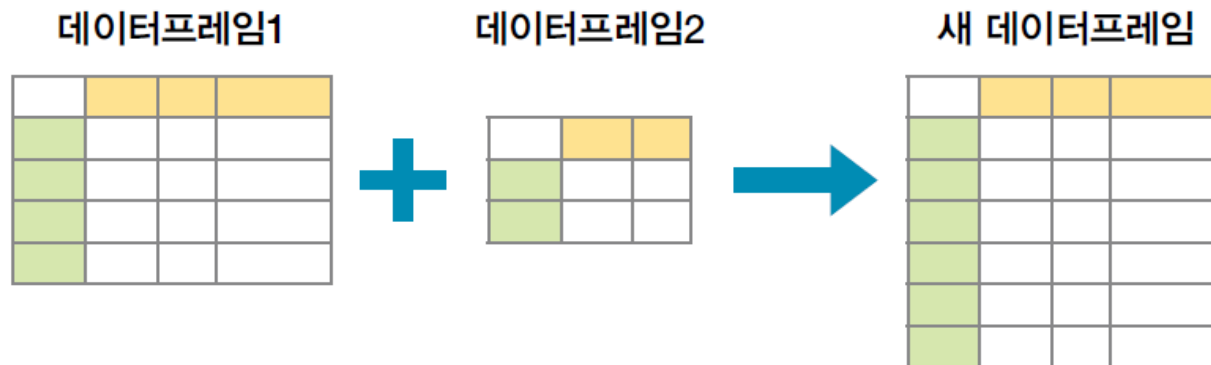


# 판다스와 맷플롯립

- 판다스의 개념과 사용법을 익힌다.
- 엑셀 없이 엑셀의 기능을 사용하는 방법을 학습한다.
- 그래프 출력을 위한 맷플롯립의 사용법을 익힌다.
- 판다스 및 맷플롯립을 활용한 앱을 작성한다.

## Section 01 이 장에서 만들 프로그램

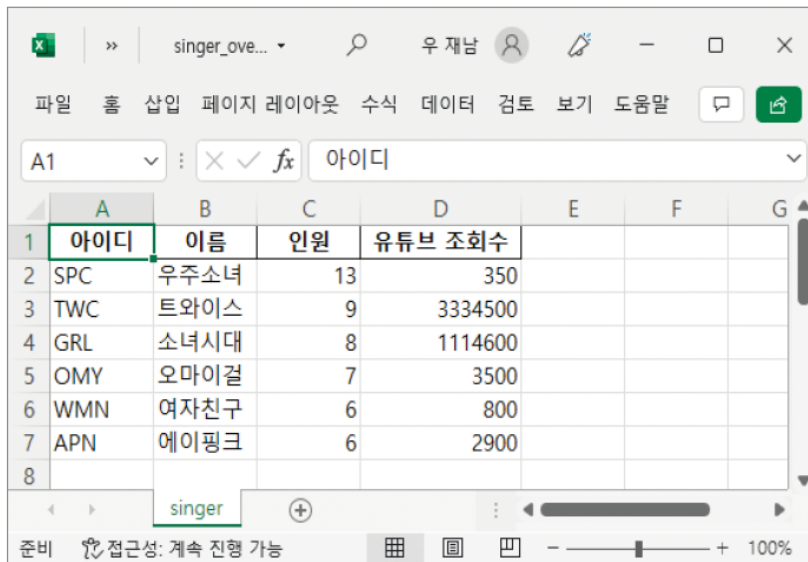
- [프로그램 1] 데이터프레임의 연산
  - 데이터프레임을 처리하기 위한 기본적인 방법을 익힘



# Section 01 이 장에서 만들 프로그램

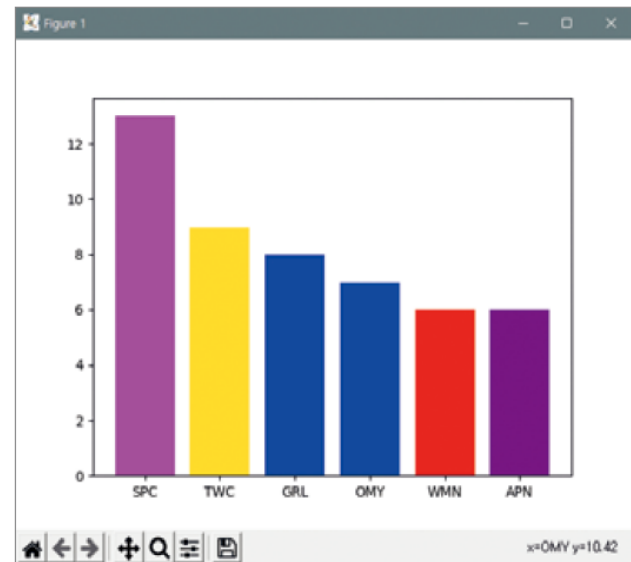
## ■ [프로그램 2] 엑셀 파일 처리와 출력

- 엑셀 파일을 판다스에서 처리하고 그 결과를 맷플롯립을 통해 그래프로 출력하는 프로그램



The screenshot shows an Excel spreadsheet with the following data:

아이디	이름	인원	유튜브 조회수
SPC	우주소녀	13	350
TWC	트와이스	9	3334500
GRL	소녀시대	8	1114600
OMY	오마이걸	7	3500
WMN	여자친구	6	800
APN	에이핑크	6	2900



## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 개념

- Panel Datas의 약자로 패널 자료를 처리한다는 뜻
- 엑셀의 워크시트를 처리하듯이 패널을 처리하는 기능이 통합된 라이브러리로 볼 수 있음
- 대개 넘파이 및 맷플롯립과 함께 사용됨



그림 11-1 엑셀과 같은 기능을 구현할 수 있는 판다스

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 데이터프레임

- 넘파이는 동일한 데이터 형식의 배열을 사용하지만, 판다스는 엑셀의 워크시트처럼 다양한 데이터 형식을 배열로 사용할 수 있음
- 따라서 배열(Array) 대신 데이터프레임(DataFrame)이라는 용어를 사용함

넘파이 배열					판다스 데이터프레임			
1	2	3	4	5		이름	나이	생일
6	7	8	9	10	1번	유정	30	5월 2일
11	12	13	14	15	2번	유나	28	4월 6일
16	17	18	19	20	3번	민영	31	9월 12일
21	22	23	24	25	4번	은지	29	7월 19일

그림 11-2 넘파이 배열과 판다스 데이터프레임 비교

## Section 02 판다스와 맷플롯립 기초

### ■ 데이터프레임 핵심 용어

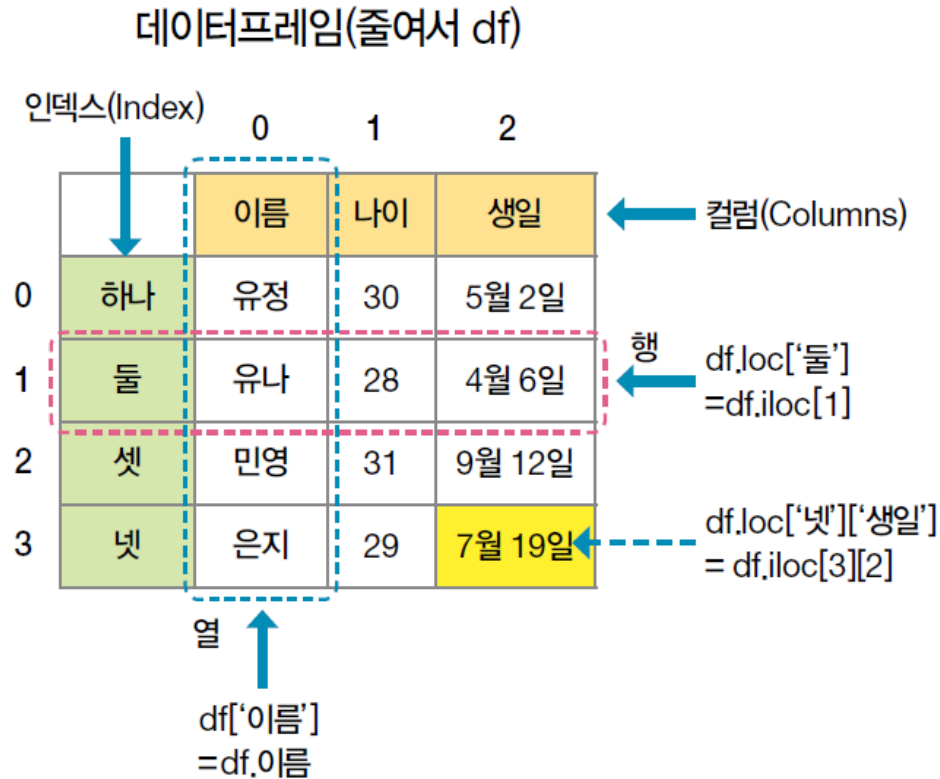


그림 11-3 데이터프레임 핵심 용어

## Section 02 판다스와 맷플롯립 기초

### ■ 데이터프레임 핵심 용어

- 행, 열, 셀의 접근 방식
  - 행 하나에 접근: `df.loc[인덱스이름]` 또는 `df.iloc[행번호]`
  - 열 하나에 접근: `df[열이름]` 또는 `df.열이름`
  - 셀 하나에 접근: `df.loc[인덱스이름, 열이름]` 또는 `df.iloc[행번호][열번호]`



## Section 02 판다스와 맷플롯립 기초

### ■ 시리즈

- 행 또는 열 하나만 추출한 것
- 인덱스와 데이터로 이루어짐

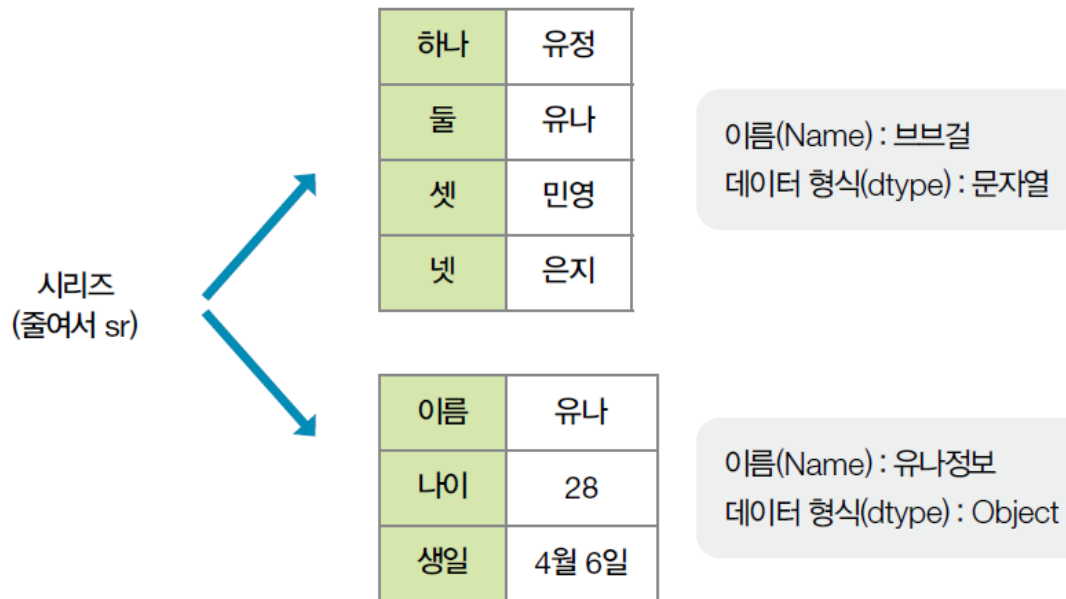


그림 11-4 시리즈 핵심 개념

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

- 외부 라이브러리 pandas 설치

```
pip install pandas
```

- 데이터프레임 만들기

- [그림 11-3]을 데이터프레임으로 그대로 구현

```
import pandas as pd
data = { '이름' : ['유정', '유나', '민영', '은지'],
        '나이' : [30, 28, 31, 29],
        '생일' : ['1991.5.2', '1993.4.6', '1990.9.12', '1992.7.19'] }
df1 = pd.DataFrame(data)
df1
```

#### 실행 결과

	이름	나이	생일
0	유정	30	1991.5.2
1	유나	28	1993.4.6
2	민영	31	1990.9.12
3	은지	29	1992.7.19

## Section 02 판다스와 맷플롯립 기초

- 판다스 사용 방법
  - 데이터프레임 만들기
    - 인덱스 별도 지정

```
df2 = pd.DataFrame(data, index=['하나', '둘', '셋', '넷'])  
df2
```

### 실행 결과

	이름	나이	생일
하나	유정	30	1991.5.2
둘	유나	28	1993.4.6
셋	민영	31	1990.9.12
넷	은지	29	1992.7.19

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

- 열과 행의 추출
  - df2에서 인덱스 및 컬럼 확인

```
df2.index  
df2.columns
```

#### 실행 결과

```
Index(['하나', '둘', '셋', '넷'], dtype='object')  
Index(['이름', '나이', '생일'], dtype='object')
```

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

- 열과 행의 추출
  - 열 추출

```
sr_name = df2['이름']  
sr_name  
type(sr_name)
```

#### 실행 결과

```
하나  유정  
둘    유나  
셋    민영  
넷    은지  
Name: 이름, dtype: object  
  
<class 'pandas.core.series.Series'>
```

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 열과 행의 추출

##### ■ name 별도 지정

```
sr_name.name = '브브걸'  
sr_name
```

##### 실행 결과

```
하나  유정  
둘    유나  
셋    민영  
넷    은지  
Name: 브브걸, dtype: object
```

##### ■ 인덱스가 '둘'인 행 추출

```
sr_two = df2.loc['둘']  
type(sr_two)  
sr_two
```

##### 실행 결과

```
<class 'pandas.core.series.Series'>  
  
이름    유나  
나이   28  
생일   1993.4.6  
Name: 둘, dtype: object
```

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 하나의 값 추출

##### ■ 추출 방법

- `df2.loc[인덱스이름][컬럼이름]` 또는 `df2.loc[인덱스이름, 컬럼이름]`
- `df2.iloc[행번호][열번호]` 또는 `df2.iloc[행번호, 열번호]`

```
df2.loc['넷']['생일']  
df2.loc['넷', '생일']  
df2.iloc[3][2]  
df2.iloc[3,2]
```

##### 실행 결과

```
'1992.7.19.'  
'1992.7.19.'  
'1992.7.19.'  
'1992.7.19'
```

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 열 추가

##### ■ 추가 방법 - 기본

- `df[새 열이름] = [값1, 값2 ...]`

```
df2['키'] = [163, 165, 168, 166]
```

#### 실행 결과

	이름	나이	생일	키
하나	유정	30	1991.5.2	163
둘	유나	28	1993.4.6	165
셋	민영	31	1990.9.12	168
넷	은지	29	1992.7.19	166



## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 열 추가

- 추가 방법 – 순서를 다르게 지정하고 싶거나, 데이터의 일부만 있는 경우
  - `pd.Series( [값...], index = [인덱스....])`

```
sr_vision = pd.Series( [1.8 , 0.9 , 1.2], index =['셋', '하나', '넷'] )  
df2['시력'] = sr_vision  
df2
```

#### 실행 결과

	이름	나이	생일	키	시력
하나	유정	30	1991.5.2	163	0.9
둘	유나	28	1993.4.6	165	NaN
셋	민영	31	1990.9.12	168	1.8
넷	은지	29	1992.7.19	166	1.2

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 열 추가

- 추가 방법 – 열을 중간에 삽입하는 경우
  - `df.insert(위치, 열이름, [값1, 값2...])`

```
df2.insert(1, '꽃', [ '장미', '백합', '튤립', '데이지'])  
df2
```

#### 실행 결과

	이름	꽃	나이	생일	키	시력
하나	유정	장미	30	1991.5.2	163	0.9
둘	유나	백합	28	1993.4.6	165	NaN
셋	민영	튤립	31	1990.9.12	168	1.8
넷	은지	데이지	29	1992.7.19	166	1.2

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 열 추가

- 추가 방법 – 열을 중간에 삽입하면서 순서까지 지정하는 경우
  - `df.insert(위치, 시리즈)`
- 추가 방법 – 제일 뒤에 열을 삽입하는 경우
  - `df.assign(열이름=[값1, 값2...])`

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 행 추가

##### ■ 추가 방법 – 기본

- `df.loc[새인덱스이름] = ([값1, 값2 ...])`

```
df2.loc['다섯'] = ['재남', '들꽃', 33, '1988.8.8', 177, 1.1]  
df2
```

#### 실행 결과

	이름	꽃	나이	생일	키	시력
하나	유정	장미	30	1991.5.2	163	0.9
둘	유나	백합	28	1993.4.6	165	NaN
셋	민영	튤립	31	1990.9.12	168	1.8
넷	은지	데이지	29	1992.7.19	166	1.2
다섯	재남	들꽃	33	1988.8.8	177	1.1

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 행 추가

##### ■ 추가 방법 – 기본

- 딕셔너리를 사용하여 {열이름1:값1, 열이름2:값2 ...} 방식으로 행을 추가할 때, 필요한 열에만 삽입할 수도 있음
- 생략된 값은 역시 NaN이 들어감

```
df2.loc['여섯'] = { '이름': '보라', '꽃': '민들레', '키': 163, '나이': 34 }  
df2
```

#### 실행 결과

	이름	꽃	나이	생일	키	시력
하나	유정	장미	30	1991.5.2	163	0.9
둘	유나	백합	28	1993.4.6	165	NaN
셋	민영	tulip	31	1990.9.12	168	1.8
넷	은지	데이지	29	1992.7.19	166	1.2
다섯	재남	들꽃	33	1988.8.8	177	1.1
여섯	보라	민들레	34	NaN	163	NaN

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 행 추가

##### ■ 추가 방법 – 여러 행을 한번에 추가

- `pd.concat( [old_df, new_df] )`

```
new_data = { '이름' : ['리사', '제니'], '나이' : [23, 22] }  
new_df = pd.DataFrame(new_data, index=['블핑', '블핑'])  
new_df  
df2 = pd.concat( [ df2, new_df] )
```

#### 실행 결과

	이름	나이
블핑	리사	23
블핑	제니	22

	이름	꽃	나이	생일	키	시력
하나	유정	장미	30	1991.5.2	163.0	0.9
둘	유나	백합	28	1993.4.6	165.0	NaN
셋	민영	tulip	31	1990.9.12	168.0	1.8
넷	은지	데이지	29	1992.7.19	166.0	1.2
다섯	재남	들꽃	33	1988.8.8	177.0	1.1
여섯	보라	민들레	34	NaN	163.0	NaN
블핑	리사	NaN	23	NaN	NaN	NaN
블핑	제니	NaN	22	NaN	NaN	NaN

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 열과 행 삭제

##### ■ 삭제 방법 - 기본

- axis의 1은 열, 0은 행을 의미
- 열 삭제: `df.drop([열이름], axis=1)`
- 행 삭제: `df.drop([인덱스이름], axis=0)`

```
df2 = df2.drop(['키'], axis=1)
df2 = df2.drop(['셋'], axis=0)
df2
```

#### 실행 결과

	이름	꽃	나이	생일	시력
하나	유정	장미	30	1991.5.2	0.9
둘	유나	백합	28	1993.4.6	NaN
넷	은지	데이지	29	1992.7.19	1.2
다섯	재남	들꽃	33	1988.8.8	1.1
여섯	보라	민들레	34	NaN	NaN
블핑	리사	NaN	23	NaN	NaN
블핑	제니	NaN	22	NaN	NaN

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### ■ 열과 행 삭제

- 삭제 방법 – 여러 열 또는 행을 삭제
  - 열 삭제: axis=1 또는 axis='columns' 지정
  - 행 삭제: axis를 생략

```
df2 = df2.drop( ['꽃', '시력'], axis=1 )
df2
df2 = df2.drop( ['블핑', '하나'] )
df2
```

#### 실행 결과

	이름	꽃	나이	생일	시력
하나	유정	장미	30	1991.5.2	0.9
둘	유나	백합	28	1993.4.6	NaN
넷	은지	데이지	29	1992.7.19	1.2
다섯	재남	들꽃	33	1988.8.8	1.1
여섯	보라	민들레	34	NaN	NaN
블핑	리사	NaN	23	NaN	NaN
블핑	제니	NaN	22	NaN	NaN

	이름	나이	생일
둘	유나	28	1993.4.6
넷	은지	29	1992.7.19
다섯	재남	33	1988.8.8
여섯	보라	34	NaN



## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

#### SELF STUDY 11-1

다음 표를 데이터프레임으로 생성하고 열과 행을 추가 및 삭제해 보자.

	이름	인원	데뷔 일자
WMN	여자친구	6	2015.01.15
GRL	소녀시대	8	2007.08.02
RED	레드벨벳	4	2014.08.01
APN	에이핑크	6	2011.02.10
MMU	마마무	4	2014.06.19



	인원	데뷔 일자
WMN	6	2015.01.15
MMU	4	2014.06.19
ABC	1	2023.03.03

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

- 시리즈 연산
  - 두 시리즈의 인덱스가 동일한 경우에는 그냥 연산하면 됨
  - 두 시리즈의 인덱스가 다를 때는 주의가 필요함

Code11-01.py

```
01     import pandas as pd
02
03     sr1 = pd.Series( [ 10, 20, 30, 40], index = [ '다현', '정연', '쯔위', '사나' ] )
04     sr2 = pd.Series( [ 50, 60, 70, 80], index = [ '다현', '정연', '쯔위', '사나' ] )
05     sr3 = pd.Series( [ 11, 22, 33, 44], index = [ '다현', '사나', '모모', '재남' ] )
06
07     sr12 = sr1 + sr2
08     print(sr12, '\n')
09
10     sr13 = sr1 + sr3
11     print(sr13)
```

## Section 02 판다스와 맷플롯립 기초

### ■ 판다스 사용 방법

- 시리즈 연산
  - 두 시리즈의 인덱스가 동일한 경우에는 그냥 연산하면 됨
  - 두 시리즈의 인덱스가 다를 때는 주의가 필요함

#### 실행 결과

```
다현    60
정연    80
쫘위   100
사나   120
dtype: int64
```

```
다현    21.0
모모    NaN
사나    62.0
재남    NaN
정연    NaN
쫘위    NaN
dtype: float64
```

## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 그래프

- 맷플롯립 개념
  - 그래프나 차트를 생성할 때 사용하는 라이브러리
  - '데이터 시각화'를 위한 라이브러리
  - 넘파이나 판다스에서 처리된 결과를 시각적으로 표현할 때 추가로 사용하는 라이브러리
  - 막대 그래프, 선 그래프, 원 그래프, 히스토그램, 산점도 등 통계 그래프 및 지도를 그릴 때 편리함

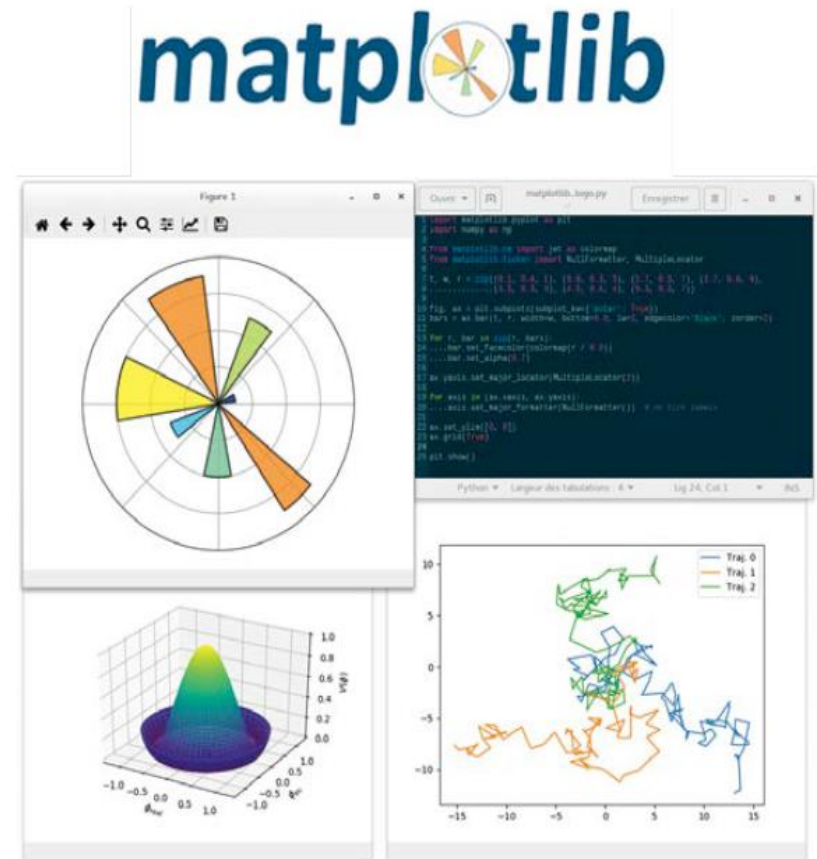


그림 11-5 맷플롯립 로고와 시각화한 그래프(출처: 위키피디아)

## Section 02 판다스와 맷플롯립 기초

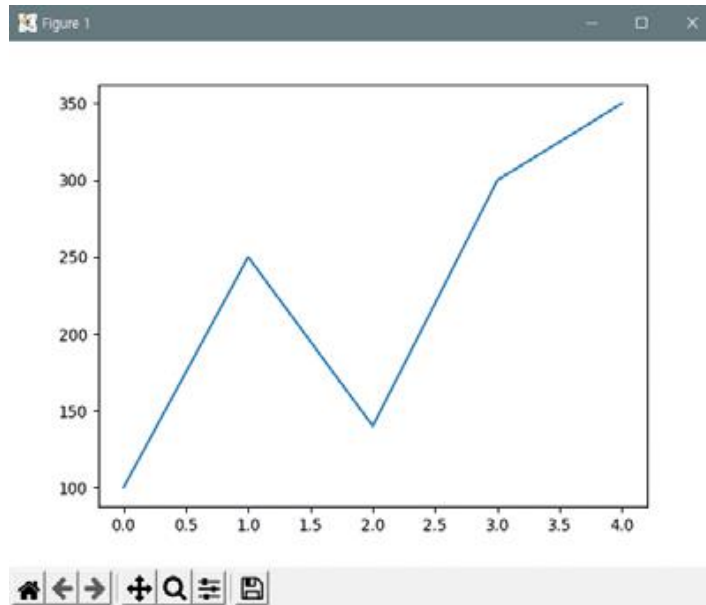
### ■ 맷플롯립 사용 방법

- 외부 라이브러리 matplotlib 설치

```
pip install matplotlib
```

- 예시 코드

```
import matplotlib.pyplot as plt  
data = [100, 250, 140, 300, 500]  
plt.plot(data)  
plt.show()
```



## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 사용 방법

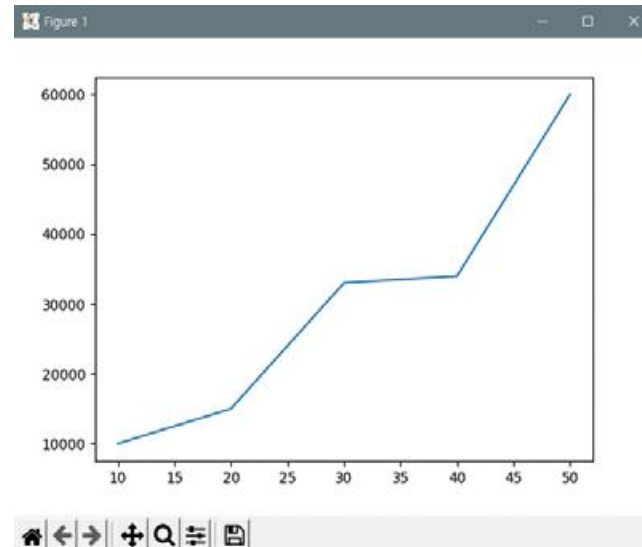
- X축과 Y축의 값을 지정해서 출력

표 11-1 거리에 따른 택시비

거리	10km	20km	30km	40km	50km
택시비	10000	15000	33000	34000	60000



```
x_data = [ 10, 20, 30, 40, 50 ]  
y_data = [10000, 15000, 33000, 34000, 60000]  
plt.plot( x_data, y_data )  
plt.show()
```



## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 사용 방법

#### ■ 그래프 스타일 지정

표 11-2 매개변수 지정 방법

(a) linestyle

종류	지정 방법
실선(Solid)	'—'
대쉬선(Dashed)	'--'
점선(Dotted)	'.'
대쉬점선(Dash-dot)	'-.'

(b) marker

종류	지정 방법
o	'o'
x	'x'
+	'+'
사각형	's'
오각형	'p'
마름모	'd'
영문자	'\$영문자\$'

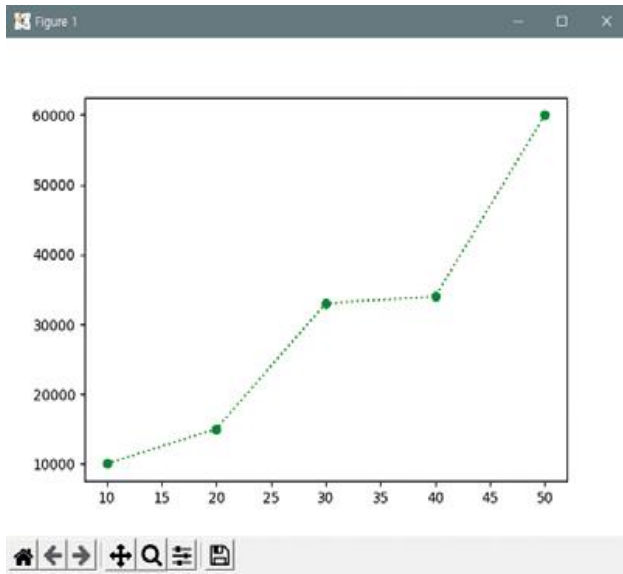
## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 사용 방법

#### ■ 그래프 스타일 지정

- 선 색상을 초록색으로, 선 모양을 점선으로 변경
- 선과 선 사이에 원 표시를 넣어서 값이 명확하게 보이도록 변경

```
plt.plot(x_data, y_data, color='green', linestyle=':', marker='o')  
plt.show()
```



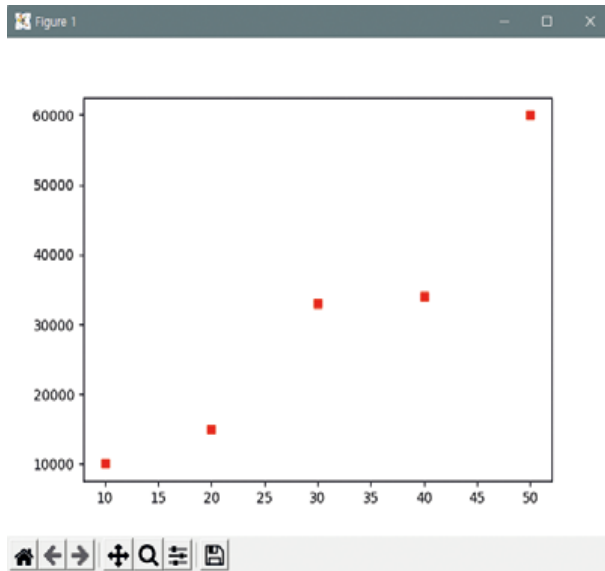


## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 사용 방법

- 그래프 스타일 지정
  - 선을 표시하지 않고 빨간색으로 사각형 마커를 사용

```
plt.plot(x_data, y_data, 'rs')  
plt.show()
```



## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 사용 방법

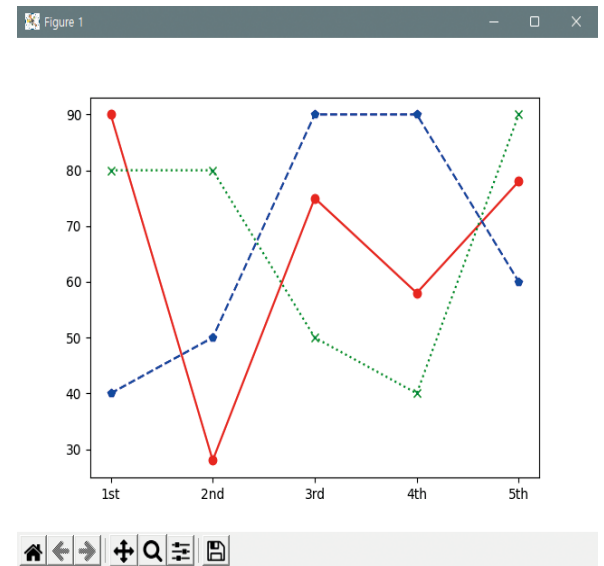
#### ■ 여러 개의 선 그리기

표 11-3 시험회차에 따른 3명의 성적

시험회차	1st	2nd	3rd	4th	5th
다현	90	82	75	58	78
정연	80	80	50	40	90
모모	40	50	90	90	60



```
x_data = ['1st', '2nd', '3rd', '4th', '5th']  
y1_data = [ 90, 28, 75, 58, 78]  
y2_data = [ 80, 80, 50, 40, 90]  
y3_data = [ 40, 50, 90, 90, 60]  
plt.plot( x_data, y1_data, 'r-o', x_data, y2_data, 'g:x', x_data, y3_data, 'b--p')  
plt.show()
```



## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 사용 방법

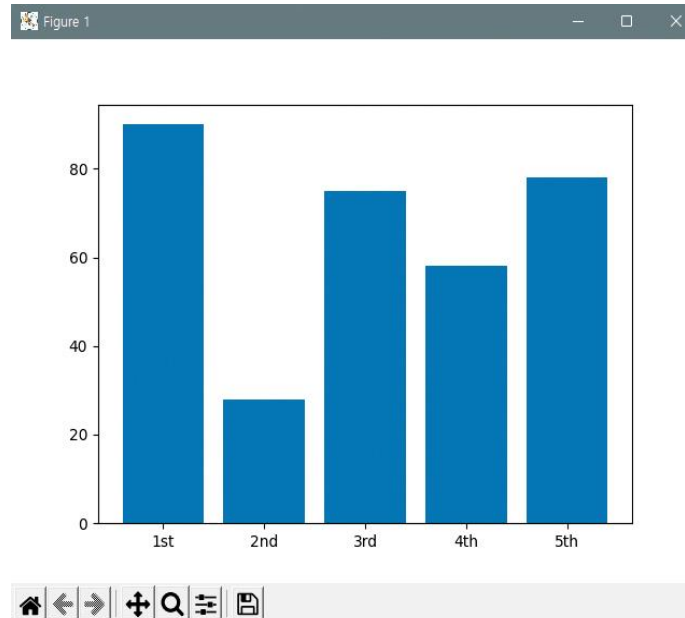
#### ■ 막대 그래프 그리기

표 11-3 시험회차에 따른 3명의 성적

시험회차	1st	2nd	3rd	4th	5th
다현	90	82	75	58	78
정연	80	80	50	40	90
모모	40	50	90	90	60



```
plt.bar(x_data, y1_data)  
plt.show()
```



## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 사용 방법

- 여러 개의 막대 그래프 그리기
  - x축의 데이터가 숫자여야 처리하기 편하며 넘파이 배열을 사용해야 코드가 단순해짐

#### Code11-02.py

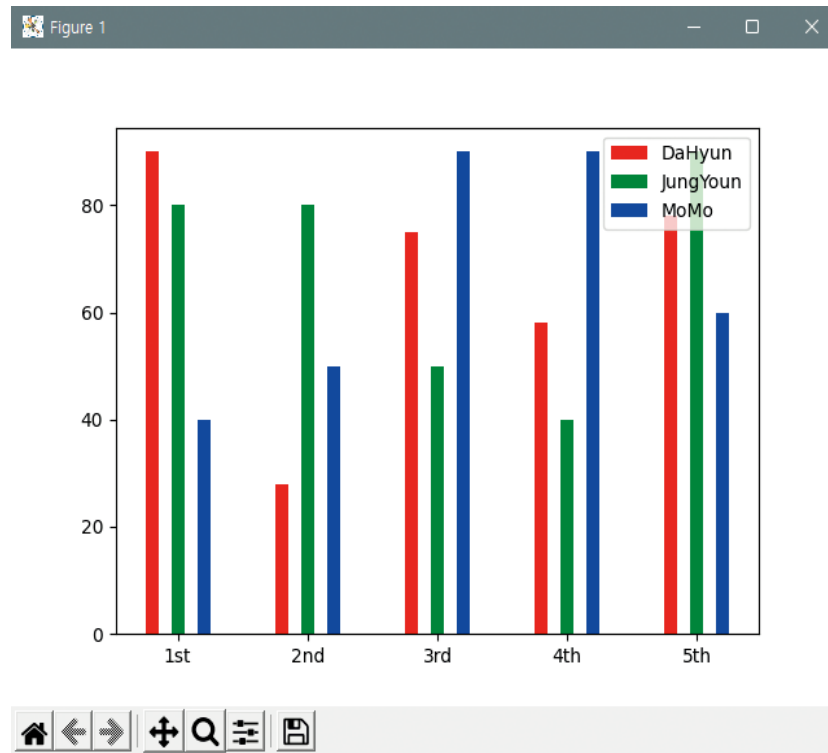
```
01 import numpy as np
02 import matplotlib.pyplot as plt
03
04 x_data = ['1st', '2nd', '3rd', '4th', '5th' ]
05 x_value = np.array([1, 2, 3, 4, 5])
06 y1_data = np.array([ 90, 28, 75, 58, 78])
07 y2_data = np.array([ 80, 80, 50, 40, 90])
08 y3_data = np.array([ 40, 50, 90, 90, 60])
09
10 plt.bar( x_value, y1_data, color='red', width=0.1, label='DaHyun')
11 plt.bar( x_value+0.2, y2_data, color='green', width=0.1, label='JungYoun')
12 plt.bar( x_value+0.4, y3_data, color='blue', width=0.1, label='MoMo')
13 plt.xticks(x_value+0.2, x_data)
14 plt.legend(loc='upper right')
15 plt.show()
```

## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 사용 방법

#### ■ 여러 개의 막대 그래프 그리기

- x축의 데이터가 숫자여야 처리하기 편하며 넘파이 배열을 사용해야 코드가 단순해짐



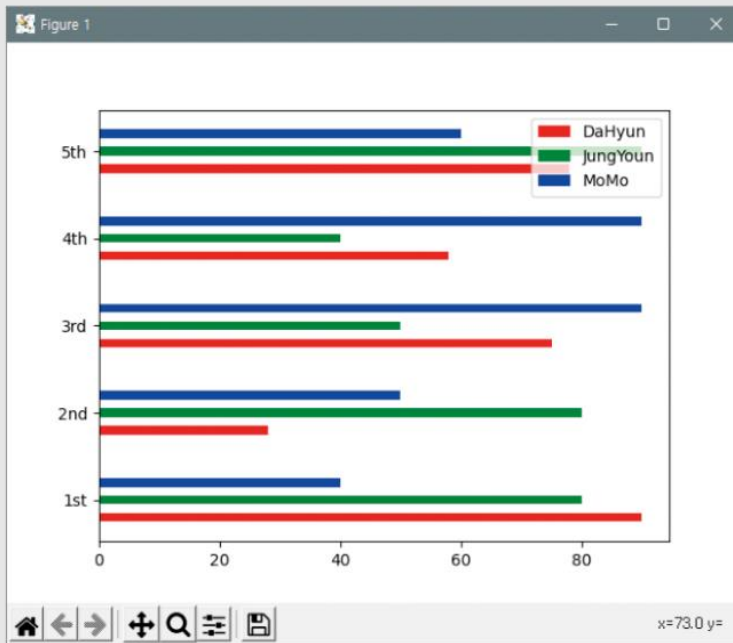
## Section 02 판다스와 맷플롯립 기초

- 맷플롯립 사용 방법
  - 여러 개의 막대 그래프 그리기

### SELF STUDY 11-2

Code 11-02.py를 참조해서 가로로 막대 그래프를 그려보자.

**힌트** plt.barh()를 사용하고 옵션은 width를 사용한다. 또 plt.xticks() 대신에 plt.yticks()를 사용한다.



## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 사용 방법

#### ■ 산점도 그리기

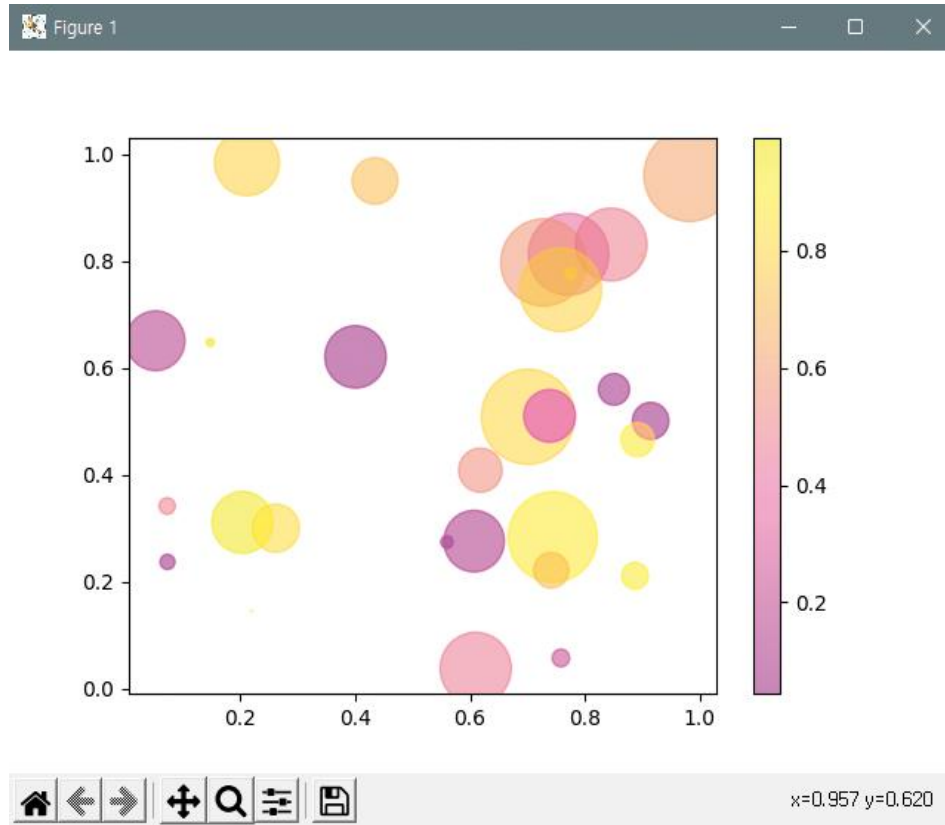
##### Code11-03.py

```
01 import numpy as np
02 import matplotlib.pyplot as plt
03
04 SIZE = 30
05 x_value = np.random.rand(SIZE)
06 y_value = np.random.rand(SIZE)
07 sizeAry = (50 * np.random.rand(SIZE))**2
08 colorAry = np.random.rand(SIZE)
09
10 plt.scatter(x_value, y_value, s=sizeAry, c=colorAry, alpha = 0.5, cmap='spring')
11 plt.colorbar()
12 plt.show()
```

## Section 02 판다스와 맷플롯립 기초

### ■ 맷플롯립 사용 방법

#### ■ 산점도 그리기





## Section 02 판다스와 맷플롯립 기초

### ■ [프로그램 1] 완성

Code11-04.py

```
01 import pandas as pd
02 import numpy as np
03
04 data1 = np.arange(9).reshape((3,3))
05 data2 = np.arange(12).reshape((4,3))
06 df1 = pd.DataFrame(data1, columns=list('가나다'), index=['서울', '부산', '광주'])
07 df2 = pd.DataFrame(data2, columns=list('가다라'), index=['고양', '서울', '광주', '대전'])
08 print(df1, '\n')
09 print(df2, '\n')
10
11 newDf = df1 + df2
12 print(newDf)
```

## Section 02 판다스와 맷플롯립 기초

### ■ [프로그램 1] 완성

실행 결과				
	가	나	다	
서울	0	1	2	
부산	3	4	5	
광주	6	7	8	
	가	다	라	
고양	0	1	2	
서울	3	4	5	
광주	6	7	8	
대전	9	10	11	
	가	나	다	라
고양	NaN	NaN	NaN	NaN
광주	12.0	NaN	15.0	NaN
대전	NaN	NaN	NaN	NaN
부산	NaN	NaN	NaN	NaN
서울	3.0	NaN	6.0	NaN

## Section 03 판다스와 맷플롯립 활용

### ■ CSV와 엑셀 데이터 처리

- 판다스에서 CSV 파일을 데이터프레임으로 읽어서 데이터를 정렬한 후 출력하는 코드

표 11-4 singer2.csv 데이터

아이디	이름	인원	주소	평균 키	데뷔 일자	유튜브 조회수
TWC	트와이스	9	서울	167	2015.10.19	3,334,500
BLK	블랙핑크	4	경남	163	2016.08.08	443,700
WMN	여자친구	6	경기	166	2015.01.15	800
OMY	오마이걸	7	서울	160	2015.04.21	3,500
GRL	소녀시대	8	서울	168	2007.08.02	1,114,600
ITZ	있지	5	경남	167	2019.02.12	21,300
RED	레드벨벳	4	경북	161	2014.08.01	44,500
APN	에이핑크	6	경기	164	2011.02.10	2,900
SPC	우주소녀	13	서울	162	2016.02.25	350
MMU	마마무	4	전남	165	2014.06.19	6,900

## Section 03 판다스와 맷플롯립 활용

### ■ CSV와 엑셀 데이터 처리

- 판다스에서 CSV 파일을 데이터프레임으로 읽어서 데이터를 정렬한 후 출력하는 코드

#### Code11-05.py

```
01 import pandas as pd
02
03 filename = 'C:/CookAnalysis/CSV/singer2.csv'
04 df = pd.read_csv(filename, index_col=None, encoding='CP949')
05
06 df_sort1 = df.sort_values(by=['평균 키'], axis=0)
07 print(df_sort1)
```

## Section 03 판다스와 맷플롯립 활용

### ■ CSV와 엑셀 데이터 처리

- 판다스에서 CSV 파일을 데이터프레임으로 읽어서 데이터를 정렬한 후 출력하는 코드

#### 실행 결과

	아이디	이름	인원	주소	평균 키	데뷔 일자	유튜브 조회수
3	OMY	오마이걸	7	서울	160	2015.04.21	3,500
6	RED	레드벨벳	4	경북	161	2014.08.01	44,500
8	SPC	우주소녀	13	서울	162	2016.02.25	350
1	BLK	블랙핑크	4	경남	163	2016.08.08	443,700
7	APN	에이핑크	6	경기	164	2011.02.10	2,900
9	MMU	마마무	4	전남	165	2014.06.19	6,900
2	WMN	여자친구	6	경기	166	2015.01.15	800
0	TWC	트와이스	9	서울	167	2015.10.19	3,334,500
5	ITZ	있지	5	경남	167	2019.02.12	21,300
4	GRL	소녀시대	8	서울	168	2007.08.02	1,114,600

## Section 03 판다스와 맷플롯립 활용

### ■ [프로그램 2] 완성

- 엑셀 파일의 워크시트 여러 개를 읽어서 하나의 데이터프레임으로 읽는 코드

표 11-5 singer.xlsx 파일의 워크시트 2개

(a) senior

아이디	이름	인원	주소	평균 키	데뷔 일자	유튜브 조회수
WMN	여자친구	6	경기	166	2015.01.15	800
GRL	소녀시대	8	서울	168	2007.08.02	1,114,600
RED	레드벨벳	4	경북	161	2014.08.01	44,500
APN	에이핑크	6	경기	164	2011.02.10	2,900
MMU	마마무	4	전남	165	2014.06.19	6,900

(b) junior

아이디	이름	인원	주소	평균 키	데뷔 일자	유튜브 조회수
TWC	트와이스	9	서울	167	2015.10.19	3,334,500
BLK	블랙핑크	4	경남	163	2016.08.08	443,700
OMY	오마이걸	7	서울	160	2015.04.21	3,500
ITZ	있지	5	경남	167	2019.02.12	21,300
SPC	우주소녀	13	서울	162	2016.02.25	350

## Section 03 판다스와 맷플롯립 활용

### ■ [프로그램 2] 완성

- 엑셀 파일의 워크시트 여러 개를 읽어서 하나의 데이터프레임으로 읽는 코드

Code11-06.py

```
01 import pandas as pd
02 import numpy as np
03 import matplotlib.pyplot as plt
04
05 inFilename = 'C:/CookAnalysis/Excel/singer.xlsx'
06 outFilename = 'C:/CookAnalysis/Excel/singer_over6.xlsx'
07
08 df_seniro = pd.read_excel(inFilename, 'senior', index_col=None)
09 df_junior = pd.read_excel(inFilename, 'junior', index_col=None)
10
11 df_singer = pd.concat( [df_seniro, df_junior] )
12 df_singer_over6 = df_singer[df_singer['인원'] >= 6]
13 df_singer_over6 = df_singer_over6.sort_values(by=['인원'], axis=0,
14
```

## Section 03 판다스와 맷플롯립 활용

### ■ [프로그램 2] 완성

- 엑셀 파일의 워크시트 여러 개를 읽어서 하나의 데이터프레임으로 읽는 코드

Code11-06.py

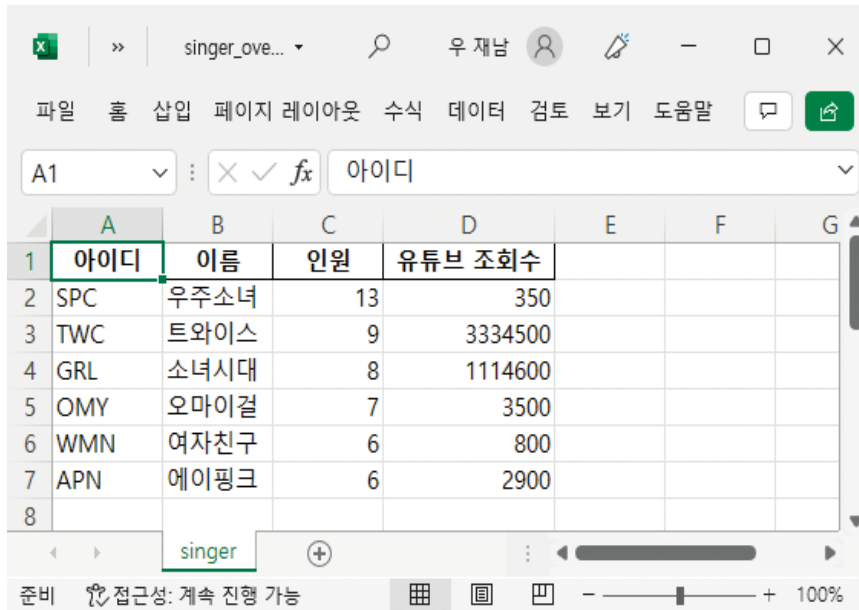
```
15 df_singer_over6 = df_singer_over6.loc[:, ['아이디', '이름', '인원', '유튜브 조회수']]
16
17 x_data = df_singer_over6['아이디']
18 y_data = df_singer_over6['인원']
19 colorAry = [ np.random.choice(['red', 'green', 'blue', 'brown', 'gold', 'lime', \
20                                'aqua', 'magenta', 'purple']) for _ in
21 plt.bar(x_data, y_data, color=colorAry)
22 plt.show()
23
24 writer = pd.ExcelWriter(outFilename)
25 df_singer_over6.to_excel(writer, sheet_name='singer', index=False)
26 writer.save()
27 print('Save. OK~')
```



## Section 03 판다스와 맷플롯립 활용

### ■ [프로그램 2] 완성

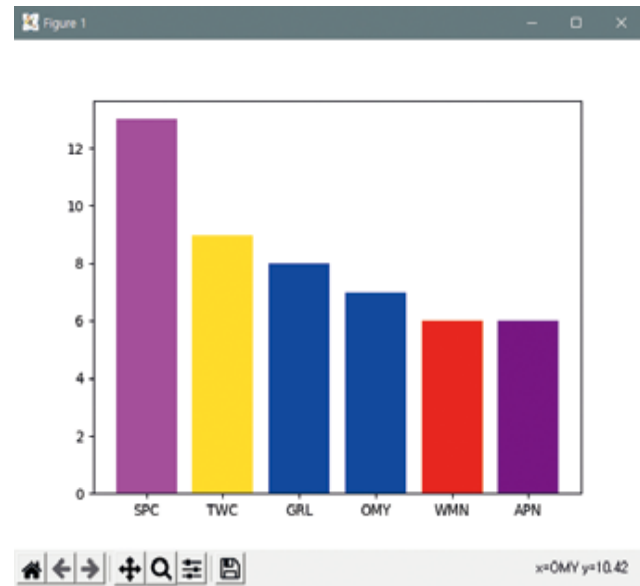
- 엑셀 파일의 워크시트 여러 개를 읽어서 하나의 데이터프레임으로 읽는 코드



The screenshot shows an Excel spreadsheet with a table containing singer information. The columns are labeled '아이디' (ID), '이름' (Name), '인원' (Members), and '유튜브 조회수' (YouTube Views). The data rows are as follows:

	A	B	C	D	E	F	G
1	아이디	이름	인원	유튜브 조회수			
2	SPC	우주소녀	13	350			
3	TWC	트와이스	9	3334500			
4	GRL	소녀시대	8	1114600			
5	OMY	오마이걸	7	3500			
6	WMN	여자친구	6	800			
7	APN	에이핑크	6	2900			
8							

The bottom of the window shows the 'singer' worksheet tab is selected.



## Section 03 판다스와 맷플롯립 활용

### ■ [프로그램 2] 완성

#### SELF STUDY 11-3

Code11-06.py를 참고하여 유튜브 조회수가 높은 가수별로 정렬한 결과를 출력한 후 산점도로 그려보자. 산점도의 원의 크기는 유튜브 조회수를 제곱근해서 처리하자.

