

시와 데이터 기초

데이터 시각화

오늘 수업은

- ❖ 데이터 시각화의 정의와 필요성
- ❖ 상황에 맞는 차트 선택 방법
- ❖ 데이터 시각화를 위한 대표 라이브러리
- ❖ [실습 내용]
 - 혼인건수와 출생아수의 변화 살펴보기
 - 데이터 시각화
 - plot차트 중심
- ❖ bar 차트 그리기
- ❖ bar 차트의 속성 변경하기
- ❖ pie 차트 그리기
- ❖ scatter 차트 그리기
- ❖ [실습 내용]
 - 시도별 교통사고 현황 알아보기
 - bar 차트
 - pie 차트
 - scatter 차트

데이터 시각화의 정의

❖ 사람의 시각 및 이해도

- 사람은 긴 글보다 요약된 내용을 선호
- 글자보다 임팩트 있는 시각 요소에 집중
- 데이터 원자료나 통계표는 수많은 숫자와 문자로 구성됨으로 내용 파악이 어려움

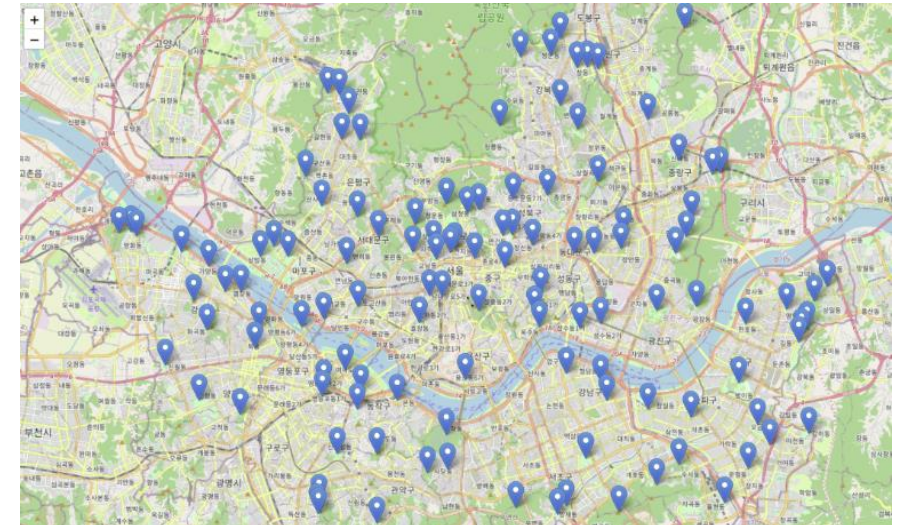
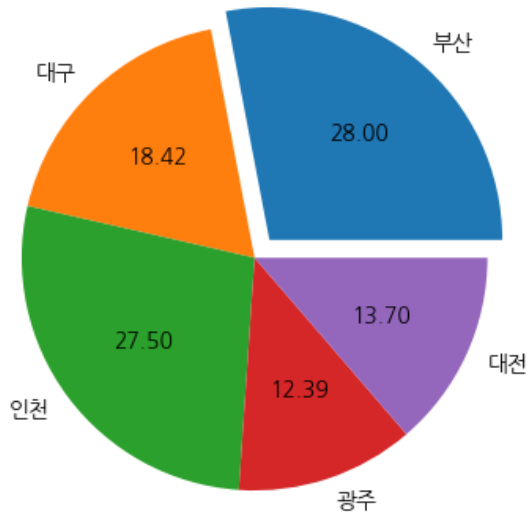
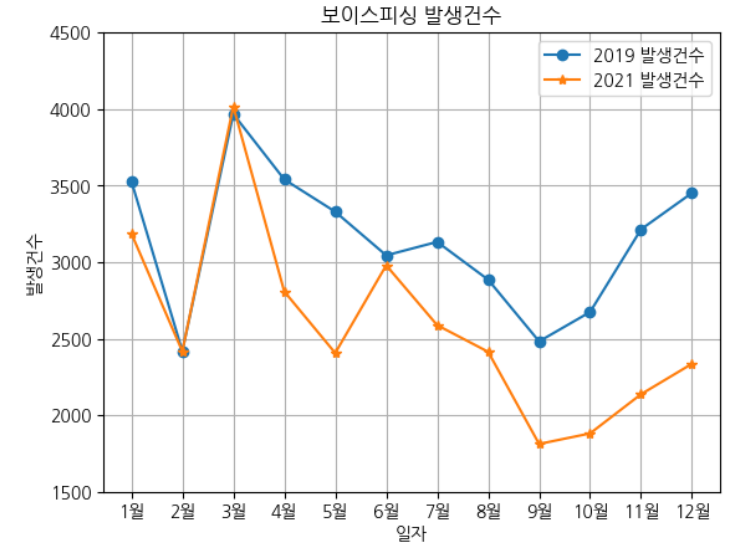
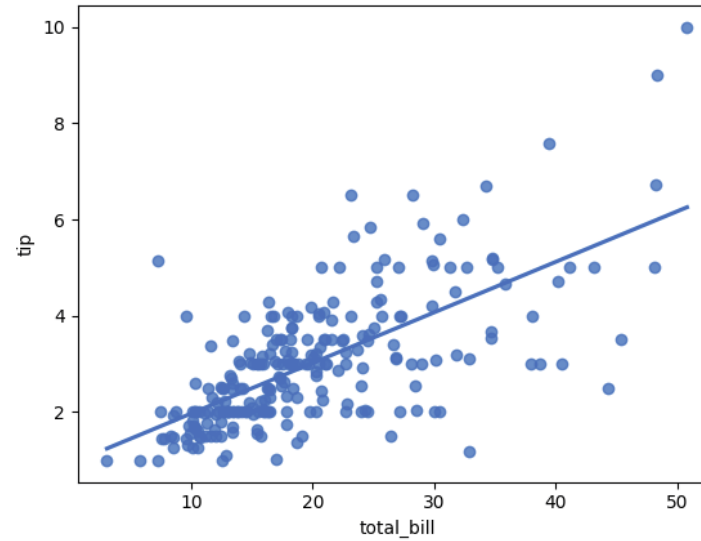
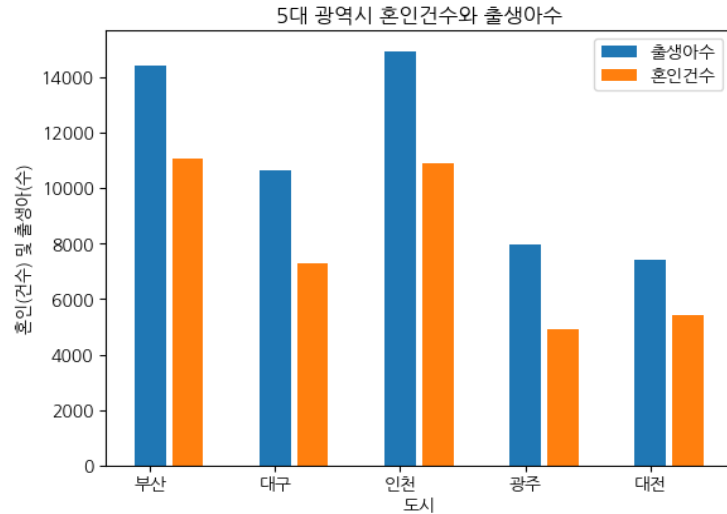
❖ 데이터 시각화

- 일반적인 데이터를 차트, 그래프, 지도와 같이 이해하기 쉬운 시각적 요소로 표현하는 것
- 많은 양의 데이터를 빠르게 분석

데이터를 시각화 하는 이유

- ❖ 많은 양의 데이터를 빠르게 분석할 수 있다.
- ❖ 시각 요소를 통해 데이터를 명확히 전달할 수 있다.
- ❖ 추세와 경향성이 쉽게 드러나 이해가 쉽다.
- ❖ 새로운 패턴을 발견하기도 한다.
- ❖ 항목간의 관계를 발견할 수 있다.
- ❖ 데이터에 숨어 있는 트렌드를 찾아낼 수 있다.

대표적인 차트 종류



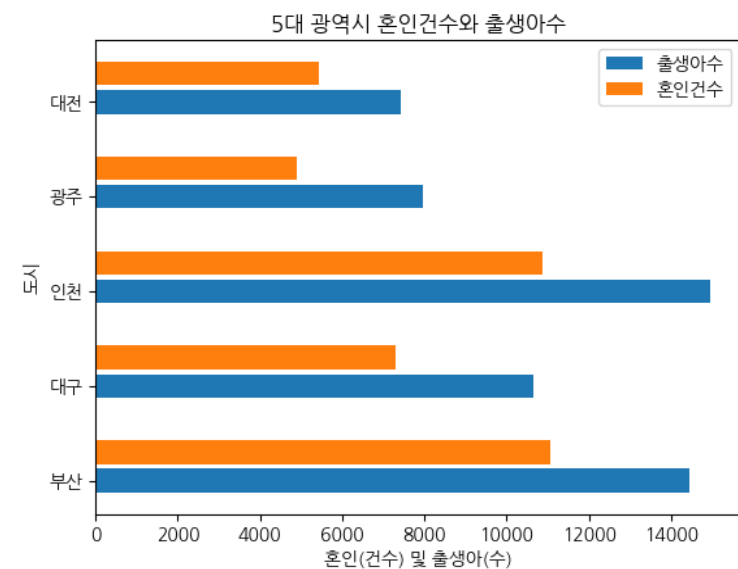
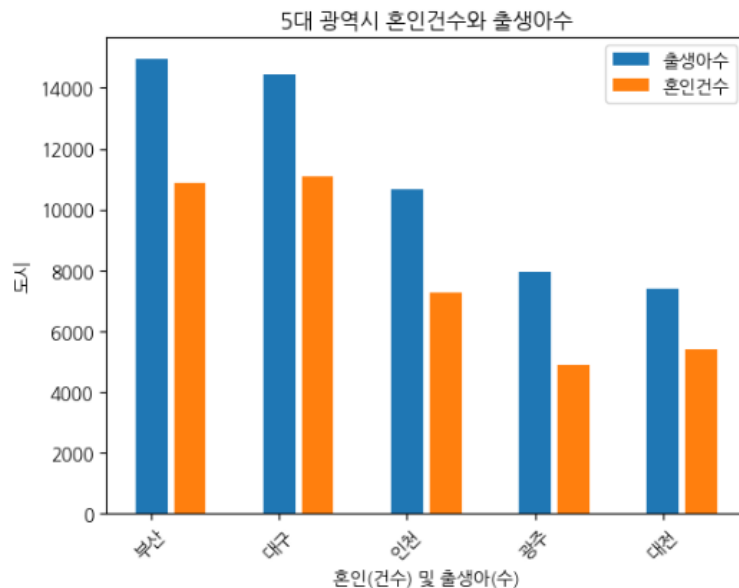
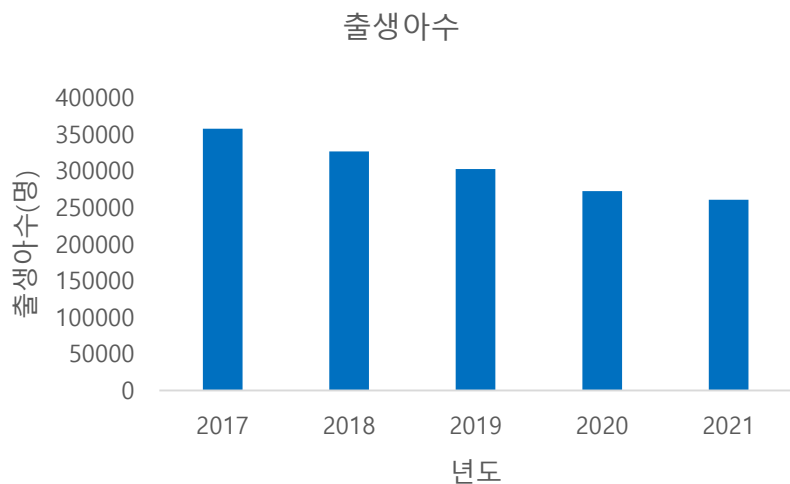
상황에 맞는 차트 선택 방법

❖ 막대차트(Bar Chart)

- 데이터의 트렌드를 파악할 때 유용
- 데이터의 순위를 비교할 때 유용

❖ 참고사항

- 비교 데이터가 많다면 수평 막대가 유용
- 순위를 강조하고 싶다면 정렬(오름차순/내림차순)하여 표현
- 데이터를 구분하고자 한다면 데이터별로 다른 색상을 지정



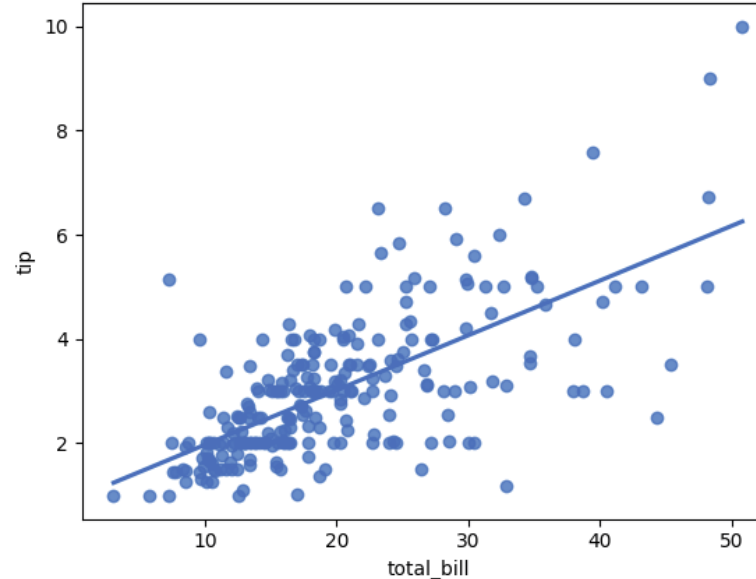
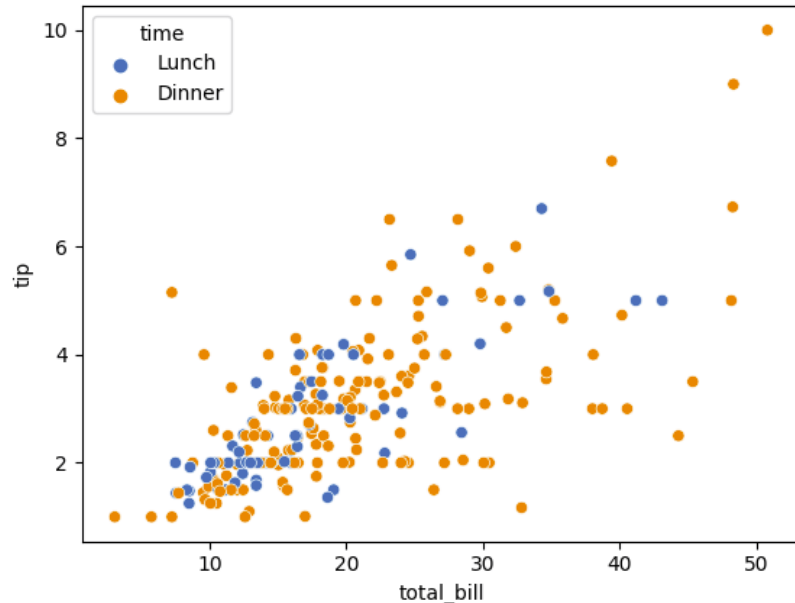
상황에 맞는 차트 선택 방법

❖ 분산형 차트(Scatter Chart)

- 직교 좌표계를 이용하여 좌표상의 점들을 표시
- 두 개의 데이터(변수)간의 관계를 나타낼 때 유용
- 분포 양상을 비교할 때 유용

❖ 참고사항

- 데이터(변수) 간의 관계를 명확하게 표현하고 싶다면 추세선을 사용
- 변수의 관계가 2개 이상일때는 색상으로 표현



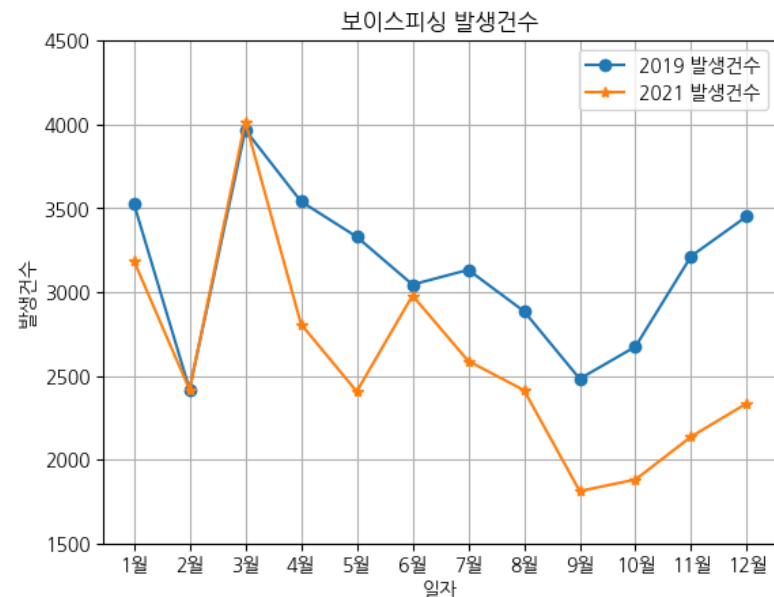
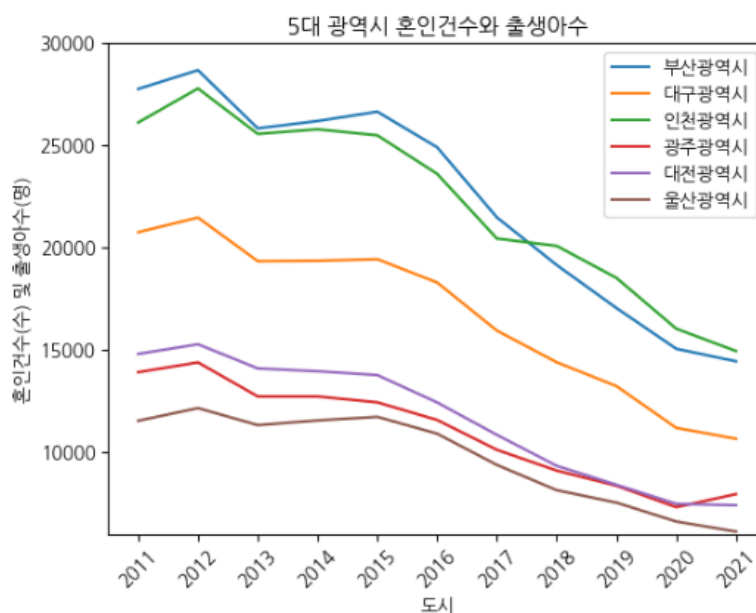
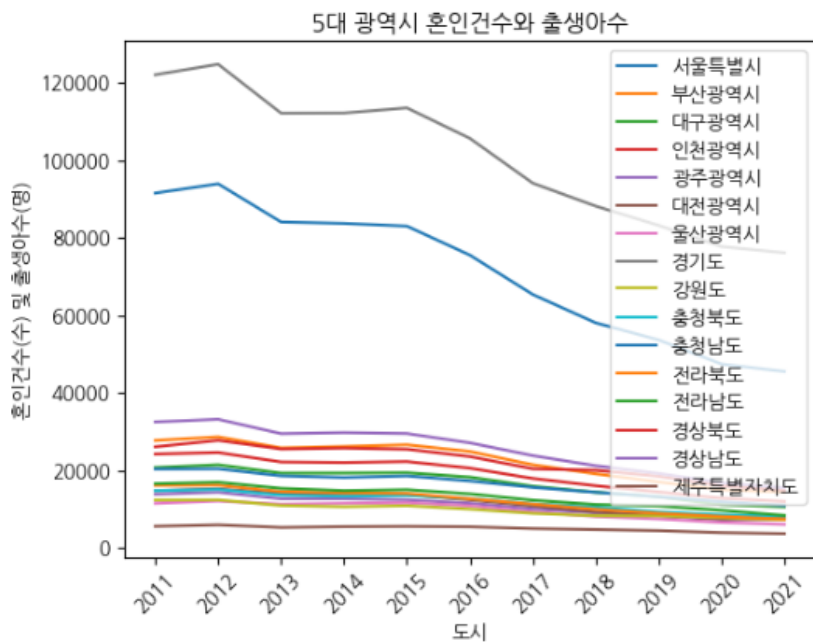
상황에 맞는 차트 선택 방법

❖ 선 차트(Plot Chart)

- 데이터 트렌드의 변화를 비교할 때 유용
- 시간의 흐름에 따른 데이터 변화를 확인하고자 할때 유용

❖ 참고사항

- 선의 개수는 한눈에 파악할 수 있을 정도로만
- 데이터 구분을 위해 색상 활용
- 데이터값이 과장되지 않도록 세로축의 눈금 및 범위는 적절히



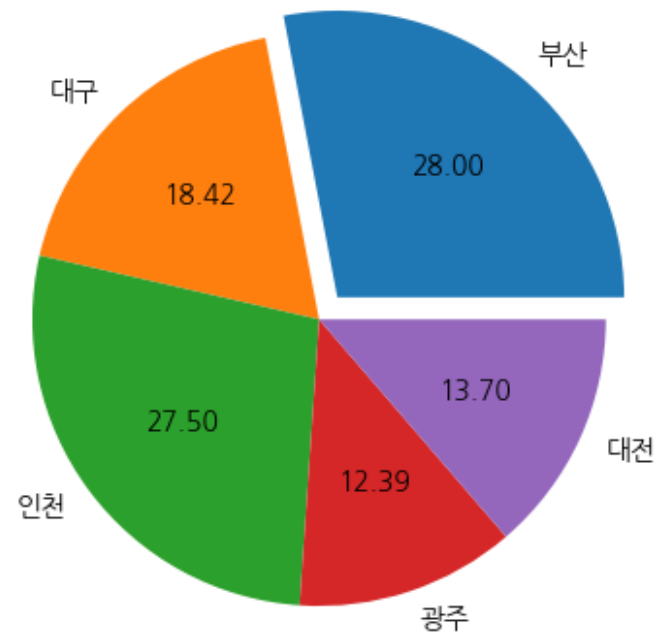
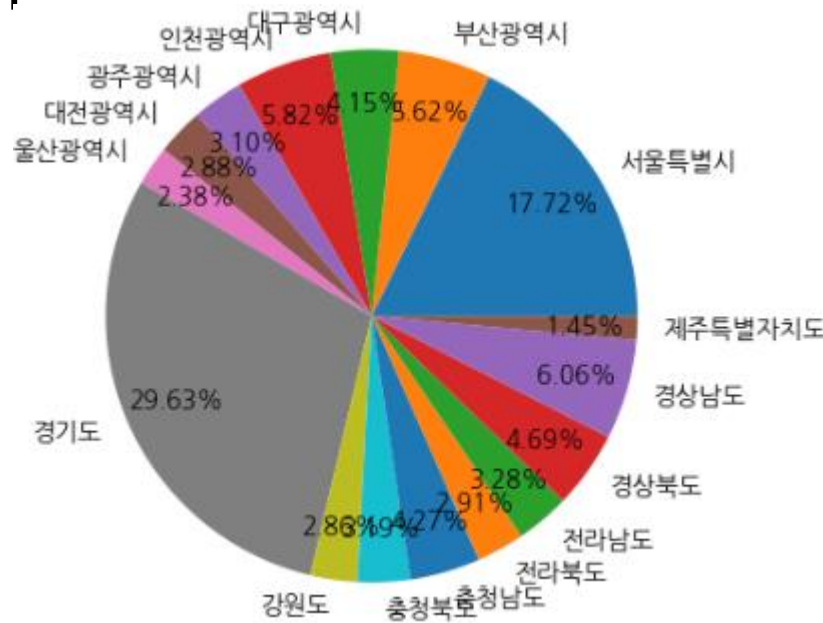
상황에 맞는 차트 선택 방법

❖파이 차트(Pie Chart)

- 데이터간의 비율을 파악할 때 활용
- 데이터간의 상대적 크기를 비교할 때 활용

❖참고사항

- 핵심 전달을 위해 조각 개수는 한눈에 보일 정도로만
- 명확한 크기를 위해 조각을 큰 순서대로 배열하는 것이 효율적임
- 강조하고자 하는 핵심 정보만 표시



상황에 맞는 차트 선택 방법

워드클라우드(WordCloud)

- 서술형 데이터를 분석할 때 활용
- 빈도, 중요도가 높은 텍스트를 강조할 때 유용

참고사항

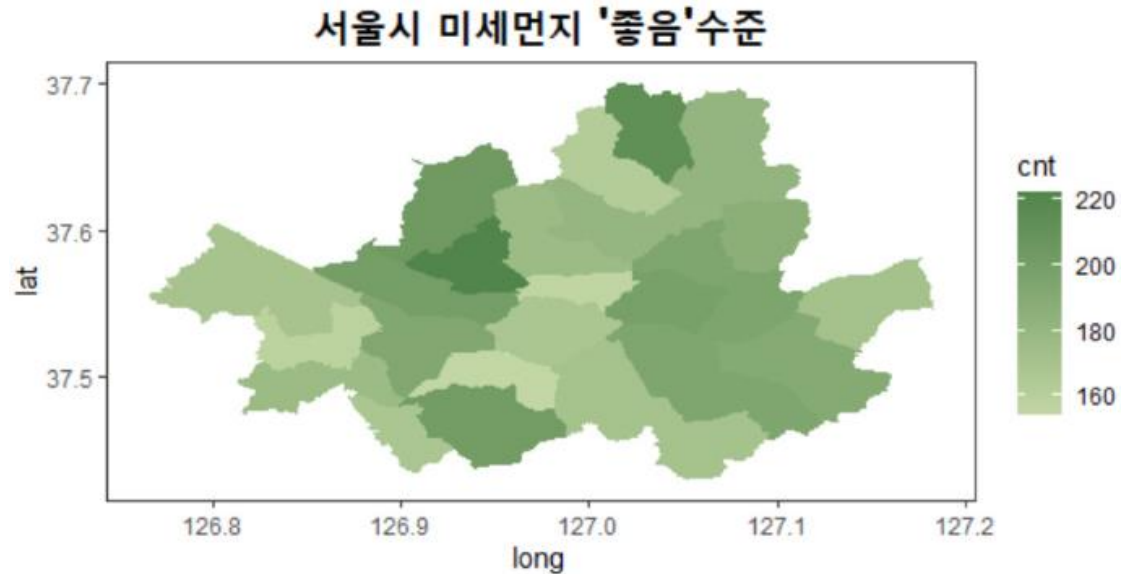
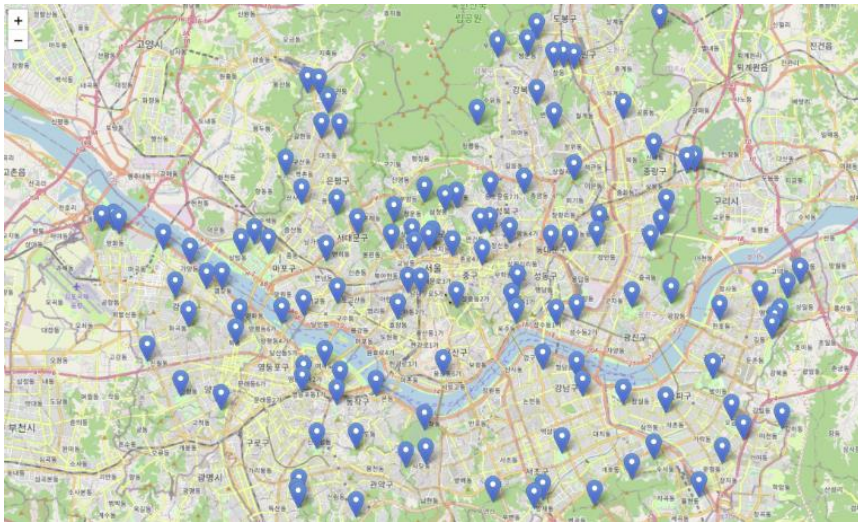
- 빈도가 낮은 단어들은 제외하여 표현
- 의미없는 단어는 제거하여 표현



상황에 맞는 차트 선택 방법

❖ 지도 차트

- 데이터를 지리적으로 비교할 때
- 시간별 지리적 데이터 변화를 확인할 때
- 데이터에 맞는 지도 배경 사용



이미지 출처: <https://brunch.co.kr/@sjh001/45>

상황에 맞는 차트 선택 방법 정리

차트 종류		상황	참고사항
막대	수직	<ul style="list-style-type: none"> 트렌드를 파악할 때 순위를 비교할 때 	<ul style="list-style-type: none"> 비교 데이터가 많다면 수평 막대가 유용 순위를 강조하고 싶다면 정렬 데이터를 구분하고자 한다면 색상 활용
	수평	<ul style="list-style-type: none"> 달성도를 확인할 때 	
선		<ul style="list-style-type: none"> 트렌드의 변화를 비교할 때 시간의 흐름에 따른 데이터 변화를 확인하고자 할 때 	<ul style="list-style-type: none"> 선의 개수는 한눈에 파악할 수 있을 정도로만 데이터 구분을 위해 색상 활용 데이터값이 과장되지 않도록 세로축 범위는 적절히
파이		<ul style="list-style-type: none"> 비율을 파악할 때 데이터간의 상대적 크기를 비교할 때 	<ul style="list-style-type: none"> 핵심 전달을 위해 조각 개수는 적당히 명확한 크기를 위해 조각을 큰 순서대로 배열 강조하고자 하는 핵심 정보만 표시
워드 클라우드		<ul style="list-style-type: none"> 서술형 데이터를 분석할 때 빈도, 중요도가 높은 텍스트를 강조할 때 	<ul style="list-style-type: none"> 빈도가 낮은 단어들은 제외 의미없는 단어는 제거
지도		<ul style="list-style-type: none"> 데이터를 지리적으로 비교할 때 시간별 지리적 데이터 변화를 확인할 때 	<ul style="list-style-type: none"> 데이터에 맞는 지도 배경 사용

데이터 시각화를 위한 대표 라이브러리(matplotlib)

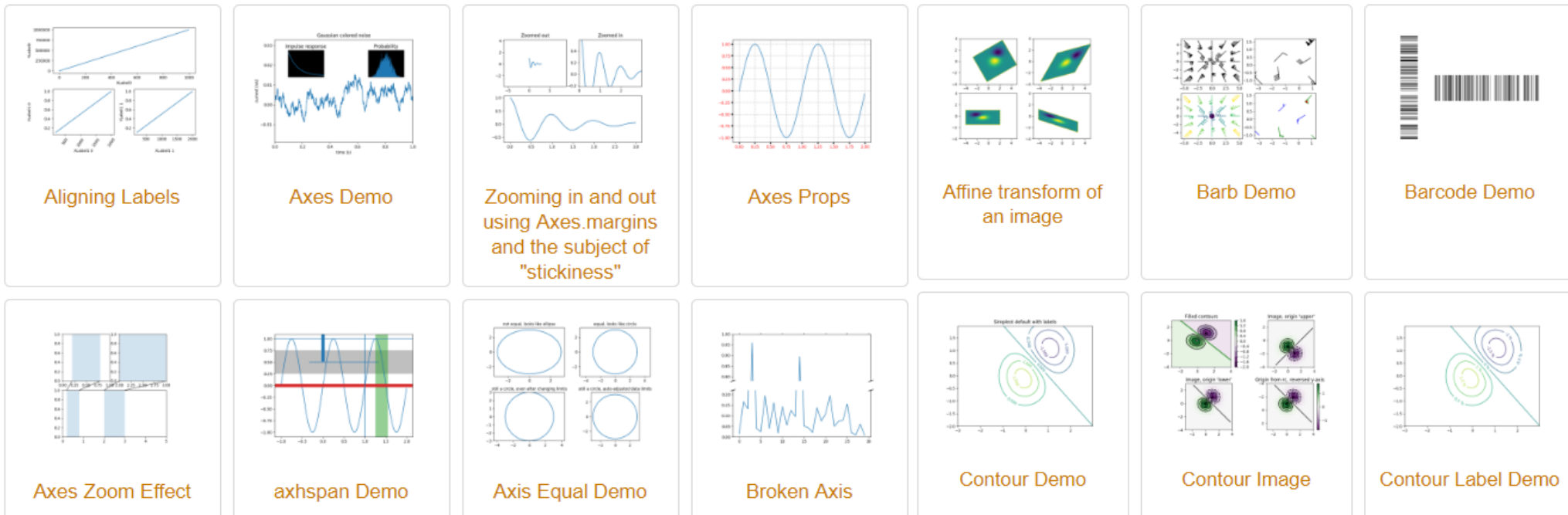
❖ matplotlib(mat + plot + lib)

- 데이터를 다양한 형태의 차트로 그려주는 데이터 시각화 패키지
- 설치 : `pip install matplotlib`

<https://matplotlib.org/>



❖ 라이브러리 선언 `import matplotlib.pyplot as plt`

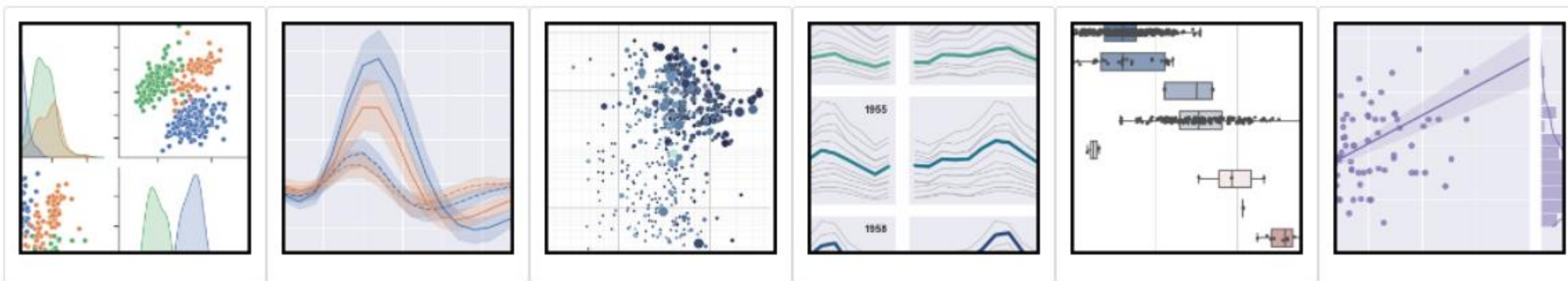


데이터 시각화를 위한 대표 라이브러리(seaborn)

<https://seaborn.pydata.org/>

❖ seaborn 라이브러리

- matplotlib 위에서 동작되는 시각화 라이브러리
- 22종의 데이터 셋 제공
- matplotlib에 비해 손쉽게 그래프를 그리고 스타일을 설정할 수 있음



❖ seaborn 라이브러리 선언

```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
```

데이터 시각화 (plot 차트 중심)

[실습내용]

1. 차트 구성
2. plot 차트 생성
3. plot 차트 속성 설정

matplotlib 라이브러리

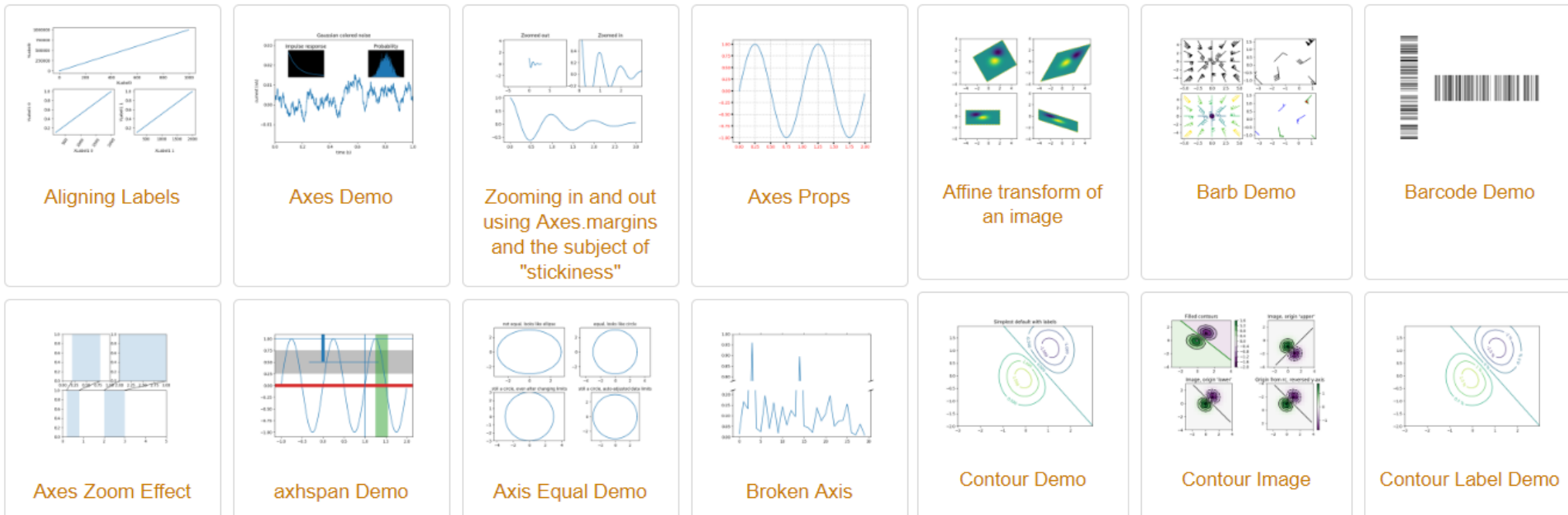
❖ matplotlib(mat + plot + lib)

- 데이터를 다양한 형태의 차트로 그려주는 데이터 시각화 패키지
- 설치 : `pip install matplotlib`

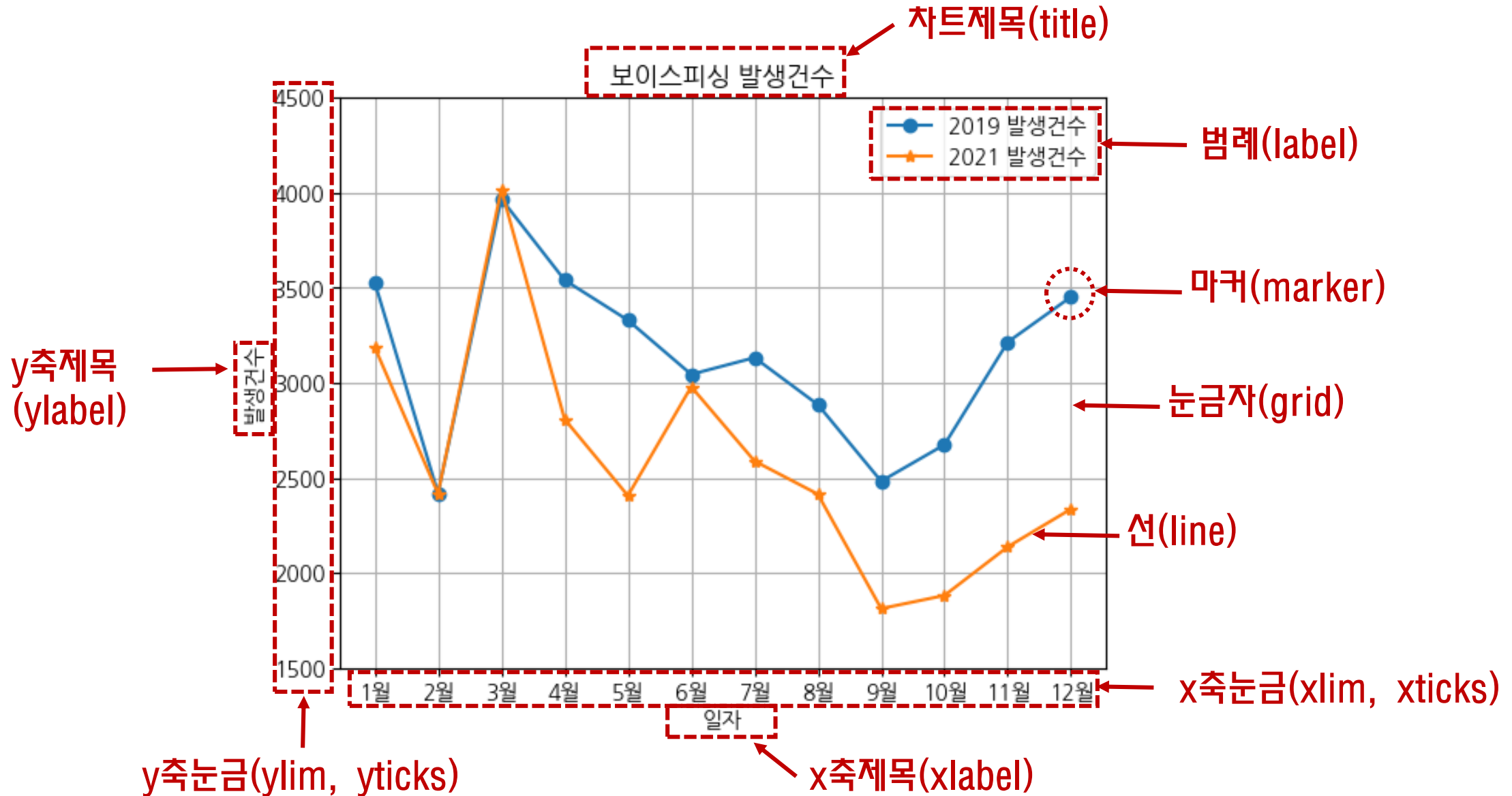
<https://matplotlib.org/>



❖ 라이브러리 선언 `import matplotlib.pyplot as plt`



차트의 구성



질문하기

❖ 차트를 생성하여 다음의 질문에 답해보자.

❖ 시간이 지날수록 출생아수의 변화가 있는가?

1. 혼인건수가 가장 적은 년도와 가장 많은 년도는 언제인가?
2. 출생아수가 가장 적은 년도와 가장 많은 년도는 언제인가?
3. 출생아수가 혼인건수에 영향이 있는가?

혼인건수와 출생아수 데이터 수집하기

❖ <https://kosis.kr/>

KOSIS
국가통계포털
Korean Statistical Information Service



통합검색 

출생 혼인 



상세검색 

☐ 결과 내 재검색

인기검색어

3 사망원인 

통합검색

통계표

통계분류

온라인간행물

통계설명자료

통계용어

게시물

"출생 혼인"에 대한 검색결과는 725건입니다.

통계표 : 574건 

인구동태건수 및 동태율 추이(출생,사망,혼인,이혼) 

통계청, 인구동향조사, 1970 ~ 2022

월.분기.연간 인구동향(출생,사망,혼인,이혼)

통계청, 인구동향조사, 1981 ~ 2023

시군구/인구동태건수 및 동태율(출생,사망,혼인,이혼) 

통계청, 인구동향조사, 2000 ~ 2022

시도/인구동태건수 및 동태율(출생,사망,혼인,이혼)

통계청, 인구동향조사, 1990 ~ 2022

읍면동/성별/인구동태건수(출생,사망,혼인,이혼)

통계청, 인구동향조사, 2004 ~ 2022

시도/법적혼인상태별 출생

통계청, 인구동향조사, 1981 ~ 2021

1. 혼인건수와 출생아수 데이터 읽어오기

❖ `import pandas as pd` # 데이터 관리와 정제 기능을 가진 라이브러리

❖ *.CSV 데이터 읽어오기

- `변수명 = pd.read_csv('파일경로명' , encoding= '인코딩방식')`
 - ", "로 분리된 .csv 파일을 불러올 때
 - delimiter 옵션은 생략하면 ',' 로 인식
 - 인코딩방식 : 'EUC_KR' (한글이 포함된 일반적인 경우)/ 'cp949' (MS office에서 저장한 파일 형식)

2. 혼인건수와 출생아수 데이터 확인하기

❖ 데이터에서 일부 내용 보기

- 변수명 : 전체 데이터 보기
- 변수명.head() : 위에서 5행 보기 / 변수명.head(3) : 위에서 3행 보기
- 변수명.tail() : 아래서 5행 보기 / 변수명.tail(3) : 아래서에서 3행 보기

❖ 데이터 정보 보기

- 변수명.info() : 데이터 타입, 각 아이템 개수, 누락데이터 수 등 확인
- 변수명.describe() : 수치형 데이터에 대한 통계자료 확인

3. 혼인건수와 출생아수 데이터 재정리하기

❖ 데이터에서 열이름 변경하기

- 변수명.rename(columns = { '열이름' : '새로운 열이름' }, inplace= True)
 - 데이터가 저장된 변수명의 열이름을 새로운 열이름으로 변경
 - inplace = True 옵션은 원본데이터를 변경함

❖ 인덱스 리셋

- 변수명.reset_index(drop=True, inplace=True)
 - drop=True 옵션은 기존 인덱스는 버리고 새로 인덱스 설정

4. 시각적으로 분석하며 질문에 답하기

❖ `import matplotlib.pyplot as plt`
#데이터 시각화를 지원하는 라이브러리

❖ 한글 지원을 위한 코드 실행필요

```
!pip install koreanize-matplotlib  
import koreanize_matplotlib
```

Colab의 경우













5. plot 차트 생성하기

❖ plot 차트 생성하기

- `plt.plot(데이터, 속성들)`
 - 데이터: x축과 y축에 표현할 데이터들
 - plot차트의 경우 y축만으로도 데이터 표현 가능
 - `label='범례이름'`: 선이 여러 개인 경우 각 선의 이름을 차트내에 표시
 - `marker='마커종류'`: 선 위에 표현할 마커 모양 설정
 - `markersize=정수`: 마커의 크기 변경
 - `linestyle='선종류'`: 선의 모양 설정
 - `linewidth=정수`: 선 두께 설정
 - `color='색상'`: 선 색상 설정
 - `color, marker, linestyle` 순으로 약식 표현 가능

plt.plot() 속성들

❖ marker 종류

"."		point
","		pixel
"o"		circle
"v"		triangle_down
"^"		triangle_up
"<"		triangle_left
">"		triangle_right
"1"		tri_down
"2"		tri_up
"3"		tri_left
"4"		tri_right
"8"		octagon
"s"		square

❖ linestyle 종류

종류	Line style
'_'	
'--'	
'-.'	
'...'	

https://matplotlib.org/3.1.1/api/markers_api.html#module-matplotlib.markers

6. 차트 꾸며보기

❖ 차트 추가 함수들

- `plt.show()` :데이터를 이용한 차트 결과를 출력
- `plt.legend()` :지정한 범례(label)를 차트내에 표시
- `plt.title('차트제목')` :지정한 차트제목을 차트 중앙 상단에 표시
- `plt.xlabel('x축제목')`, `plt.ylabel('y축제목')` : x축과 y축의 제목을 지정한 문자열로 설정
- `plt.grid()` : 차트내에 눈금선 표시
- `plt.xlim(시작값, 마지막값)`, `plt.ylim(시작값, 마지막값)` : x, y축 눈금의 범위 지정
- `plt.xticks(눈금값, 레이블)`, `plt.yticks(눈금값, 레이블)` : x, y축에 표시할 눈금 지정
 - **눈금값**: x, y축 눈금에 표시할 값
 - **레이블**: 눈금값으로 표시할 레이블로 튜플이나 리스트로 설정



질문하기(시도 교통사고현황)

❖ 차트를 생성하여 다음의 질문에 답해보자.

1. 사고건수 평균을 기준으로 상위 5위의 도시에 대한 중상자수, 경상자수, 부상신고자수 평균을 비교하시오.
2. 사망자수 평균 비율이 가장 많은/가장 적은 도시는 어디인가?
3. 중상자수 평균 비율이 많은 상위 5개 도시는 어디인가?
4. 모든 지역의 사망자수와 중상자수의 분포를 확인하시오.
5. 지자체 지역의 사고건수와 중상자수의 분포를 시도별로 확인하시오.




시도 구군별 교통사고 데이터 수집하기

❖ <https://www.data.go.kr/data/15070297/fileData.do>

 이 누리집은 대한민국 공식 전자정부 누리집입니다.
DATA 공공데이터포털 .GO.KR 검색어를 입력해 주세요. 



[홈페이지 바로가기](#)

데이터셋

   [URL 복사](#)

도로교통공단_ 시도 시군구별 교통사고 통계

- 경찰에서 조사, 처리한 교통사고에 대한 통계 정보로 인적 피해가 있는 사고만 집계 됨
- 시도 및 시군구별 교통사고 사고건수 사망자수, 중상자수, 경상자수, 부상신고자수 통계
- 교통사고분석시스템(<http://taas.koroad.or.kr>)의 데이터를 바탕으로 함

 0  0 [관심](#)

1. 시도 구군별 교통사고 데이터 읽어오기

❖ `import pandas as pd` # 데이터 관리와 정제 기능을 가진 라이브러리

❖ *.CSV 데이터 읽어오기

- `변수명 = pd.read_csv('파일경로명' , encoding= '인코딩방식')`
 - ", "로 분리된 .csv 파일을 불러올 때 사용
 - 인코딩방식 : 'EUC_KR' (한글이 포함된 일반적인 경우)/ 'cp949' (MS office에서 저장한 파일 형식)

2. 시도 구군별 교통사고 데이터 확인하기

❖ 데이터에서 일부 내용 보기

- 변수명 : 전체 데이터 보기
- 변수명.head() : 위에서 5행 보기 / 변수명.head(3) : 위에서 3행 보기
- 변수명.tail() : 아래서 5행 보기 / 변수명.tail(3) : 아래서에서 3행 보기

❖ 데이터 정보 보기

- 변수명.info() : 데이터 타입, 각 아이템 개수, 누락데이터 수 등 확인
- 변수명.describe() : 수치형 데이터에 대한 통계자료 확인

3. 시도 구군별 교통사고 데이터 재정리하기

❖인덱스 재설정

- 변수명.reset_index(drop=True, inplace=True)
 - drop=True 옵션은 기존 인덱스는 버리고 새로 인덱스 설정

❖데이터 그룹화하여 간단한 통계 확인하기

- 변수명.groupby('그룹열이름') [['열이름1','열이름2']]. 통계함수()
 - 그룹열이름 : 그룹화할 열의 이름
 - 열이름: 그룹별로 통계 데이터를 확인하고자 하는 열의 이름

❖데이터 정렬

- 변수명.sort_values('정렬기준 열이름' , ascending=True)
 - ascending = True:오름차순, False:내림차순, 생략:오름차순

4. 시각적으로 분석하며 질문에 답하기

❖ `import matplotlib.pyplot as plt`
#데이터 시각화를 지원하는 라이브러리

❖ 한글 지원을 위한 코드 실행필요

```
!pip install koreanize-matplotlib  
import koreanize_matplotlib
```

Colab의 경우

5. 차트 생성하기

❖ bar 차트 생성하기

- `plt.bar(데이터, 속성들)`: 세로 막대차트 생성
 - 데이터: x축과 y축에 표현할 데이터들
 - `width=실수`: 막대 두께 설정
- `plt.barh(데이터, 속성들)`: 가로 막대차트 생성
 - `plt.bar()`에 'h'만 추가
 - 속성들은 유사. 단, `width->height, plt.xticks()->plt.yticks()`
 - `height=실수`: 막대 높이 설정
- bar차트와 barh 차트의 공통 속성들
 - `label='범례이름'`: 막대 차트가 여러 개인 경우 각 막대차트의 이름을 차트내에 표시
 - `color='색상'`: 막대차트의 색상 설정
 - `alpha=실수`: 막대의 투명도

5. 차트 생성하기

❖ pie 차트 생성하기

- `plt.pie(데이터, 속성들)`
 - 데이터: 데이터의 구성비를 확인하기 위한 데이터
 - `labels='문자열'`: 각 영역의 값을 나타내는 레이블 문자열
 - `autopct='양식문자'`: 비율을 나타낼 숫자 형식
 - `colors=리스트`: 각 영역의 색상
 - `explode=리스트`: 부채꼴이 중심에서 벗어나는 정도
- `plt.scatter(데이터, 속성들)`
 - 데이터: 분포를 확인하기 위한 열 이름
 - `c='색상'`: 점들의 색상
 - `s='크기'`: 점들의 크기

5. 차트 생성하기

❖seaborn 라이브러리의 relplot 차트

- `sns.relplot(data=표이름, x='x축열이름', y='y축열이름, 속성들');`
 - `data=표이름` : 차트를 생성할 데이터프레임 선택
 - `x='x축열이름'` : 수치형 데이터를 갖는 x축 열이름
 - `y='y축열이름'` : 수치형 데이터를 갖는 y축 열이름
 - `hue='열이름'`: 종류를 구분하기 위한 데이터의 열이름

6. 차트 꾸며보기

❖ 차트 추가 함수들

- `plt.show()` :데이터를 이용한 차트 결과를 출력
- `plt.legend()` :지정한 범례(label)를 차트내에 표시
- `plt.title('차트제목')` :지정한 차트제목을 차트 중앙 상단에 표시
- `plt.xlabel('x축제목')`, `plt.ylabel('y축제목')` : x축과 y축의 제목을 지정한 문자열로 설정
- `plt.grid()` : 차트내에 눈금선 표시
- `plt.xlim(시작값, 종료값)`, `plt.ylim(시작값, 종료값)` : x, y축 눈금의 범위 지정
- `plt.xticks(눈금값, 레이블)`, `plt.yticks(눈금값, 레이블)` : x, y축에 표시할 눈금 지정
 - **눈금값**: x, y축 눈금에 표시할 값
 - **레이블**: 눈금값으로 표시할 레이블로 튜플이나 리스트로 설정