.sin(x), '-og');

1. Matplotlib 사용법

3) 점그림(Scatter plot)

◆ 간단한 점그림

- plt.plot(x, np.sin(x), '-og');



1. Matplotlib 사용법

4) 플롯 조정

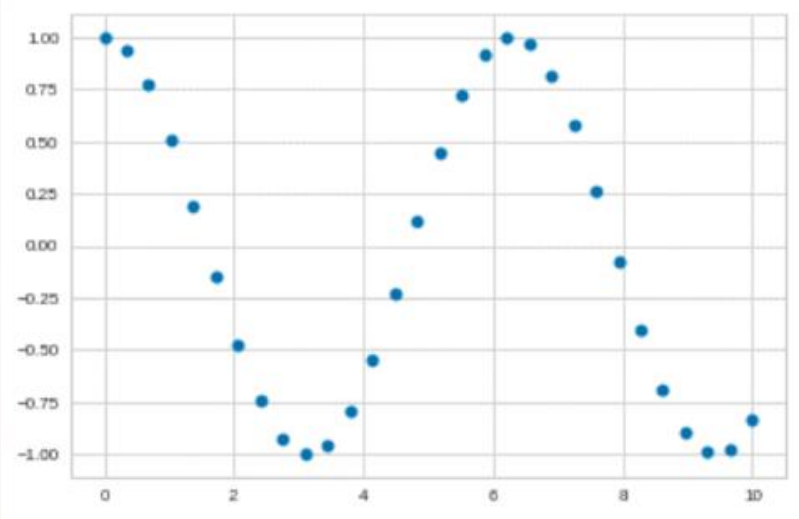
◆ 선과 마커의 종류 색상 변경

```
x = np.linspace(0, 10, 30)
y = np.cos(x)
 plt.plot(x, y,
      '-o',                    → 실선과 원형 마커
      color='gray',            → 선의 색은 회색
      markersize=10,           → 마커의 크기
      linewidth=3,             → 선의 두께
      markerfacecolor='white', → 마커의 색
      markeredgecolor='gray',  → 마커 테두리
      markeredgewidth=1.5)     → 마커 테두리 두께

plt.ylim(-1.2, 1.2);
```

1. Matplotlib 사용법

5) plt.scatter




```
plt.scatter(x, y, marker='o');
```

1. Matplotlib 사용법

5) plt.scatter

```
rng = np.random.RandomState(seed=0)
data = rng.randn(50, 2)
x, y = data[:, 0], data[:, 1] colors = rng.rand(50)
sizes = 1000 * rng.rand(50) plt.subplot(2, 1, 1)
plt.scatter(x, y, c=colors, s=sizes, alpha=0.4,
        cmap='plasma')
plt.colorbar()                                  → 색상 척도 표시
 plt.subplot(2, 1, 2)
plt.scatter(x, y, c=colors, s=sizes, alpha=0.4,
        cmap='viridis')
plt.colorbar();                                 → 색상 척도 표시
```

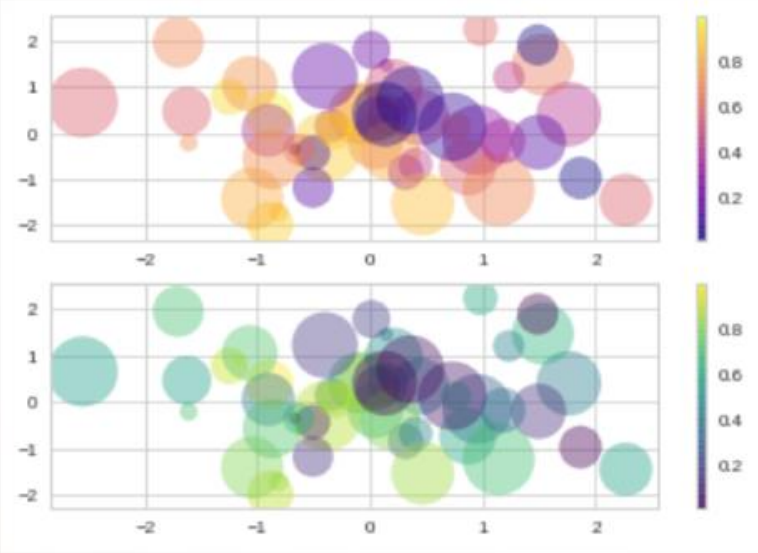
1. Matplotlib 사용법

5) plt.scatter

```
rng = np.random.RandomState(seed=0)
data = rng.randn(50, 2)
x, y = data[:, 0], data[:, 1] colors = rng.rand(50)
sizes = 1000 * rng.rand(50) plt.subplot(2, 1, 1)
plt.scatter(x, y, c=colors, s=sizes, alpha=0.4,
        cmap='plasma')
plt.colorbar()                                  → 색상 척도 표시
 plt.subplot(2, 1, 2)
plt.scatter(x, y, c=colors, s=sizes, alpha=0.4,
        cmap='viridis')
plt.colorbar();                                 → 색상 척도 표시
```

1. Matplotlib 사용법

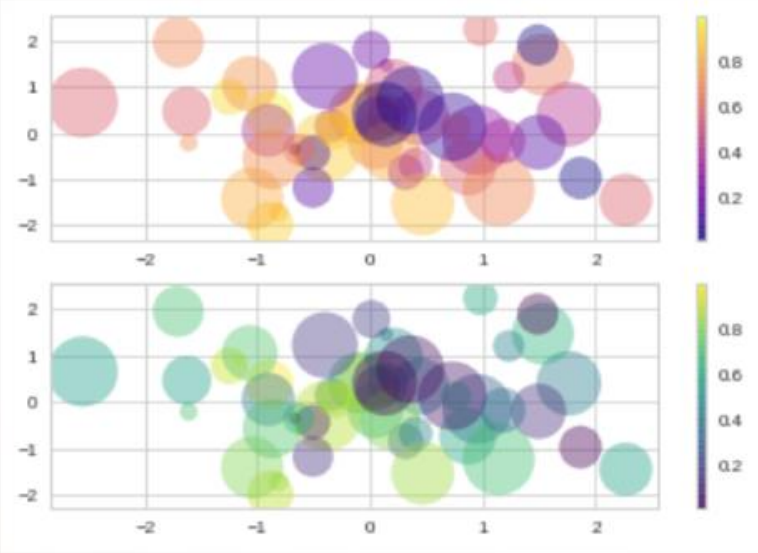
5) plt.scatter



다중 서브 플롯

1. Matplotlib 사용법

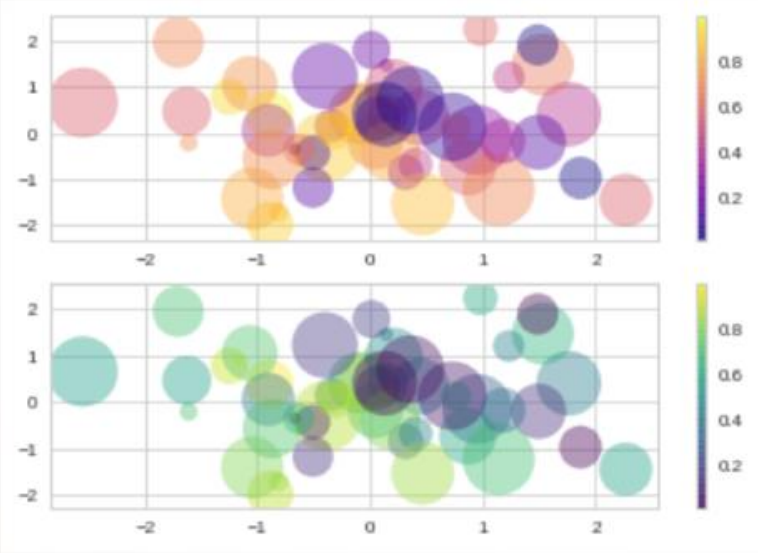
5) plt.scatter



(x, y)값으로 표준 정규 분로를 따르는 난수 생성

1. Matplotlib 사용법

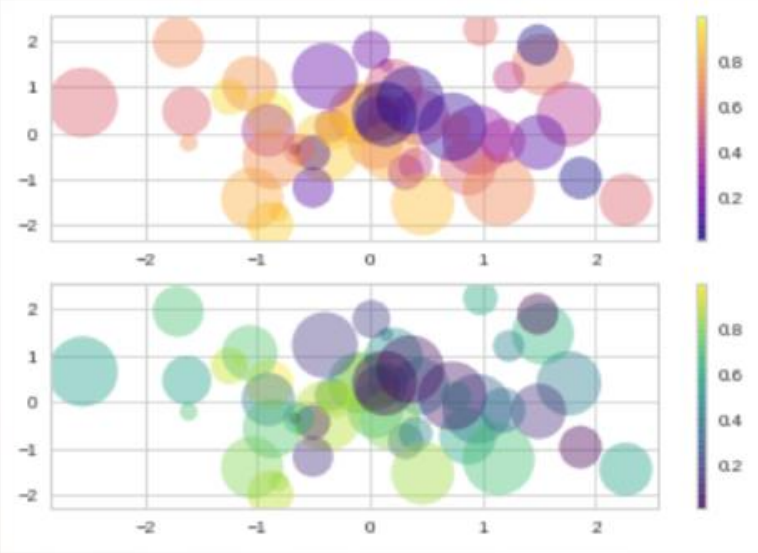
5) plt.scatter



컬러의 값은 0과 1사이

1. Matplotlib 사용법

5) plt.scatter



색상 표시는 plt.colorbar() 명령어 사용

이번 시간에는

2 선그림과 점그림 작성

- 기본 선그림 작성
- 플롯 조정 : 색상과 스타일
- 기본 점그림 작성
- 플롯 조정 : 마커 색상과 스타일

이번 시간에는

실습 참고 자료

- **Colab 노트북 파일**
- **Matplotlib 공식 사이트**
  - https://matplotlib.org/tutorials/index.html

다음 시간에는

3 다중 서브 플롯

- subplot() 함수 사용
- subplots() 함수 사용
- GridSpec() 함수 사용