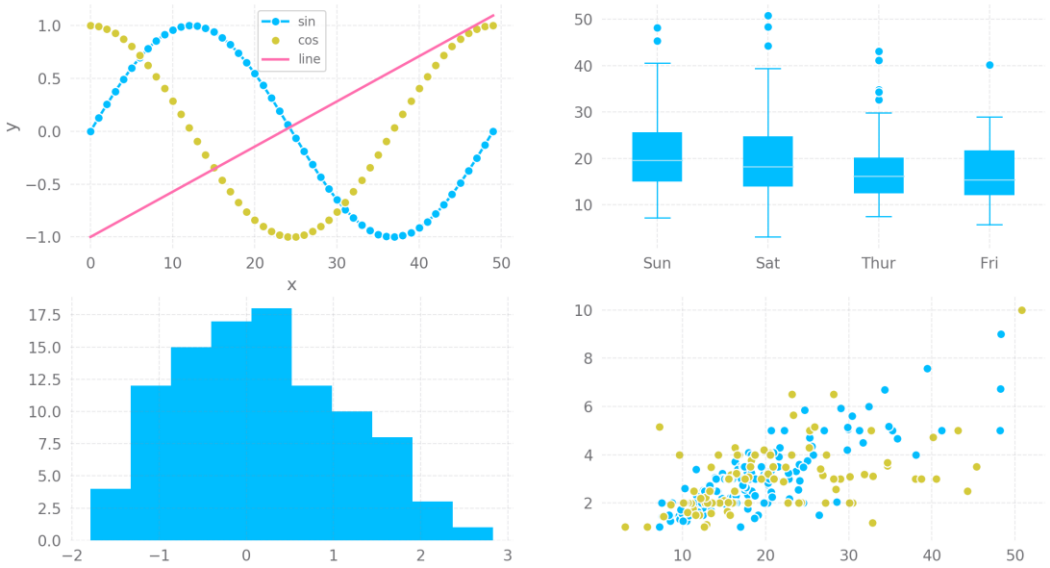
AI Programming

Lecture 22

Preview

- **Matplotlib**



matplotlib

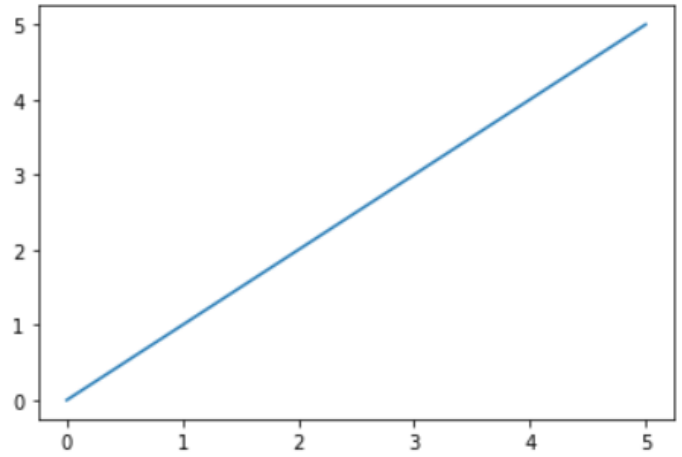
Basics

- **Import**

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

- **Plot with Python list**

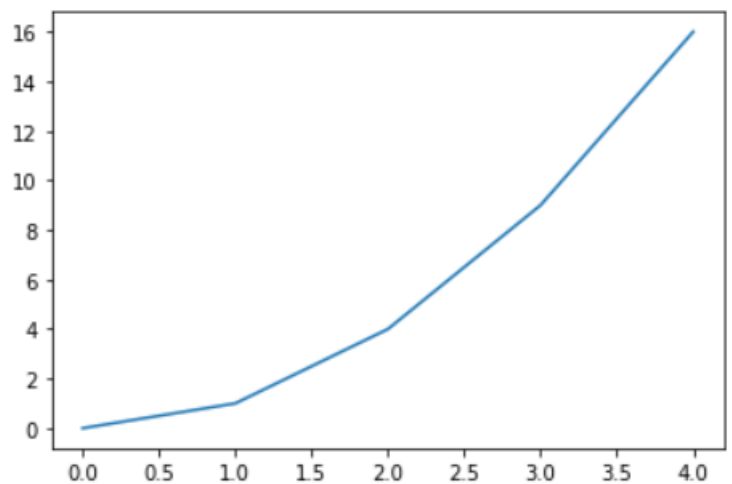
```
x = [0, 1, 2, 3, 4, 5]
plt.plot(x)
plt.show()
```



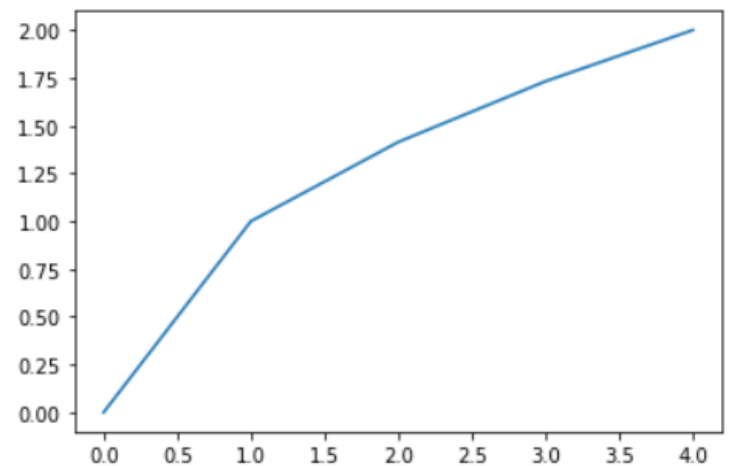
Basics

- **Plot with NumPy ndarray**

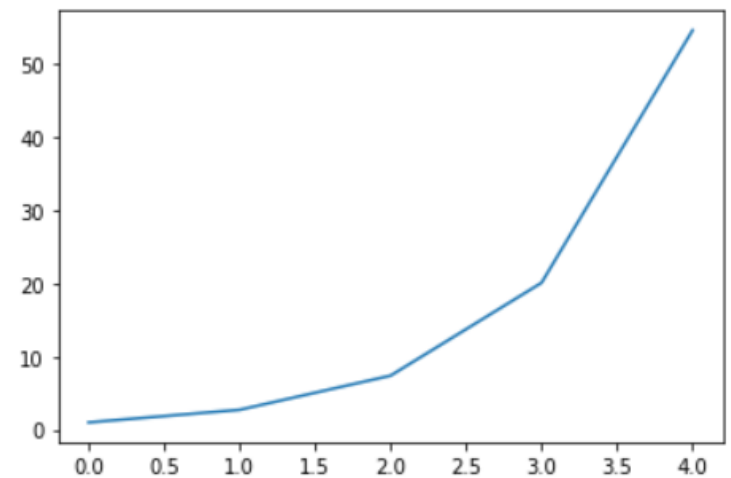
```
import numpy as np

x = np.arange(5)
plt.plot(x, x**2)
plt.show()
```




```
import numpy as np

x = np.arange(5)
plt.plot(x, np.sqrt(x))
plt.show()
```



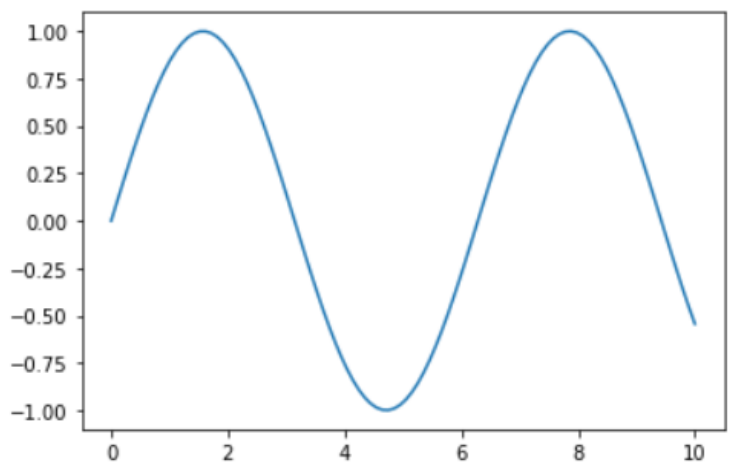

```
import numpy as np

x = np.arange(5)
plt.plot(x, np.exp(x))
plt.show()
```



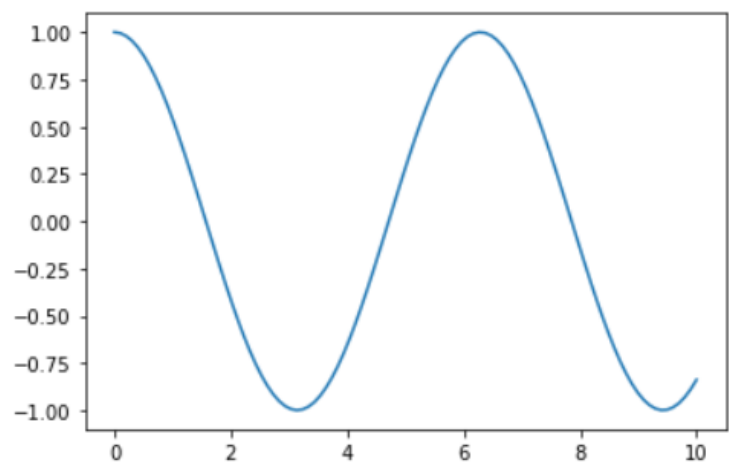
Basics

- **Smooth curves**

```
import numpy as np

x = np.linspace(0, 10, 100)

plt.plot(x, np.sin(x))
plt.show()
```




```
import numpy as np

x = np.linspace(0, 10, 100)

plt.plot(x, np.cos(x))
plt.show()
```



Styles

Styles

- **Attribute setup**

plt.xlim(최소값, 최대값) : 플롯의 x축 범위 지정

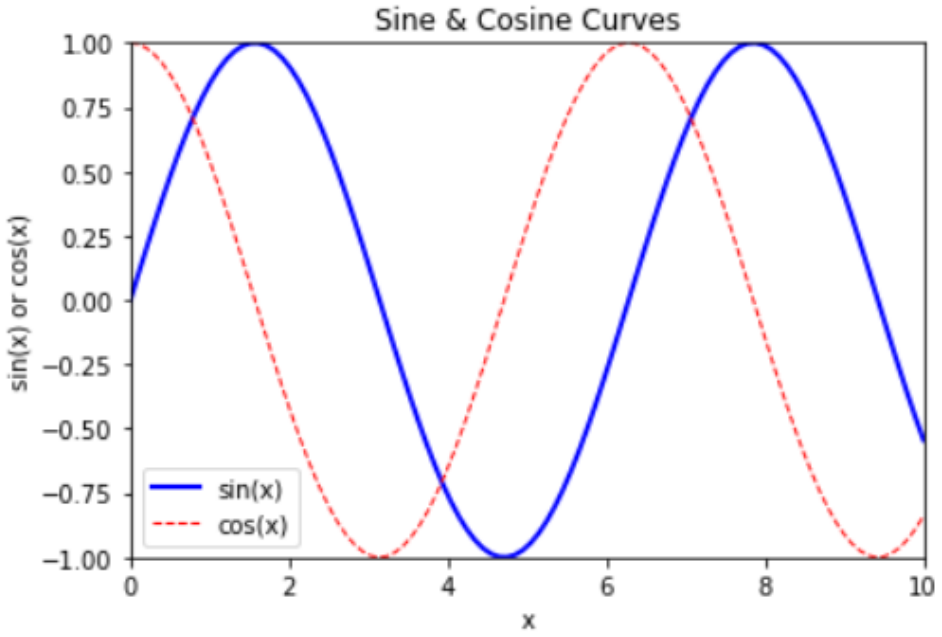
plt.ylim(최소값, 최대값) : 플롯의 y축 범위 지정

plt.title(문자열) : 플롯의 제목

plt.xlabel(문자열) / plt.ylabel(문자열) : 축 이름

plt.legend() : 범례 표시

```
import numpy as np

x =  np.linspace(0, 10, 100)

plt.plot(x, np.sin(x), 'b-', linewidth=2, label='sin(x)')
plt.plot(x, np.cos(x), 'r--', linewidth=1, label='cos(x)')
plt.xlim(0, 10)
plt.ylim(-1, 1)
plt.title('Sine & Cosine Curves')
plt.xlabel('x'), plt.ylabel('sin(x) or cos(x)')
plt.legend()
plt.show()
```



Styles

- **Styles**

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

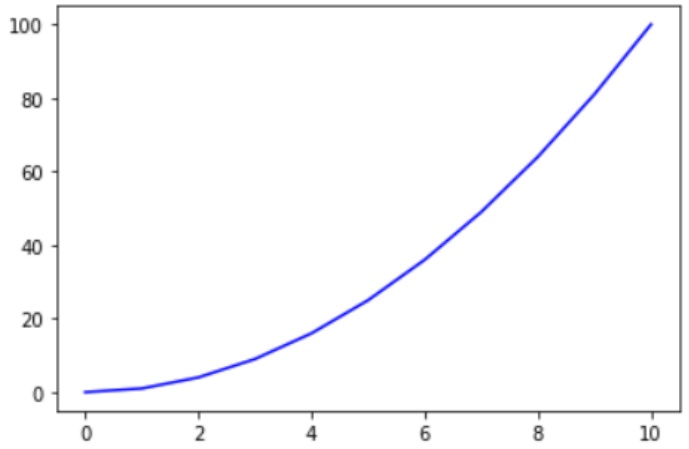
character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

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

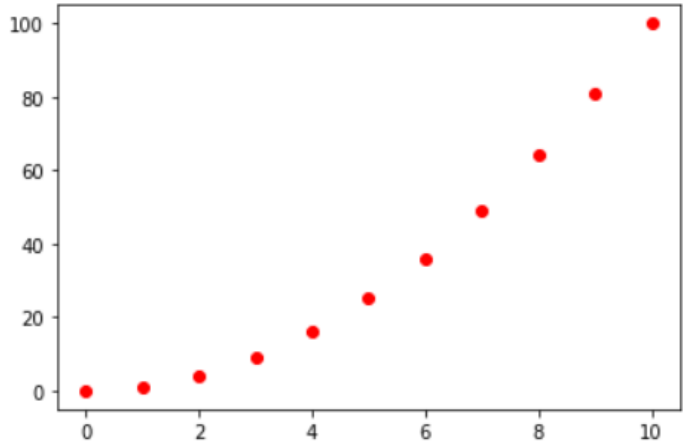
Styles

- **Styles**

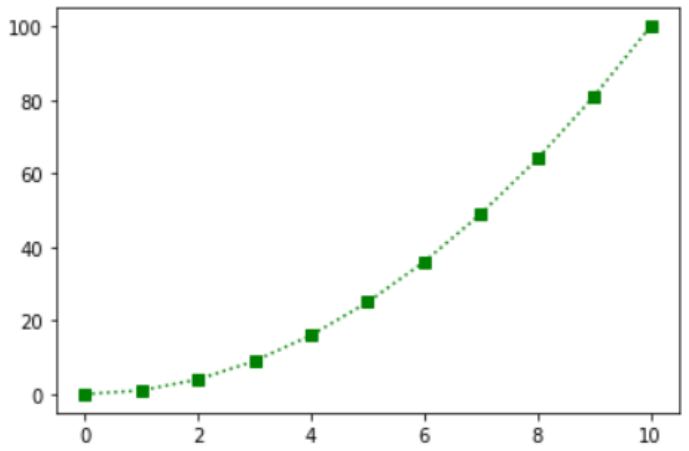
```
x = np.arange(11)

plt.plot(x, x**2, 'b')
plt.show()
```




```
plt.plot(x, x**2, 'ro')
plt.show()
```

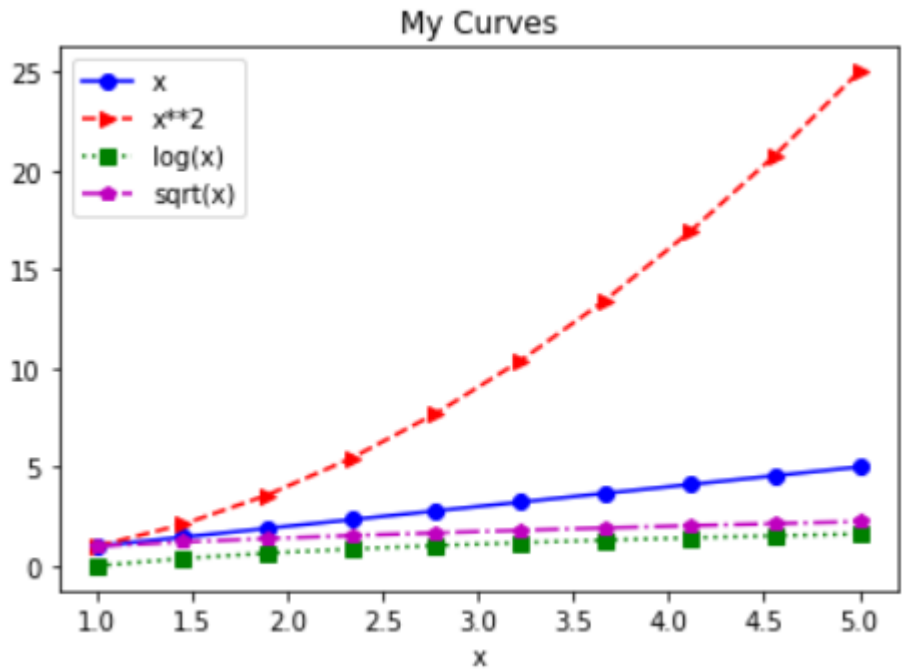



```
plt.plot(x, x**2, 'gs:')
plt.show()
```



Styles

- **Exercise**



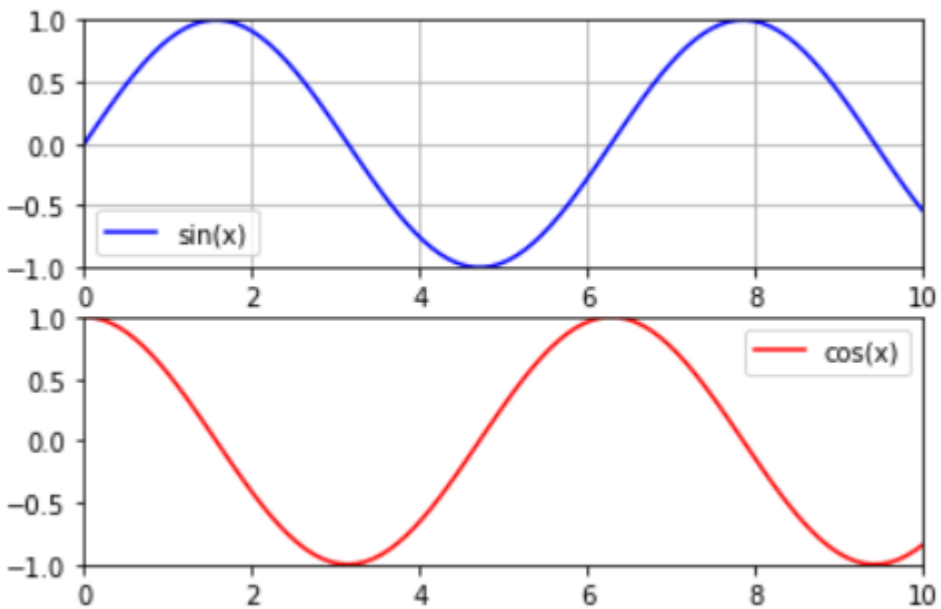

```
x =  np.linspace(1, 5, 10)

plt.plot(x, x, 'bo-', label='x')
plt.plot(x, x**2, 'r>--', label='x**2')
plt.plot(x, np.log(x), 'gs:', label='log(x)')
plt.plot(x, np.sqrt(x), 'mp-.', label='sqrt(x)')
plt.title('My Curves')
plt.xlabel('x')
plt.legend()
plt.show()
```

Subplots

Subplots

- **Subplots**

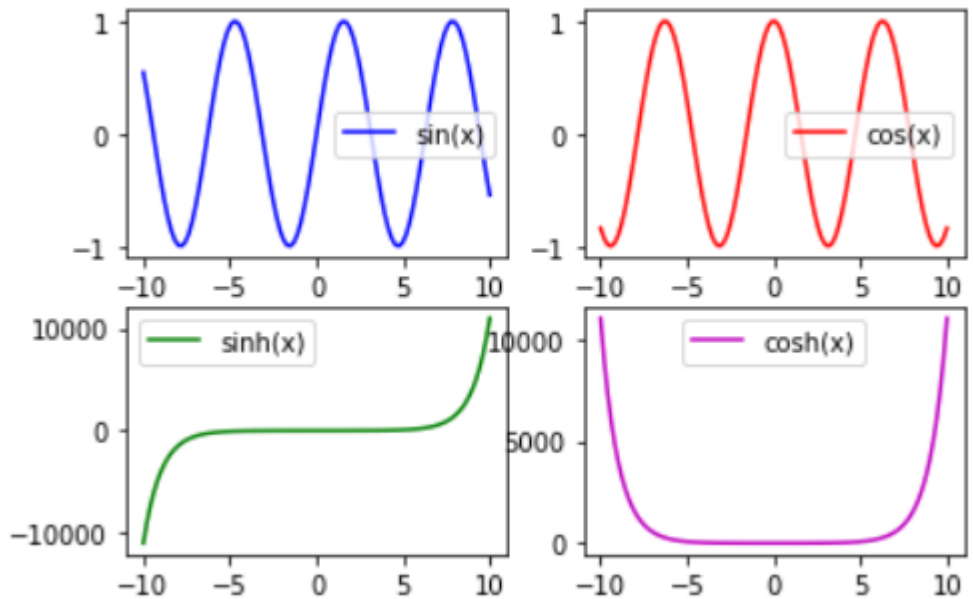
```
x = np.linspace(0, 10, 100)

plt.subplot(2, 1, 1)
plt.plot(x, np.sin(x), 'b-', label='sin(x)')
plt.axis([0, 10, -1, 1])
plt.grid(True)
plt.legend()

plt.subplot(2, 1, 2)
plt.plot(x, np.cos(x), 'r-', label='cos(x)')
plt.axis([0, 10, -1, 1])
plt.legend()
plt.show()
```



Subplots

- **Subplots**

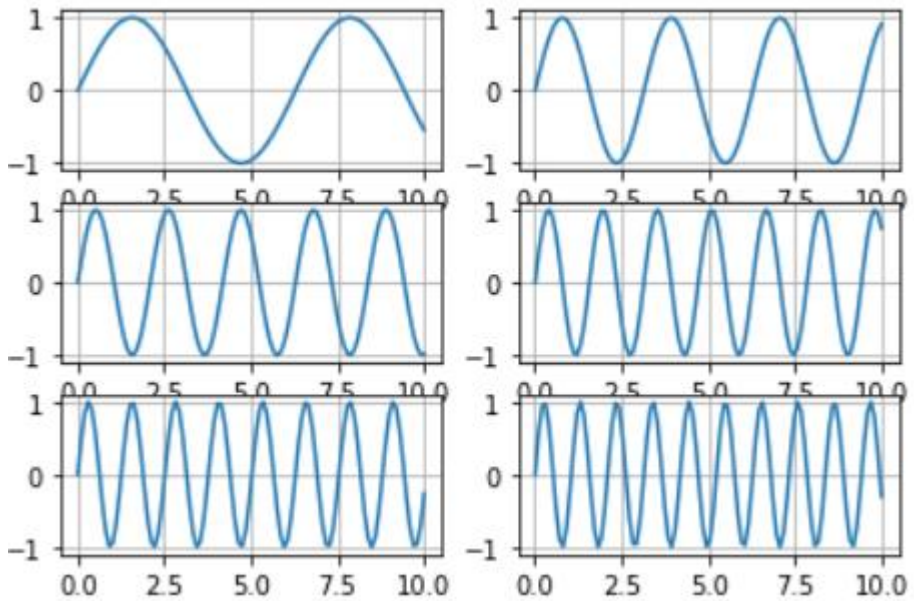
```
x = np.linspace(-10, 10, 100)

plt.subplot(2, 2, 1)
plt.plot(x, np.sin(x), 'b-', label='sin(x)')
plt.legend()

plt.subplot(2, 2, 2)
plt.plot(x, np.cos(x), 'r-', label='cos(x)')
plt.legend()


plt.subplot(2, 2, 3)
plt.plot(x, np.sinh(x), 'g-', label='sinh(x)')
plt.legend()


plt.subplot(2, 2, 4)
plt.plot(x, np.cosh(x), 'm-', label='cosh(x)')
plt.legend()
plt.show()
```



Subplots

- **Subplots & for loop**

```
x = np.linspace(0, 10, 100)

for i in range(1, 7):
  plt.subplot(3, 2, i)
  plt.plot(x, np.sin(i*x))
  plt.grid(True)
plt.show()
```

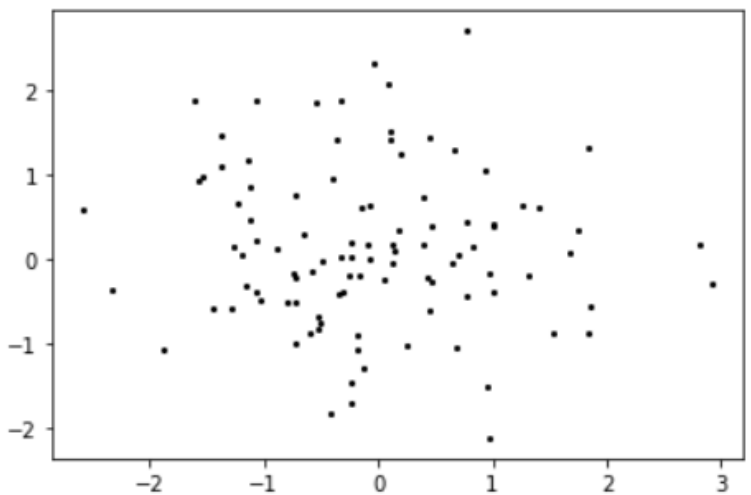


Scatter

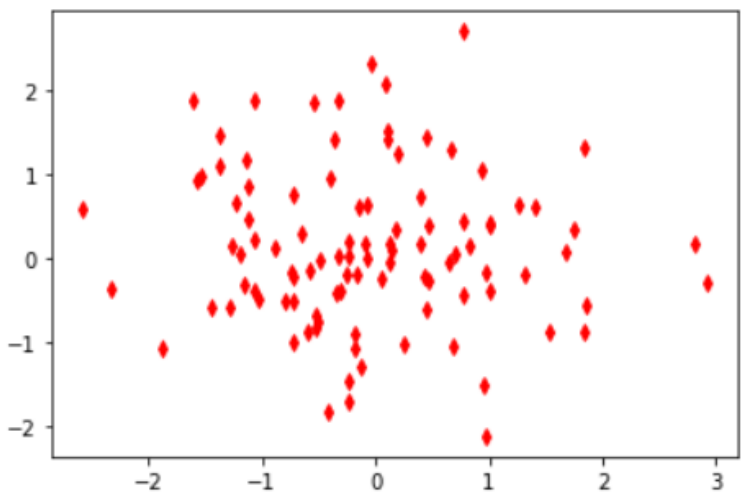
Scatter

- **Basics**

```
x = np.random.randn(100)
y = np.random.randn(100)

plt.plot(x, y, 'ko', ms=2)
plt.show()
```



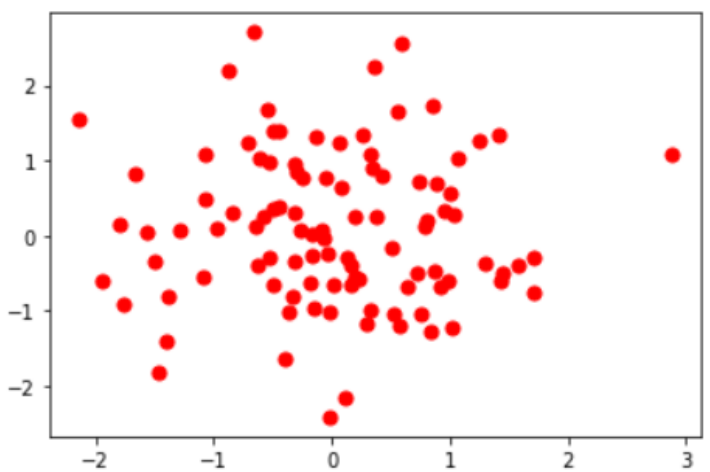

```
plt.plot(x, y, 'rd', ms=5)
plt.show()
```



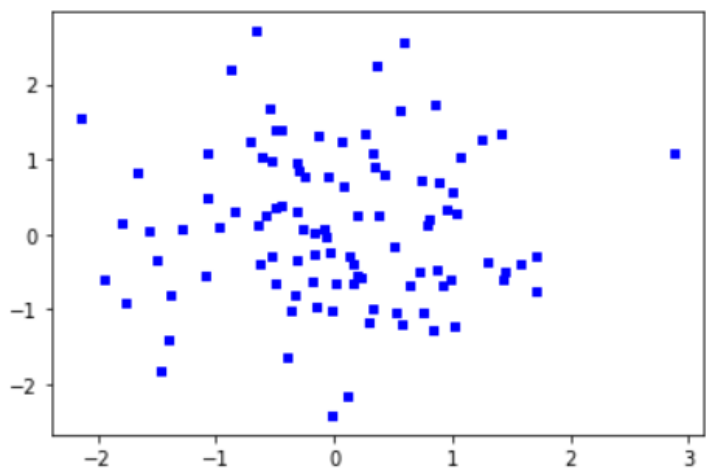
Scatter

- **Basics**

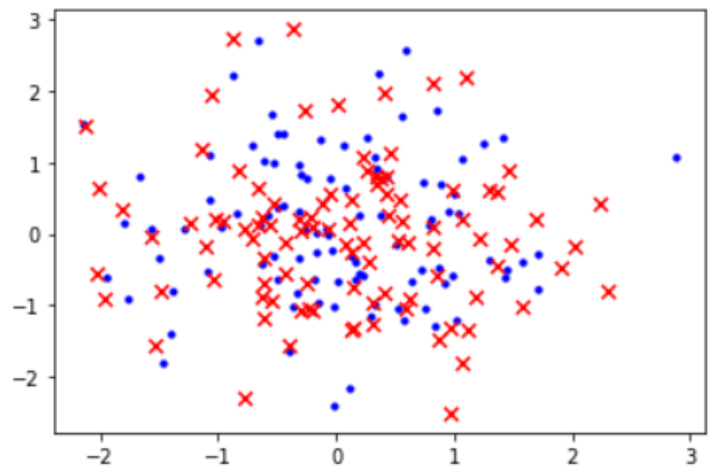
```
plt.scatter(x, y, c='r', s=50)
plt.show()
```




```
plt.scatter(x, y, c='b', s=10, marker='s')
plt.show()
```




```
z = np.random.randn(100)
w = np.random.randn(100)

plt.scatter(x, y, c='b', s=10, marker='o')
plt.scatter(z, w, c='r', s=50, marker='x')
plt.show()
```



Scatter

- **Exercise**

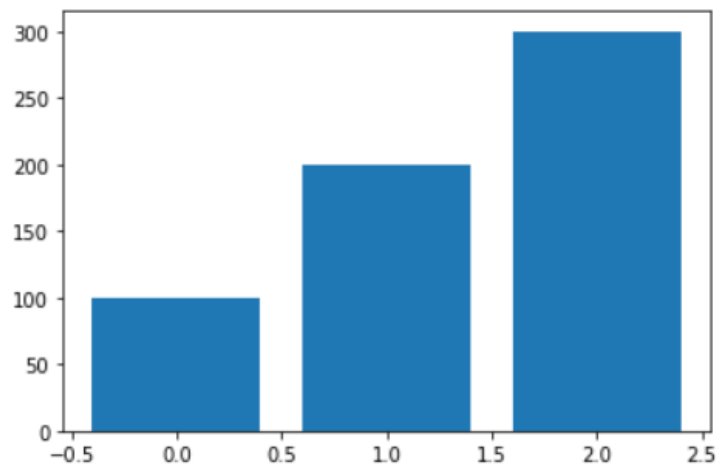

```
x = np.random.randn(20)
y = np.random.randn(20)

z = np.random.randn(20)+5
w = np.random.randn(20)+5

a = np.random.randn(20)-5
b = np.random.randn(20)-5

plt.scatter(x, y, c='b', s=10, marker='o', label='Group 1')
plt.scatter(z, w, c='r', s=50, marker='x', label='Group 2')
plt.scatter(a, b, c='g', s=30, marker='s', label='Group 3')
plt.plot(np.linspace(-5,5,10), np.linspace(-5,5,10), 'k-', label='y=x')
plt.grid(True)
plt.xlabel('x')
plt.ylabel('y')
plt.title('My Scatters')
plt.legend()
plt.show()
```

Bar

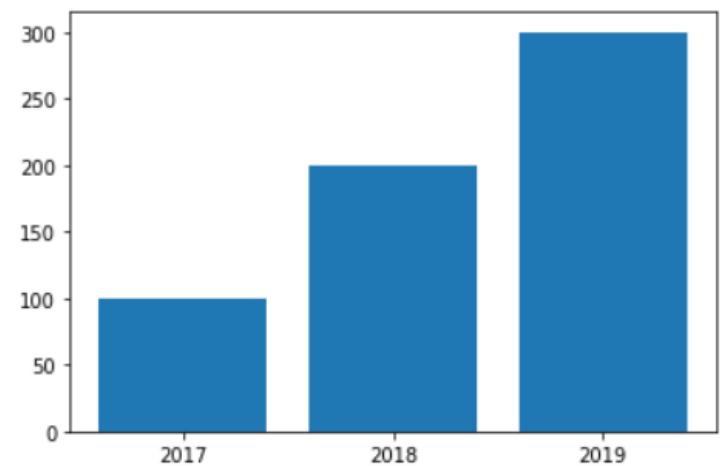
Bar

- **Basics**

```
x = np.arange(3)
y = [100, 200, 300]
plt.bar(x, y)
plt.show()
```



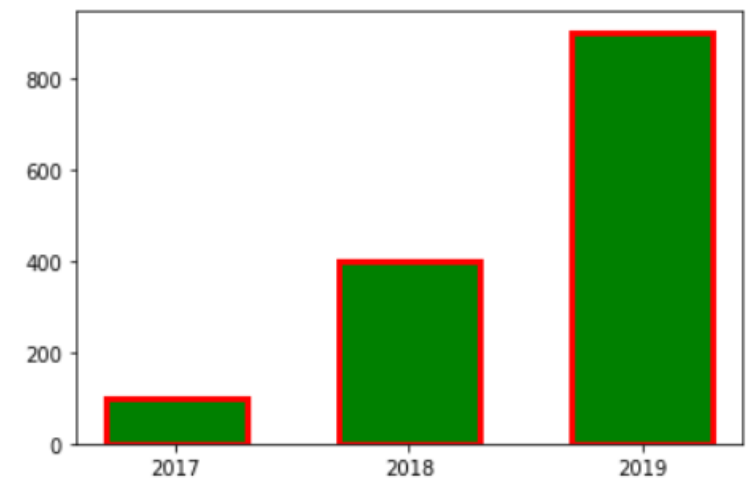

```
x = np.arange(3)
years = ['2017', '2018', '2019']
y = [100, 200, 300]

plt.bar(x, y)
plt.xticks(x, years)
plt.show()
```



Bar

- **Styles**
  - `width`: 막대의 너비
  - `color`: 막대의 색
  - `edgecolor`: 테두리 색
  - `linewidth`: 테두리 너비
  - `tick_label`: x축 tick

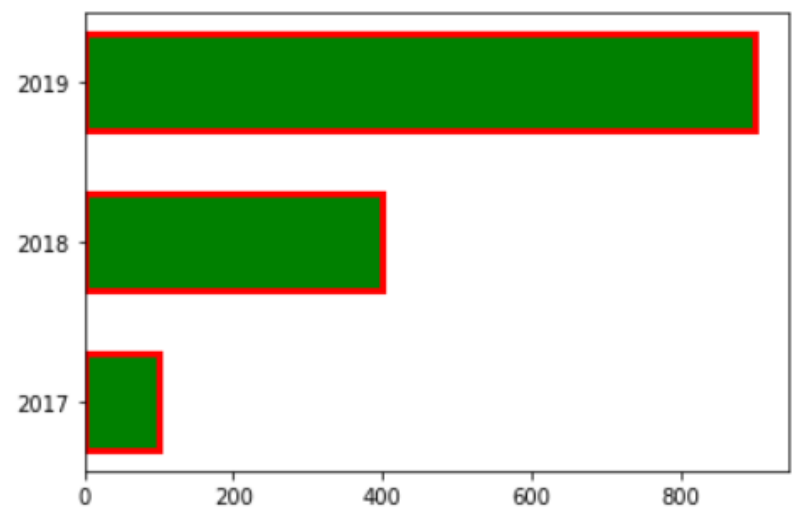
```
x = np.arange(3)
years = ['2017', '2018', '2019']
y = [100, 400, 900]

plt.bar(x, y, width=0.6, color="g", edgecolor="r", linewidth=3, tick_label=years)
plt.show()
```



Bar

- **Horizontal bar**

```
x = np.arange(3)
years = ['2017', '2018', '2019']
y = [100, 400, 900]

plt.barh(x, y, height=0.6, color="g", edgecolor="r", linewidth=3, tick_label=years)
plt.show()
```

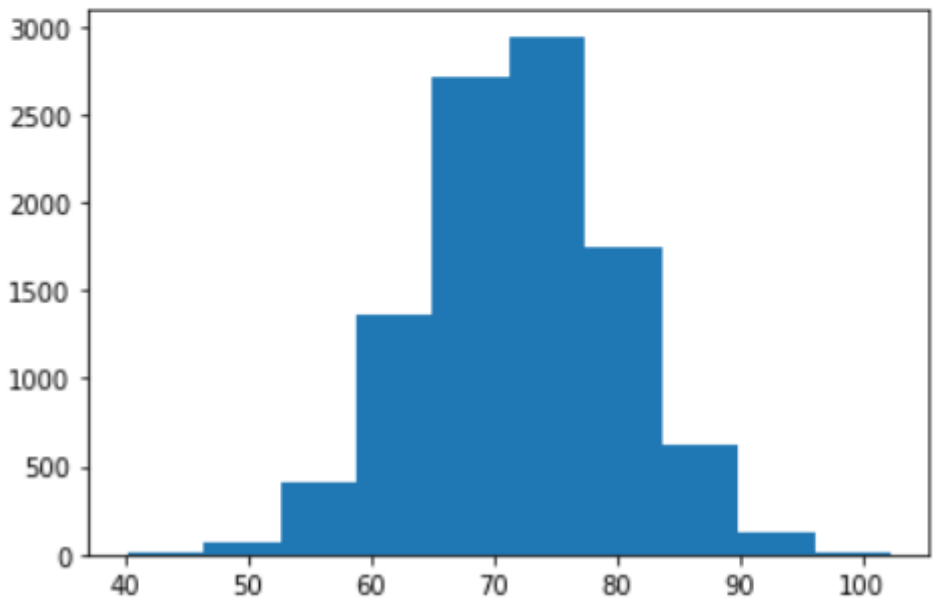


Histogram

Histogram

- **Basics**

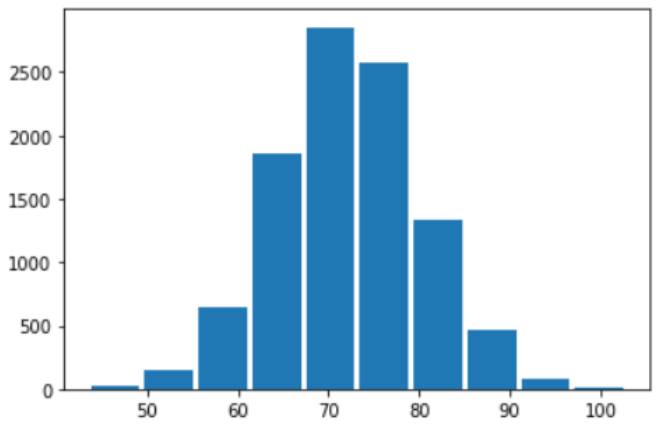
```
mu = 72
sigma = 8
x = np.random.normal(mu, sigma, 10000)

plt.hist(x)
plt.show()
```



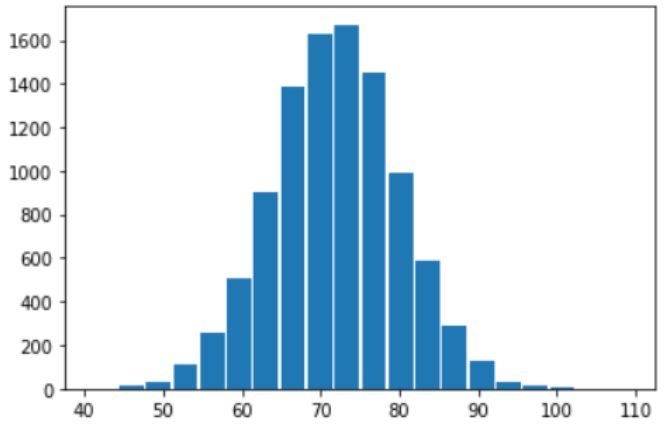
Histogram

- **Styles**
  - `rwidth`: 막대의 너비, `bins`: 계급 수/계급 범위

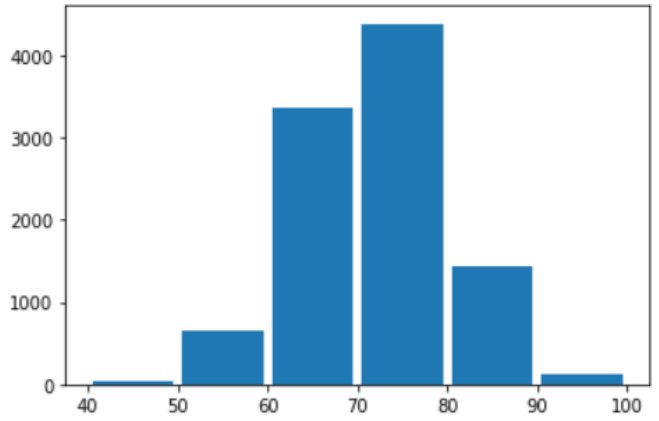
```
mu = 72
sigma = 8
x = np.random.normal(mu, sigma, 10000)

plt.hist(x, rwidth=0.9)
plt.show()
```




```
mu = 72
sigma = 8
x = np.random.normal(mu, sigma, 10000)

plt.hist(x, rwidth=0.9, bins = 20)
plt.show()
```




```
mu = 72
sigma = 8
x = np.random.normal(mu, sigma, 10000)

plt.hist(x, rwidth=0.9, bins = [40, 50, 60, 70, 80, 90, 100])
plt.show()
```



Summary

- **Matplotlib**
  - `plt.plot, plt.scatter, plt.hist, plt.bar, plt.barh`
  - `plt.xlim, plt.ylim`
  - `plt.title, plt.xlabel, plt.ylabel`
  - `plt.legend`