

# 빅데이터 실습

10주차 1차시

Pandas로 데이터 시각화하기



Goal

## ➤ Matplotlib을 통한 데이터 시각화



대표적인 시각화 라이브러리



안정적으로 다양하고 파워풀한 기능 제공



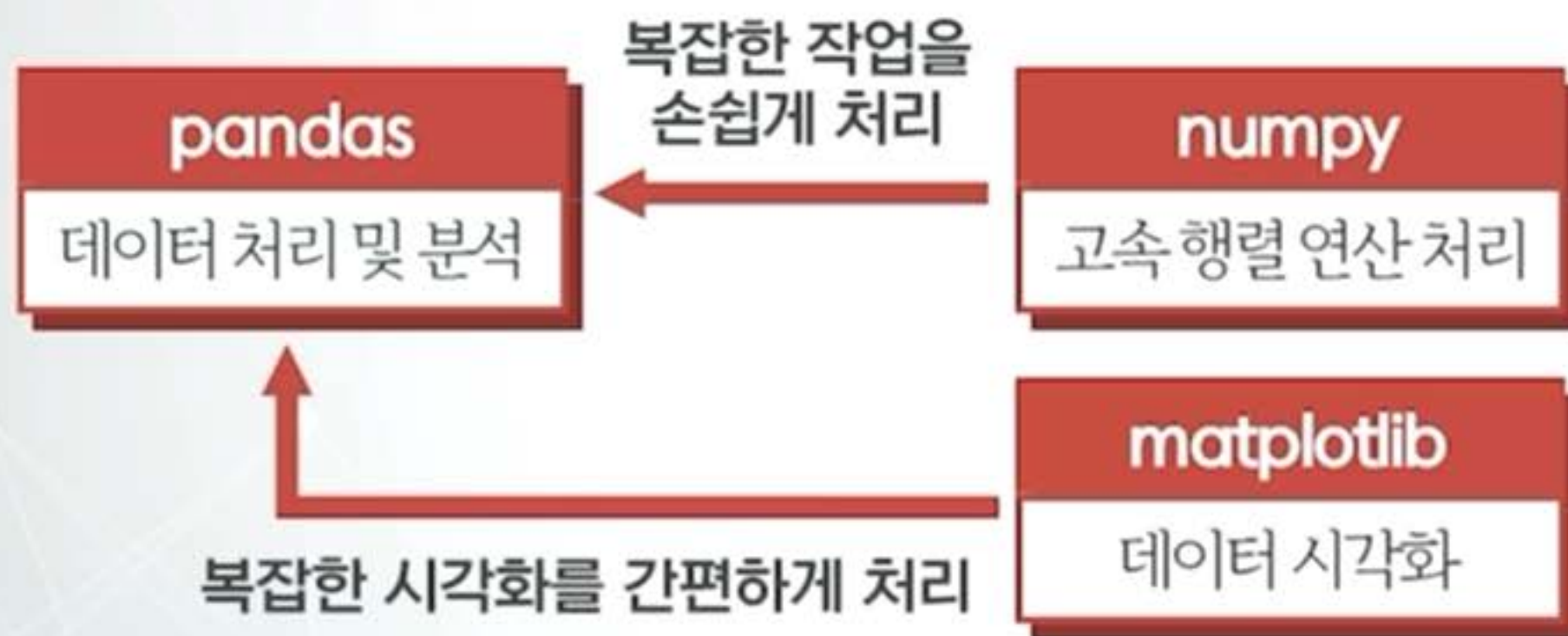
일관성 없는 인터페이스로 사용하기 어려움





Goal

## ➤ Pandas를 통한 데이터 시각화





Goal

## 데이터 시각화



### 학습개요

- 1/ Pandas의 plot( ) 함수
- 2/ Plot( ) 함수로 데이터 시각화 실습



Goal

## 데이터 시각화



Series와 DataFrame 객체는  
plot() 함수를 제공



plot() 함수는 matplotlib를 기반으로  
구현되어 있음



02

# plot( ) 함수로 데이터 시각화 실습





Goal

matplotlib

pandas

plot(x)

bar(x, height)

hist(x,bins)

plot( )

```
graph LR; pandas[plot( )] --> matplotlib_plot[plot(x)]; pandas --> matplotlib_bar[bar(x, height)]; pandas --> matplotlib_hist[hist(x,bins)];
```



Goal

pandas

matplotlib

plot(kind = 'line')

plot(x)

plot(kind = 'bar')

bar(x, height)

plot(kind = 'hist')

hist(x,bins)

## 주의하기

- ✓ Pandas.plot( ) 함수는 통일되고 간결한 인터페이스로 간단히 사용할 수 있지만, 정교한 작업은 불가능
- ✓ 정교한 작업이 필요하다면 matplotlib 사용을 권장





Pandas의 plot() 함수

## Plot함수 인자값

DataFrame.plot()	
X	x축으로 사용될 컬럼을 지정, 기본값 = row index
Y	y축의 값으로 사용할 컬럼을 지정, 기본값 = 모든 수치형 컬럼
kind	사용할 차트의 형태, 기본값 = 'line'
subplots	각각의 컬럼을 별도의 axes로 그릴지 여부, 기본값 = False
figsize	figure의 크기, 튜플(가로(width), 세로(length))의 형태로 지정 예) (3, 5) → 가로 3, 세로 5 크기의 figure를 생성하여 axes 그림
title	axes의 제목, subplots = True라면, 리스트의 형태로 지정
style	컬럼별 라인 스타일 지정
xticks, yticks	x축과 y축의 tick 각각 지정
xlim, ylim	x축과 y축의 범위를 각각 지정
xlabel, ylabel	x축과 y축 이름 지정 (버전 1.10부터 사용 가능)



# 1 Pandas의 plot() 함수

Kind		Style								
		색깔		마커		선 스타일		기타 스타일		
line	line plot (default)	blue	b	.	point marker	-	실선	color	c	선 색깔
		green	g	.	pixel marker	--	대시선	linewidth	lw	선 굵기
bar	vertical bar plot	red	r	o	circle marker	-.	점선	linestyle	ls	선 스타일
		cyan	c	v	triangle_down	:	대시-점선	marker		마커 종류
bath	horizontal bar plot	magenta	m	^	triangle_up			markersize	ms	마커 크기
		yellow	y	<	triangle_left			marker edge color	mec	마커 선 색깔
hist	histogram	black	k	>	triangle_right			marker edge width	mew	마커 선 굵기
box	boxplot	white	w	1	tri_down			marker face color	mfc	마커 내부 색깔
kde	Kernel Density Estimation plot			2	tri_up					
				3	tri_left					
				4	tri_right					
pie	pie plot			S	square marker					
scatter	scatter plot			P	pentagon marker					
				*	star marker					
				h	hexagon1 marker					
				H	hexagon2 marker					
				+	plus marker					
				x	x marker					
				D	diamond marker					
				d	thin_diamond					



## 1 Pandas의 plot() 함수

# Plot함수 인자값

DataFrame.plot()	
X	x축으로 사용될 컬럼을 지정, 기본값 = row index
Y	y축의 값으로 사용할 컬럼을 지정, 기본값 = 모든 수치형 컬럼
kind	사용할 차트의 형태, 기본값 = 'line'
subplots	각각의 컬럼을 별도의 axes로 그릴지 여부, 기본값 = False
figsize	figure의 크기, 튜플(가로(width), 세로(length))의 형태로 지정
title	axes의 제목, subplots = True라면, 형태로 지정
style	컬럼별 라인 스타일 지정
xticks, yticks	x축과 y축의 tick 각각 지정
xlim, ylim	x축과 y축의 범위를 각각 지정
xlabel, ylabel	x축과 y축 이름 지정 (버전 1.10부터 사용 가능)

**Tip**

✓ 그 외 인자값은 공식홈페이지를  
통해 확인 가능



01

# Pandas의 plot( ) 함수



# Pandas의 plot 함수로 데이터 시각화

## ◎ 기본 그리기



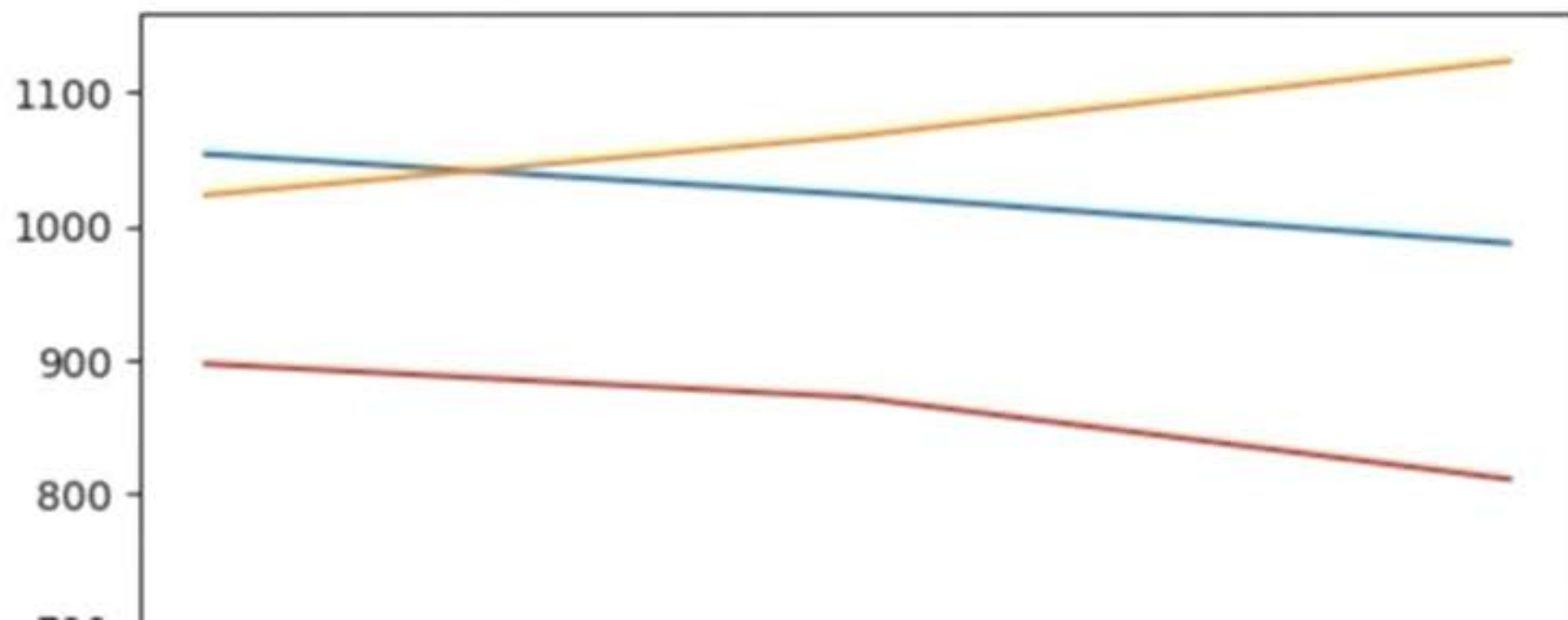
기본 그리기 - [line\\_graph](#)

```
In [79]: plt.plot(df)
```

```
Out[79]: [<matplotlib.lines.Line2D at 0x2015d4a9280>,  
<matplotlib.lines.Line2D at 0x2015d4a9340>,  
<matplotlib.lines.Line2D at 0x2015d4a9400>,  
<matplotlib.lines.Line2D at 0x2015d4a94c0>,  
<matplotlib.lines.Line2D at 0x2015d4a9580>]
```

**plt.plot(df)**

Matplotlib.plot 함수로 라인 그래프 생성



# Pandas의 plot 함수로 데이터 시각화

## ◎ 기본 그리기



서울 경기 충청 경상 전라

2018	1053.5	1023	512
2019	1023.0	1067	489
2020	987.0	1123	487

**df.plot()**

DataFrame에서 제공하는 plot 함수

기본 그리기 - [line\\_graph](#)

```
In [80]: #plt.plot(df)
df.plot()
```

```
Out[80]: <AxesSubplot:>
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning: Glyph 49436 missing from current font.
```

```
font.set_text(s, 0.0, flags=flags)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning: Glyph 50872 missing from current font.
```

```
font.set_text(s, 0.0, flags=flags)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning: Glyph 44221 missing from current font.
```

```
font.set_text(s, 0.0, flags=flags)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: Ru
```



# Pandas의 plot 함수로 데이터 시각화

## ◎ 기본 그리기

✓ 폰트가 깨진 경우

	서울	경기	충청	경상	전라
2018	1053.5	1023	512	897	451
2019	1023.0	1067	489	872	421
2020	987.0	1123	487	811	399

맑은 고딕으로 변경

기본 그리기 - [line graph](#)

```
In [81]: matplotlib.rcParams['font.family'] = 'Malgun Gothic'
```

```
Out[81]: ['sans-serif']
```

```
In [80]: #plt.plot(df)
df.plot()
```

```
Out[80]: <AxesSubplot:>
```

```
C:\ProgramData\Anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning: Glyph 49436 missing from current font.
font.set_text(s, 0.0, flags=flags)
```

# Pandas의 plot 함수로 데이터 시각화

## ◎ 기본 그리기

✓ 폰트가 깨진 경우

	서울	경기	충청	경상	전라
	1053.5	1023	512	897	451
2019	1023.0	1067	489	872	421
2020	987.0	1123	487	811	399

ios 사용자

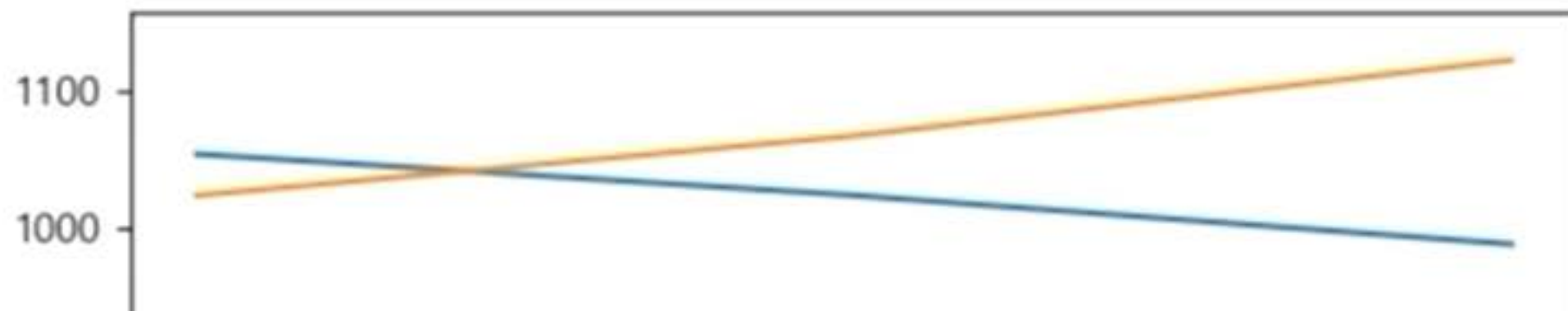
AppleGothic으로 변경

기본 그리기 - [line\\_graph](#)

```
In [82]: matplotlib.rcParams['font.family'] = 'Malgun Gothic'  
# ios 사용자는 AppleGothic으로 변경
```

```
In [83]: #plt.plot(df)  
df.plot()
```

Out[83]: <AxesSubplot:>



# Pandas의 plot 함수로 데이터 시각화

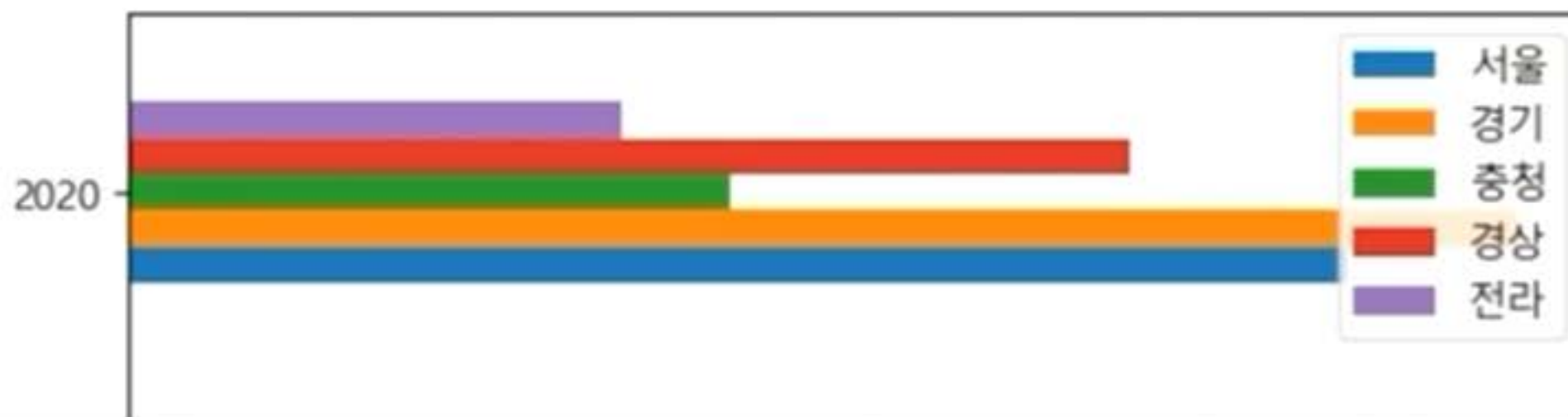
## ◎ 인자값 테스트

✓ kind : line, bar, barh

```
In [82]: matplotlib.rcParams['font.family'] = 'Malgun Gothic'  
# ios 사용자는 AppleGothic으로 변경
```

```
In [84]: #plt.plot(df)  
df.plot(kind = 'line')
```

Out[84]: <AxesSubplot:>



kind로 차트 종류를 지정할 수 있습니다.



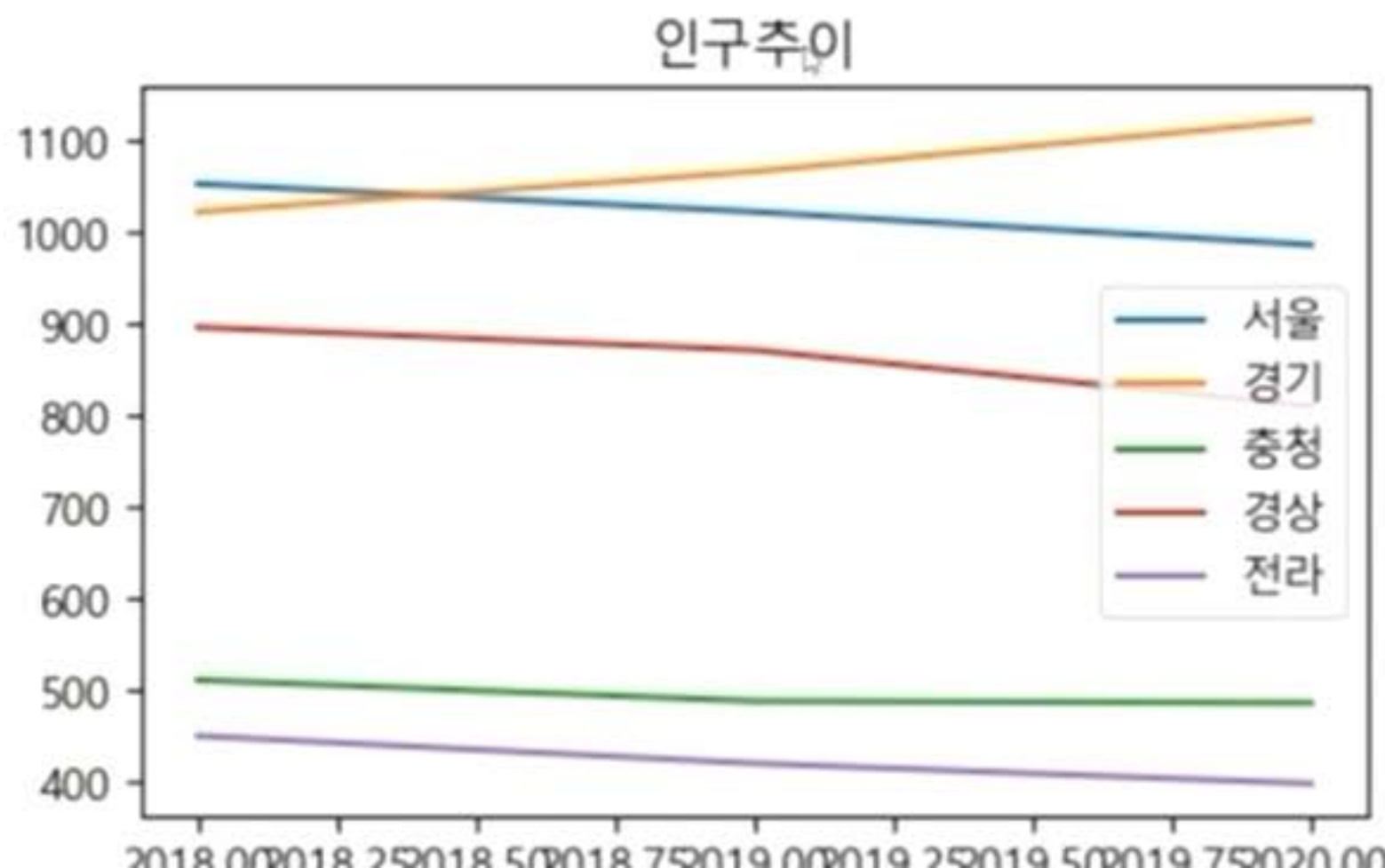
# Pandas의 plot 함수로 데이터 시각화

## ◎ 인자값 테스트

```
In [89]: #plt.plot(df)
df.plot(kind = 'line',
        figsize = (5,3),
        title = '인구추이')
```

✓ figsize, title

<AxesSubplot: title={ center : 인구추이 }>



# Pandas의 plot 함수로 데이터 시각화

## ◎ 인자값 테스트

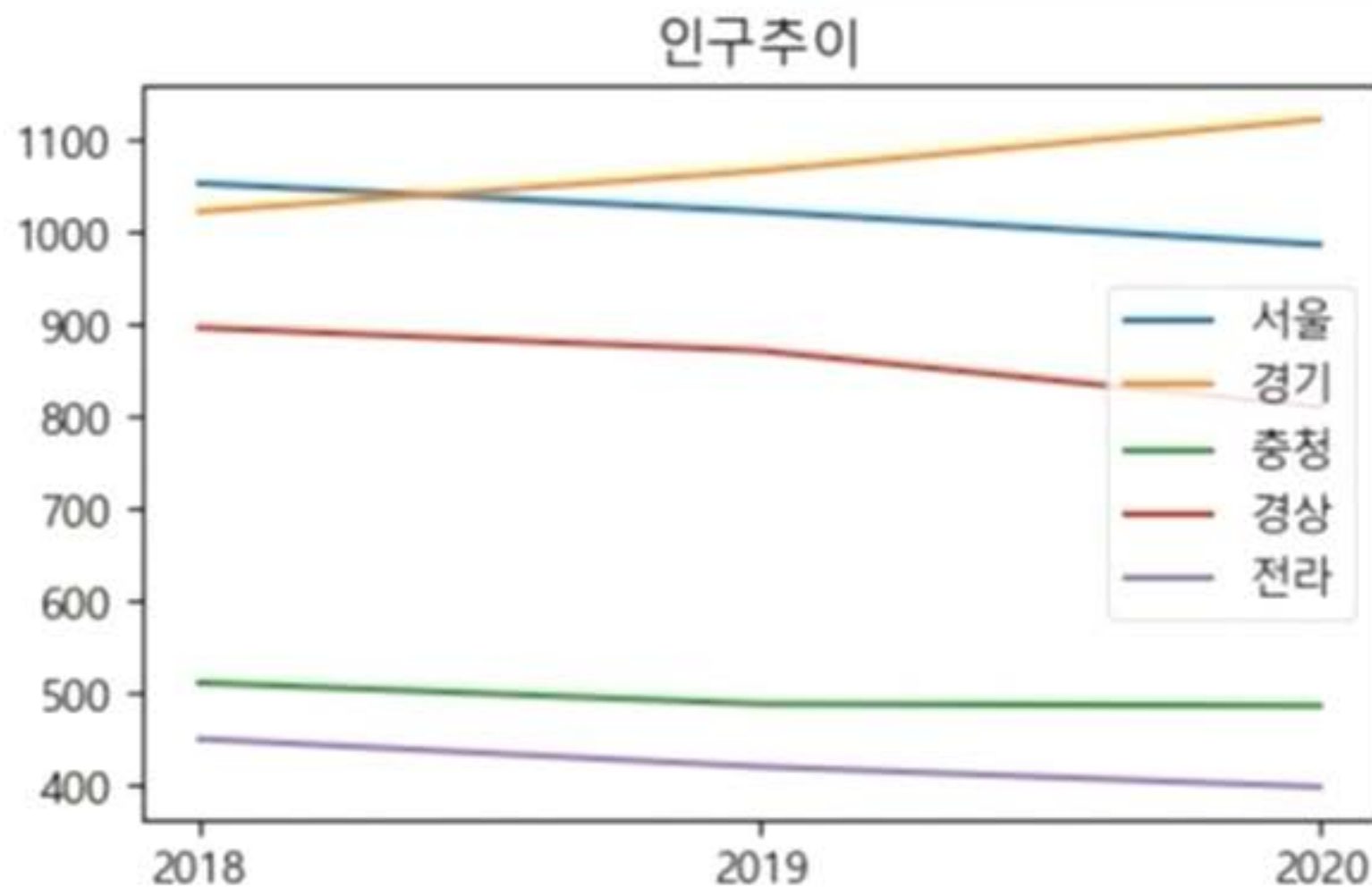
```
df.plot(kind='line',  
        figsize=(5,3),  
        title='인구추이',  
        xticks=df.index  
        )
```

Out[91]: <AxesSubplot:title={'center':'인구추이'}>

**df.index**

df의 row index를 xtick으로 지정

✓ xticks



# Pandas의 plot 함수로 데이터 시각화

## ◎ 인자값 테스트

```
xticks = df.index,  
ylim = [300, 1300]  
)
```

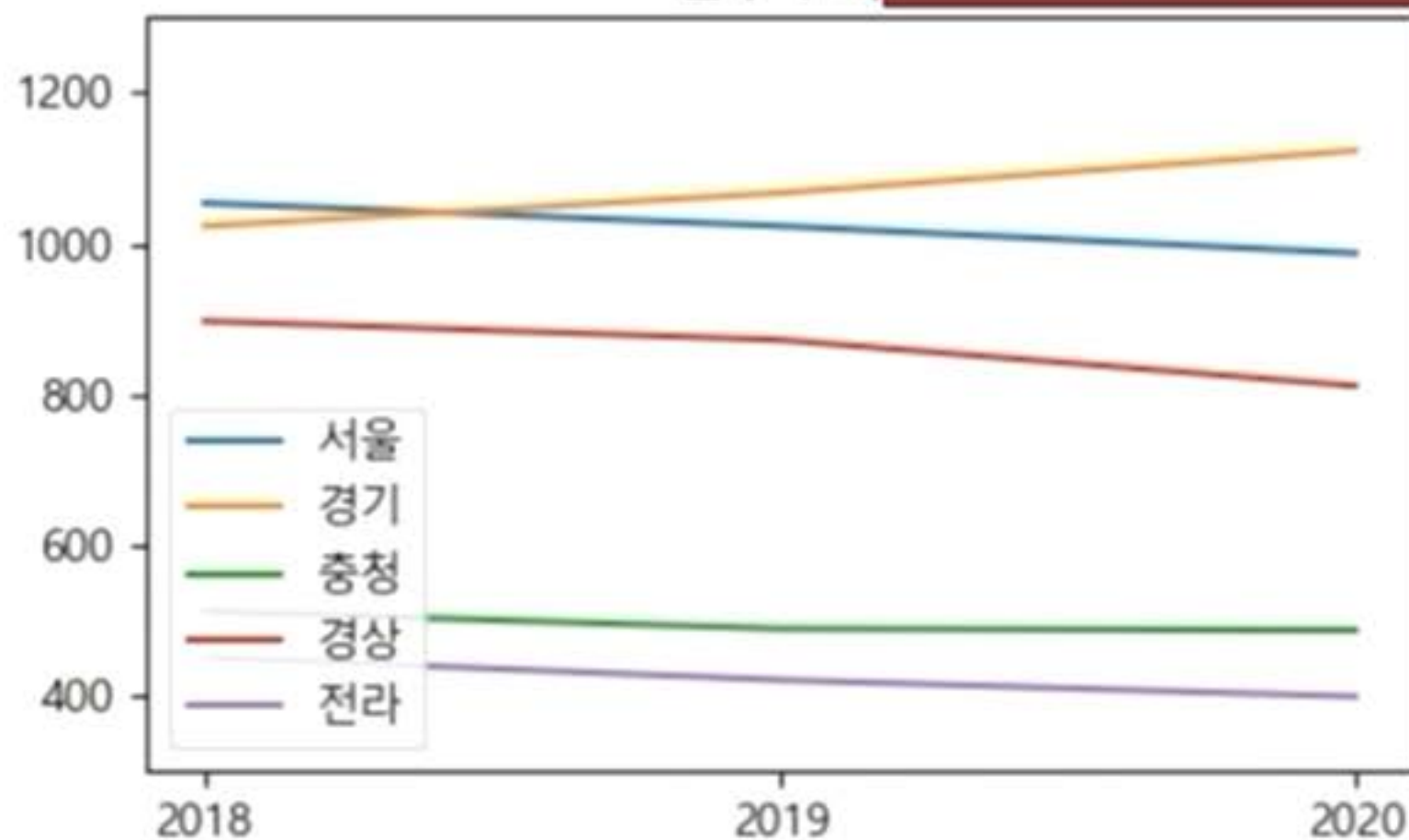
Out[92]: <AxesSubplot:title={'center': '인구추이'}

인구추이

ylim

y축의 limitation(범위값)

✓ ylim





# Pandas의 plot 함수로 데이터 시각화

## ◎ 인자값 테스트

✓ xlabel, ylabel

```
.plot(kind = 'line',  
      figsize = (5,3),  
      title = '인구추이',  
      xticks = df.index,  
      ylim = [300, 1300],  
      xlabel = '년도',  
      ylabel = '인구수' # v 1.1 이상만 가능  
      )  
# 만약 v1.1 이하였다면  
# plt.xlabel('년도')  
# plt.ylabel('인구수')
```

xlabel, ylabel

x축과 y축 제목

Out[94]: <AxesSubplot:title={'center':'인구추이'}, xlabel='년도', ylabel='인구수'>



xlabel, ylabel은 Pandas 버전이 1.1 이상만 가능합니다.

Pandas 버전이 1.1 이하인 경우, plt.xlabel()/plt.ylabel()을 사용하여 제목을 지정합니다.

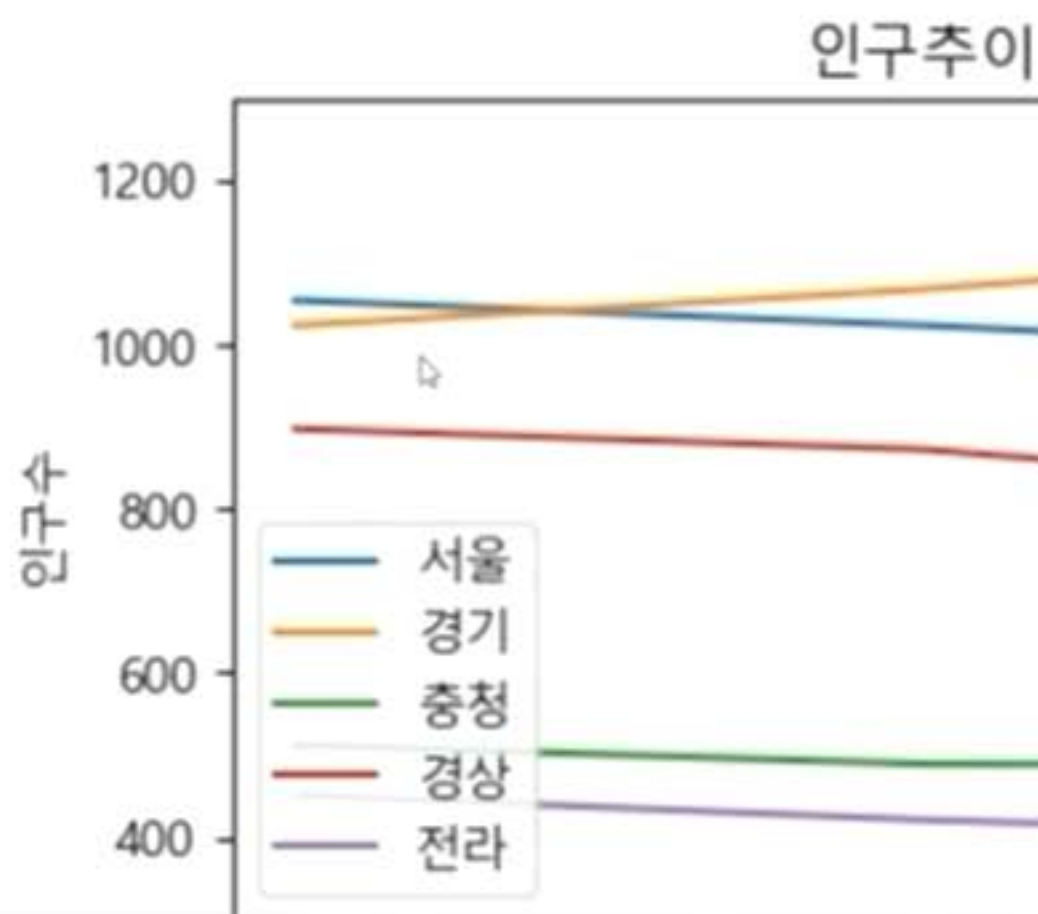
# Pandas의 plot 함수로 데이터 시각화

## ◎ 인자값 테스트

✓ style

```
# 만약 v1.1 이하였다면  
#plt.xlabel('년도')  
#plt.ylabel('인구수')
```

Out[94]: <AxesSubplot:title={'center':'인구추이'}, xlabel='인구추이', ylabel='인구수'>



style

컬럼별 라인 스타일 지정

style은 컬럼의 개수(라벨의 개수)만큼 지정해야 합니다.

# Pandas의 plot 함수로 데이터 시각화

## ◎ 인자값 테스트

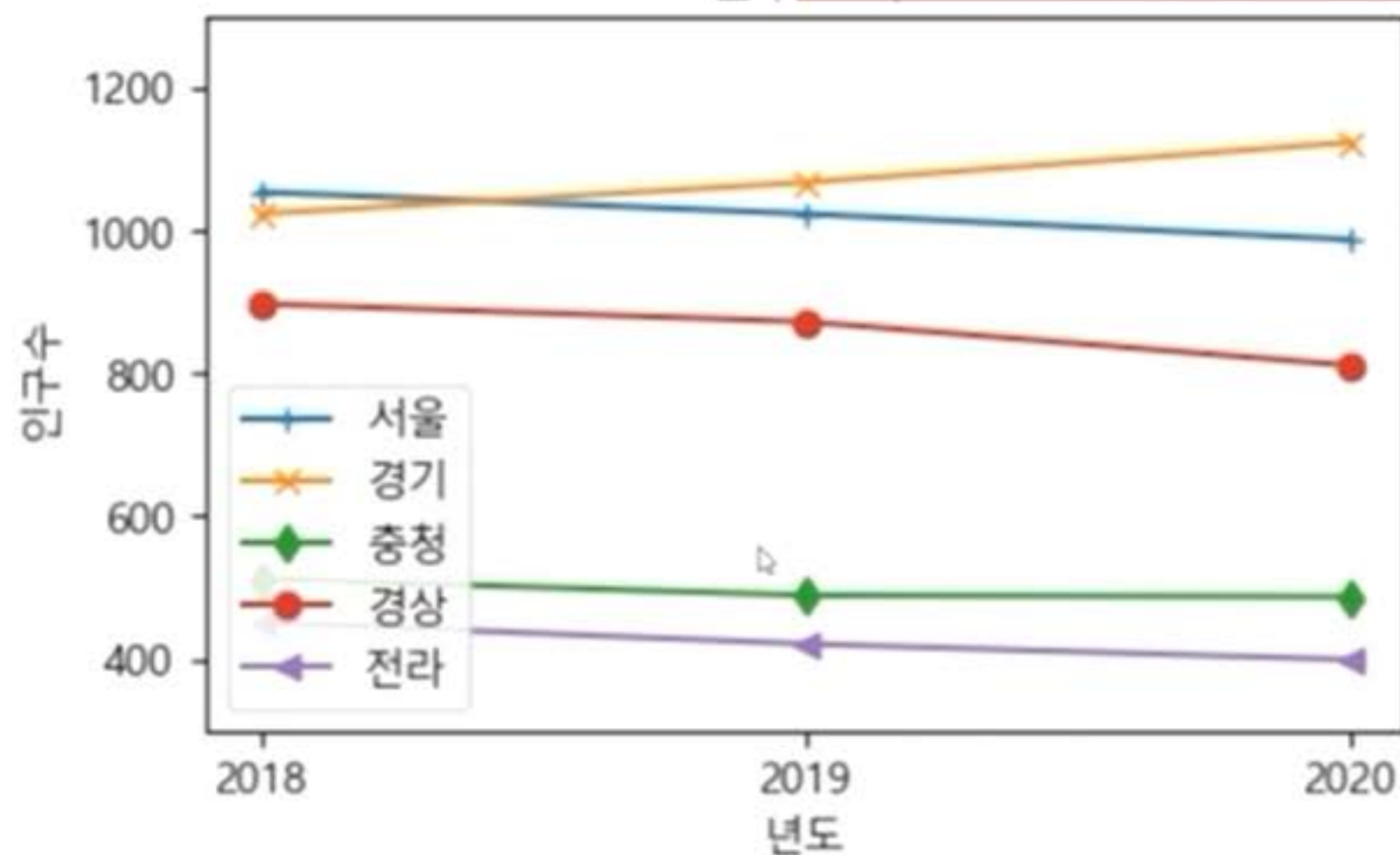
✓ subplots

```
)  
# 만약 v1.1 이하였다면  
#plt.xlabel('년도')  
#plt.ylabel('인구수')
```

Out[97]: <AxesSubplot:title={'center': '인구추'>

subplots

기본값: False - 하나의 Axes 안에 표현





# Pandas의 plot 함수로 데이터 시각화

## ◎ 인자값 테스트

✓ subplots

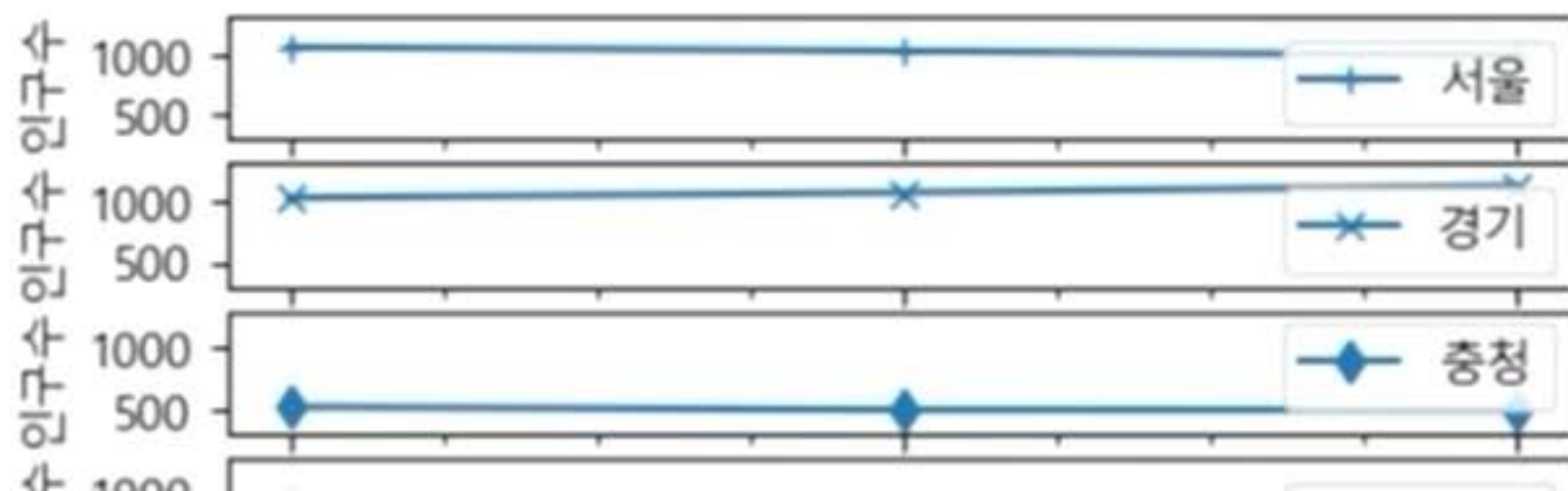
```
xlabel = '년도',  
ylabel = '인구수', # v 1.1 이상만 가능  
style = ['+-','x-','d-','o-']  
subplots = True  
)  
# 만약 v1.1 이하였다면  
# plt.xlabel('년도')  
# plt.ylabel('인구수')
```

subplots

기본값 : True - 컬럼별로 Axes를 생성

```
Out[98]: array([<AxesSubplot: xlabel='년도', ylabel='인구수'>,  
                <AxesSubplot: xlabel='년도', ylabel='인구수'>,  
                <AxesSubplot: xlabel='년도', ylabel='인구수'>,  
                <AxesSubplot: xlabel='년도', ylabel='인구수'>,  
                <AxesSubplot: xlabel='년도', ylabel='인구수'>], dtype=object)
```

인구추이



# Pandas의 plot 함수로 데이터 시각화

## © Plot 함수로 데이터 시각화 실습하기



# Pandas의 plot 함수로 데이터 시각화

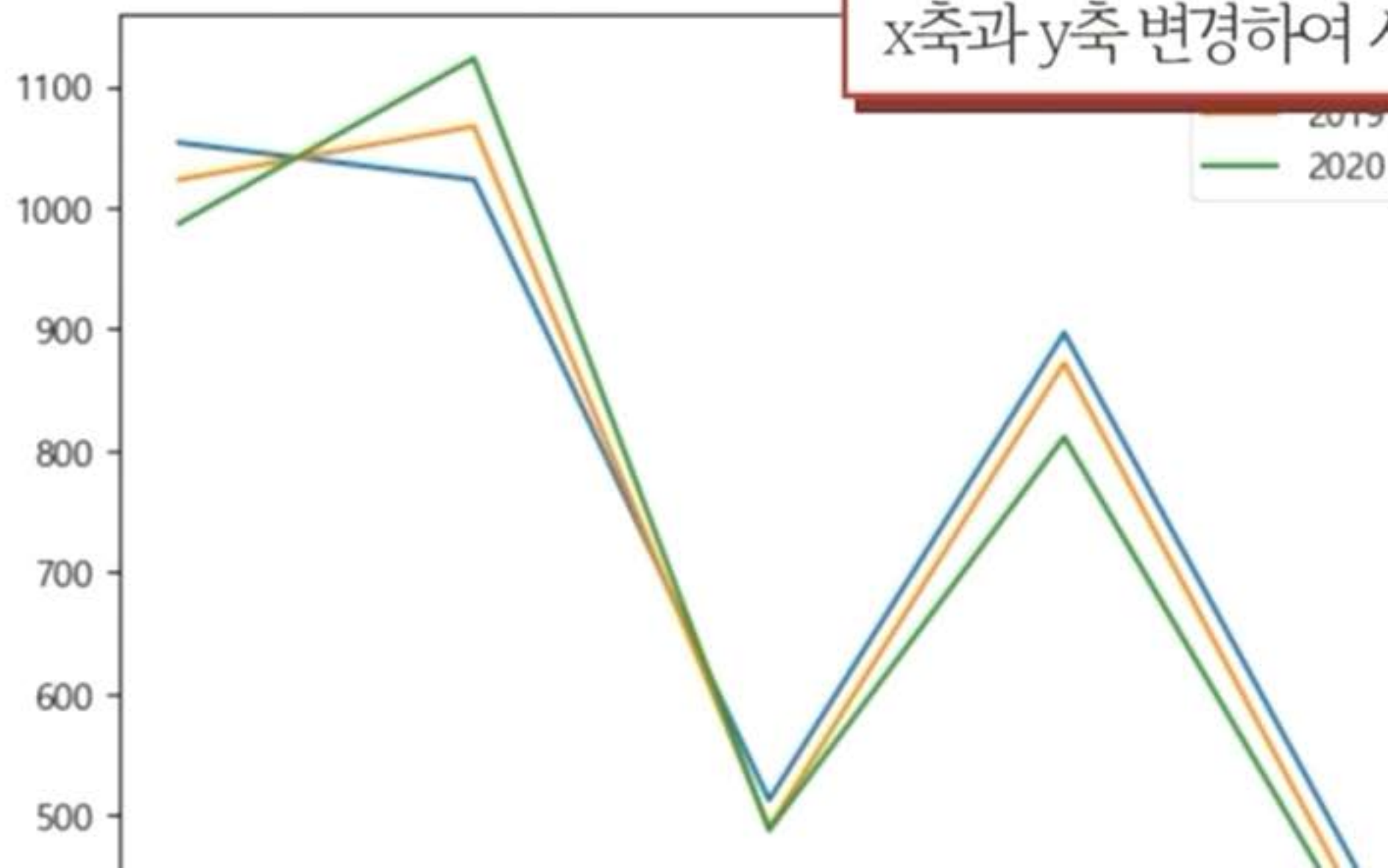
## ◎ Plot 함수로 데이터 시각화 실습하기

```
In [99]: # 실습. 연도별 지역별 인구수 그래프 그리기  
df.T.plot()
```

```
Out [99]: <AxesSubplot:>
```

**df.T.plot()**

x축과 y축 변경하여 시각화





## 데이터 시각화



### 학습완료

- 1/ Pandas의 plot( ) 함수
- 2/ Plot( ) 함수로 데이터 시각화 실습



수고하셨습니다!