

## ✓ 1차시: 프로그래밍 기초 (2)

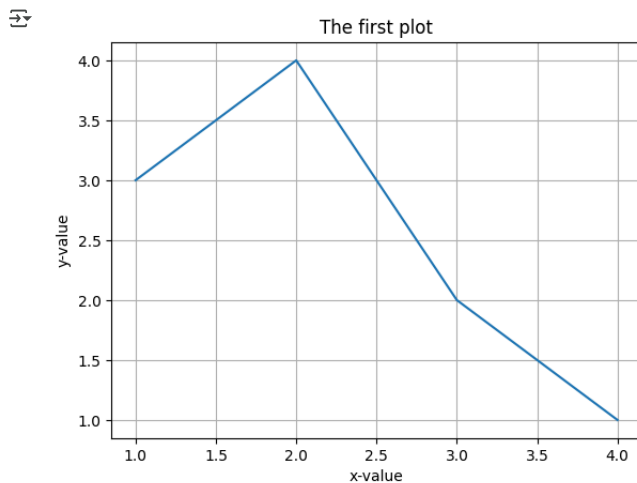
```
import numpy as np
import matplotlib.pyplot as plt
```

### Matplotlib 소개

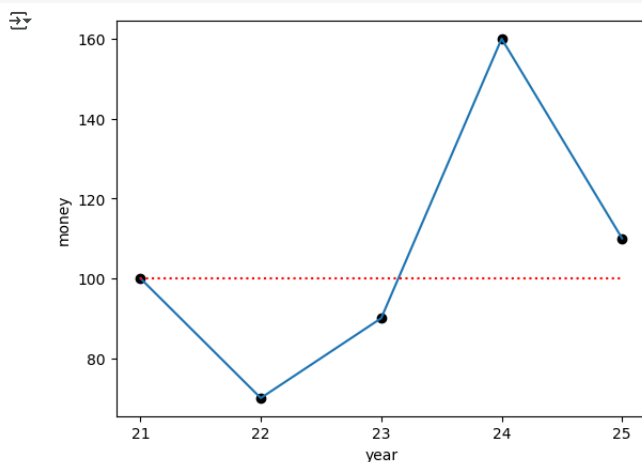
- 데이터 시각화를 위한 파이썬 라이브러리
  - Plot 종류 소개
1. 선 그래프(Line Plot): 시간 흐름 또는 순차적 데이터 표현
  2. 산점도(Scatter Plot): 두 변수 간 관계 탐색
  3. 막대 그래프(Bar Chart): 카테고리별 비교
  4. 히스토그램(Histogram): 데이터 분포 확인
  5. 상자 그림(Box Plot): 분포 요약 및 이상치 확인

### ✓ 선 그래프(Line Plot)

```
plt.plot([1,2,3,4],[3,4,2,1])
plt.xlabel('x-value')
plt.ylabel('y-value')
plt.title('The first plot')
plt.show()
```



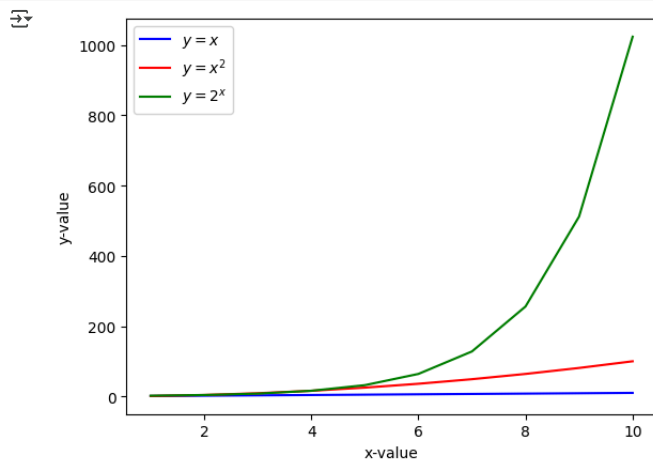
```
x = [21,22,23,24,25]
y = [100,70,90,160,110]
plt.plot(x,y)
plt.scatter(x,y, color='k')
plt.xticks(x)
plt.xlabel('year')
plt.ylabel('money')
plt.hlines(100, xmin=min(x), xmax=max(x), color='red', linestyle='dotted')
plt.show()
```



```
x = np.array([1,2,3,4,5,6,7,8,9,10])
y1 = x
y2 = x**2
y3 = 2**x

plt.plot(x, y1, color='blue', label='$y=x$')
plt.plot(x, y2, color='red', label='$y=x^2$')
plt.plot(x, y3, color='green', label='$y=2^x$')
plt.xlabel('x-value')
```

```
plt.ylabel('y-value')
plt.legend()
plt.show()
```

**실습 1:**

시간에 따라 변화하는 어떤 값에 대한 선 그래프를 그려보세요. (키, 몸무게, 성적 등)

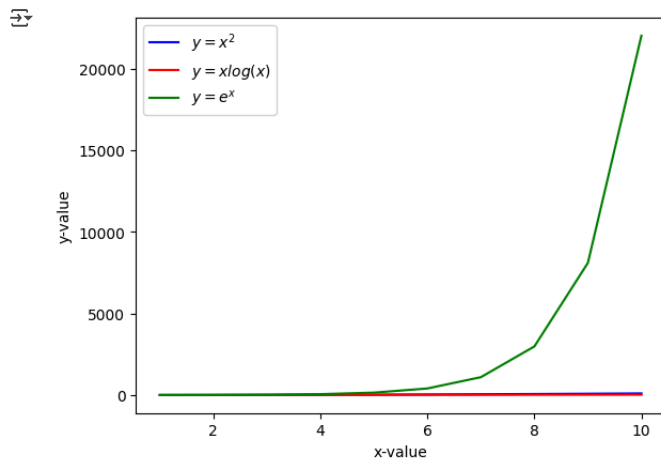
코딩을 시작하거나 AI로 코드를 생성하세요.

**실습 2:**

- $y = x^2$
- $y = x \log x$
- $y = e^x$  이 세 그래프를 그리시오.

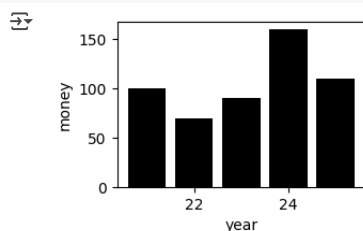
힌트:  $x$ 가 NumPy 배열 형식으로 정의해야 합니다. 로그함수  $\log x$ 는 `np.log(x)`를 사용하고, 자연상수  $e$ 를 밑으로 하는 지수함수  $e^x$ 는 `np.exp(x)`를 사용하면 됩니다.  $x$ 의 범위는 0부터 10으로 합니다.

코딩을 시작하거나 AI로 코드를 생성하세요.

**막대 그래프(Bar Chart)**

```
x = [21, 22, 23, 24, 25]
y = [100, 70, 90, 160, 110]
```

```
plt.figure(figsize=(3,2))
plt.bar(x,y, color='black')
plt.xlabel("year")
plt.ylabel("money")
plt.show()
```

**히스토그램(Histogram)**

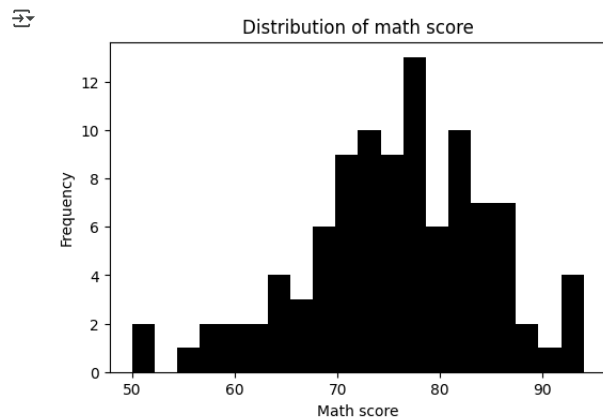
```
import pandas as pd
```

```
# 학생 점수 데이터 불러오기
df = pd.read_csv('student_scores.csv')
df
```

```
student  kor  math  eng
0        1   73    67   73
1        2   61    63   64
2        3   64    66   57
3        4   67    72   73
4        5   58    80   82
...      ...   ...   ...   ...
95       96   92    88   92
96       97   69    73   76
97       98   80    74   84
98       99   73    66   78
99      100   79    70   69
```

100 rows × 4 columns

```
plt.figure(figsize=(6,4))
plt.hist(df["eng"], bins=20, color='black')
plt.title("Distribution of math score")
plt.xlabel("Math score")
plt.ylabel("Frequency")
plt.show()
```



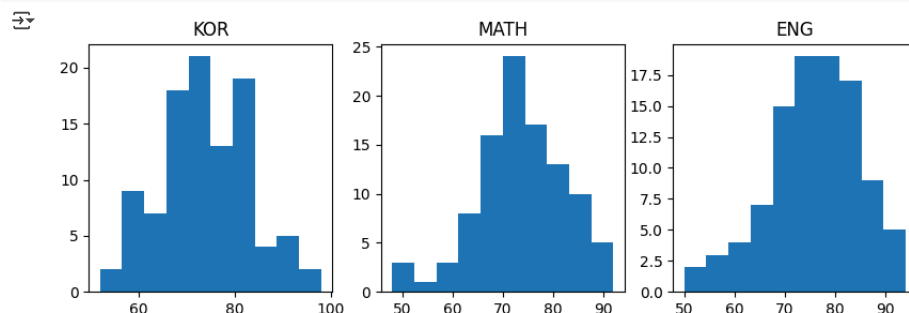
```
plt.figure(figsize=(10,3))
plt.subplot(131)
plt.hist(df["kor"], bins=10)
plt.title("KOR")

plt.subplot(132)
plt.hist(df["math"], bins=10)
plt.title("MATH")

plt.subplot(133)
plt.hist(df["eng"], bins=10)
plt.title("ENG")

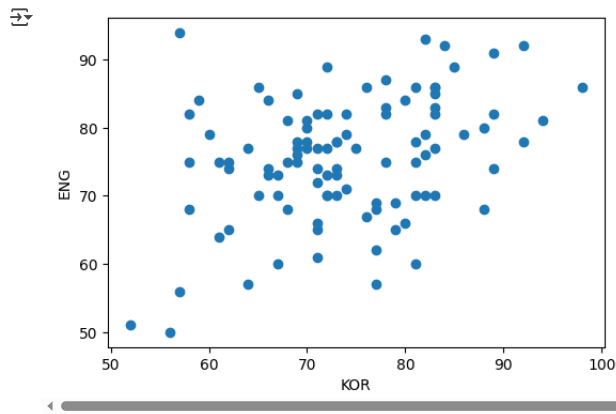
#plt.subplots_adjust(vspace=0.3)

plt.show()
```



## ▽ 산점도(Scatter Plot)

```
plt.figure(figsize=(6,4))
plt.scatter(df["kor"], df["eng"])
plt.xlabel('KOR')
plt.ylabel('ENG')
plt.show()
```



```
from scipy import stats
```

```
# 1. 데이터
x = df["kor"]
y = df["eng"]

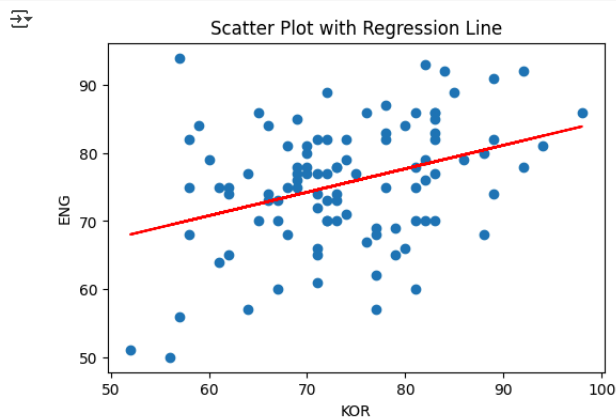
# 2. 선형 회귀 수행
# linregress는 기울기, y절편, R값, p값, 표준 오차를 반환
slope, intercept, r_value, p_value, std_err = stats.linregress(x, y)

# 3. 선형 회귀 직선 그리기
# 회귀 방정식: y_pred = slope * x + intercept
y_pred = slope * x + intercept

print(r_value)
```

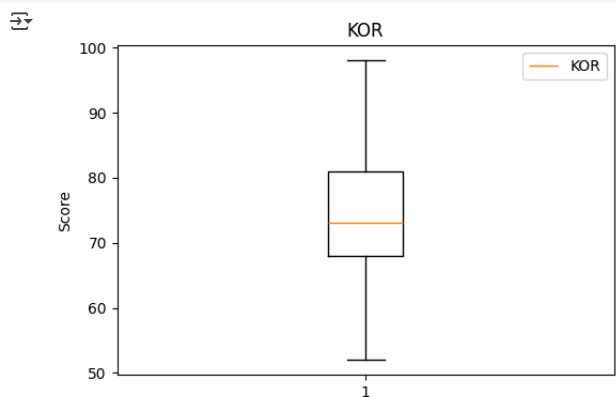
```
0.3581872369492512
```

```
plt.figure(figsize=(6,4))
plt.scatter(x, y)
plt.plot(x, y_pred, color='red')
plt.title('Scatter Plot with Regression Line')
plt.xlabel('KOR')
plt.ylabel('ENG')
plt.show()
```




## ▽ 상자 그림(Box Plot)

```
plt.figure(figsize=(6,4))
plt.boxplot(df["kor"], label="KOR")
plt.title("KOR")
plt.ylabel("Score")
plt.legend()
plt.show()
```



```
plt.figure(figsize=(6,4))
plt.boxplot([df["kor"], df["math"], df["eng"]], labels=["KOR","MATH","ENG"])
plt.show()
```

 <ipython-input-23-5adcb6daf101>:2: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick\_labels' since Matplotlib 3.9; support for the old name will be removed in the next major release.  
plt.boxplot([df["kor"], df["math"], df["eng"]], labels=["KOR","MATH","ENG"])

