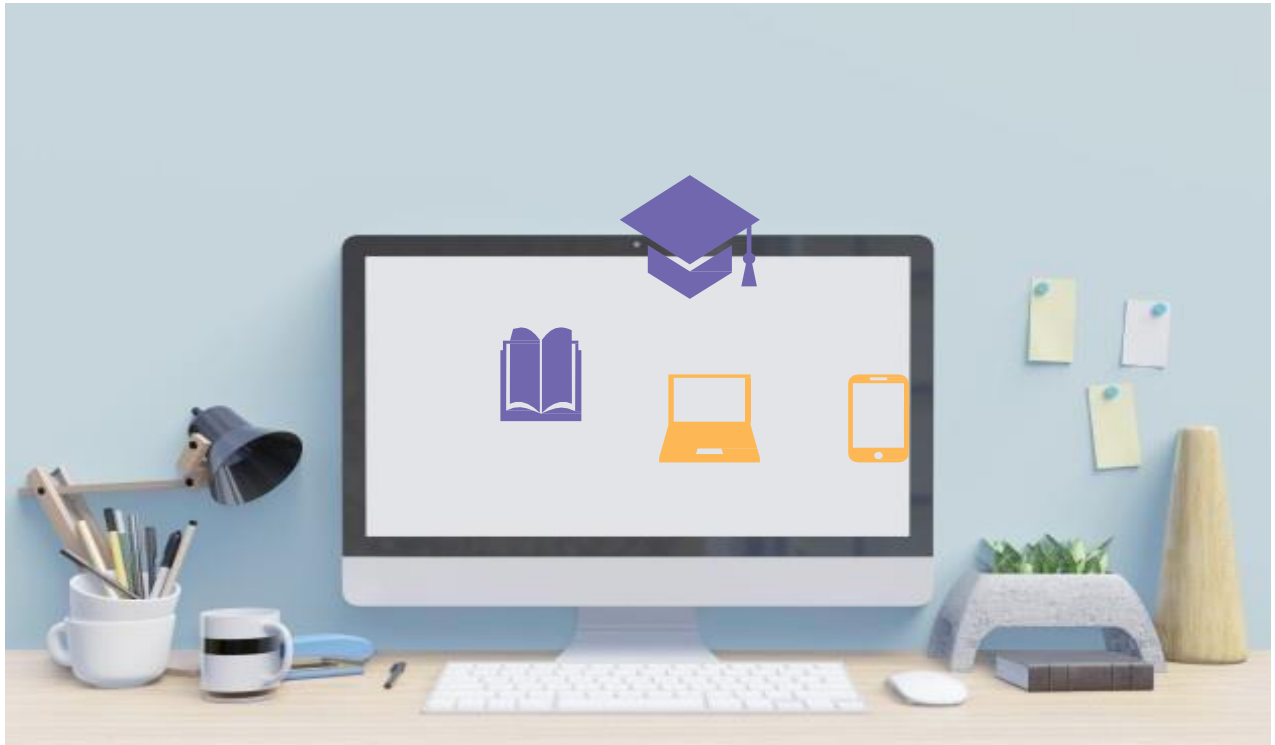
Pandas를 이용한 데이터분석

팀명: 비주얼 마이너
1조 : 안서영 유제승 박소현 이현민



미래의 시작!!!
동양미래대학교
컴퓨터소프트웨어공학과

데이터 분석이란?

- 유용한 정보를 발굴하고 결론 내용을 알리며 의사결정을 지원하는 것을 목표로 데이터를 정리, 변환, 모델링하는 과정

- 특히, 오늘날 비즈니스 부문에서 데이터 분석은 의사 결정을 더 과학적으로 만들어주고 비즈니스를 더 효율적으로 운영할 수 있도록 도와주는 역할을 함

데이터 분석 도구의 종류

1. 파이썬 (Python)
2. R
3. 엑셀 (Excel)
4. SQL (Structured Query Language, 구조화 질의어)
5. 태블로 (Tableau)
6. Power BI
7. 구글 애널리틱스 (Google Analytics, GA)

NumPy 란?

수학 및 과학 연산을 위한 Python 패키지

- Python에서 과학적 계산을 위한 핵심 라이브러리
- C언어 및 포트란으로 작성되어 실행 속도가 빠르다.
- 기본적으로 array라는 자료를 생성하고 기능을 수행한다.
- 수치해석, 통계 관련 기능을 구현 시 기본이 되는 모듈이다.
- 주로 np로 호출하여 사용한다. (import numpy as np)

Python List와 NumPy Array의 차이점

Python List	NumPy Array
여러가지 타입의 원소	동일 타입의 원소
메모리 용량이 크고 속도가 느림	메모리 최적화, 계산속도 향상
전체 연산 불가	전체 연산 가능

Python List와 NumPy Array의 차이점

1. List와 Numpy array의 자료타입 차이

```
[13]  1 p_list = [1,2,3,4,'?']
      2 p_list

      [1, 2, 3, 4, '?']

[14]  1 np.array([1,2,3,4,'?'])
      2

      array(['1', '2', '3', '4', '?'], dtype='<U21')
```

Python list는 숫자형이나 문자형 여러 가지 자료형을 한 번에 다룰 수 있다.
Numpy array는 한 가지 동일한 자료형 이어야 한다. (숫자형 + 문자형 = 모두 문자형으로 전환)
단, Numpy array를 사용하기 전에 import numpy as np 선언해주기

Python List와 NumPy Array의 차이점

2. List 와 Numpy array 연산의 차이

```
[15]  1 p_list = [1,2,3,4]
      2 p_list2 = [5,6,7,8]
      3
      4 p_list + p_list2
```

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

List :연산 시 연결이 된다.

```
[16]  1 arr = np.array([1,2,3,4])
      2 arr2 = np.array([5,6,7,8])
      3
      4 arr + arr2
```

```
array([ 6,  8, 10, 12])
```

Numpy array : 두 값에서 동일한 위치에 있는 원소별로 연산 가능
단, 원소의 개수가 동일해야 한다.

Python List와 NumPy Array의 차이점

3. List 와 Numpy array 배열 데이터 추가

```
[23]  1 p_list.append(5)
      2 p_list
```

```
[1, 2, 3, 4, 5, 5]
```

```
[24]  1 arr = np.append(arr, np.array([5]))
      2 arr
```

```
array([1, 2, 3, 4, 5])
```

List :
append 메소드에 인자 값을 넣으면 가능

Numpy array :
append 메소드를 사용하는 것은 동일하나
배열 이름과 추가 데이터가 필요하다.

Pandas 란?

Python 데이터 분석 라이브러리

- 패널 데이터(계량 경제 용어)와 Python 데이터 분석의 이름을 따서 명명
- 데이터를 구조화 하고 처리하는데 매우 편리
- 행과 열로 이루어진 2차원 데이터프레임 (Data Frame 형식)
- 최적화된 데이터
- 데이터 분석 또는 머신 러닝 분야에 사용

Pandas 주요 함수

- 데이터 로드 및 저장 : read_csv(), read_excel(), read_html(), to_csv(), to_excel()
- DataFrame 데이터 확인 : df.shape, df.info(), df.columns, df.dtypes, df.head(), df.tail()
- 특정 값 세기 : value_counts()
- 특정 컬럼 순으로 정렬하기 : sort_values()
- 각 컬럼마다 Null 개수 확인 : df.isnull().sum()
- 각 컬럼마다 Null 비율 : df.isnull.sum() / df.shape[0]
- 중복 확인 : df.duplicated()
- 중복 제거 : df.drop_duplicates()

Pandas 주요 구성 요소

- Series – 1개의 컬럼 값 만으로 1차원 데이터 셋
- DataFrame – Column x Row 2차원 데이터 셋
- Index – Series와 DataFrame의 고유한 Key 값

▪ **Series**

	A	B	C	D	E
1		이름	수학	국어	과학
2	1	둘리	84	87	78
3	2	호이	21	15	84
4	3	영미	87	84	76
5	4	길동	100	87	99
6	5	또치	59	99	59
7	6	마이클	46	77	56

index Values(1차원)

▪ **DataFrame**

columns

	A	B	C	D	E
1		이름	수학	국어	과학
2	1	둘리	84	87	78
3	2	호이	21	15	84
4	3	영미	87	84	76
5	4	길동	100	87	99
6	5	또치	59	99	59
7	6	마이클	46	77	56

index values(2차원)

pip 란?

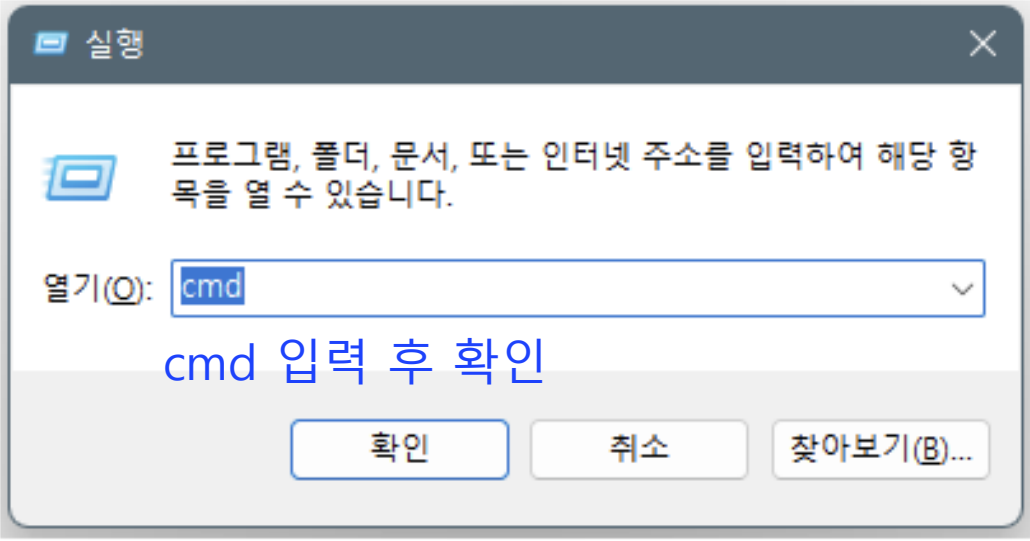
파이썬 모듈이나 패키지를 쉽게 설치할 수 있도록 도와주는 도구

- 파이썬의 표준 라이브러리가 아닌 것을 설치하기 위해서
- pip으로 프로그램 설치 시 의존성 있는 모듈이나 패키지를 함께 설치해주기 때문에 사용
  ex) B 패키지 설치 시 A 라는 패키지가 먼저 설치되어야 한다는 규칙이 있을 때 pip을 사용하여 B라는 패키지 설치하면 자동으로 A패키지도 설치해준다.

- 패키지 설치 : pip install 패키지명
- 패키지 삭제 : pip uninstall 패키지명
- 특정 버전의 패키지 설치 : pip install 패키지명==버전명
- 패키지 업그레이드 : pip install –upgrade 패키지명
- 설치된 패키지 확인 : pip list

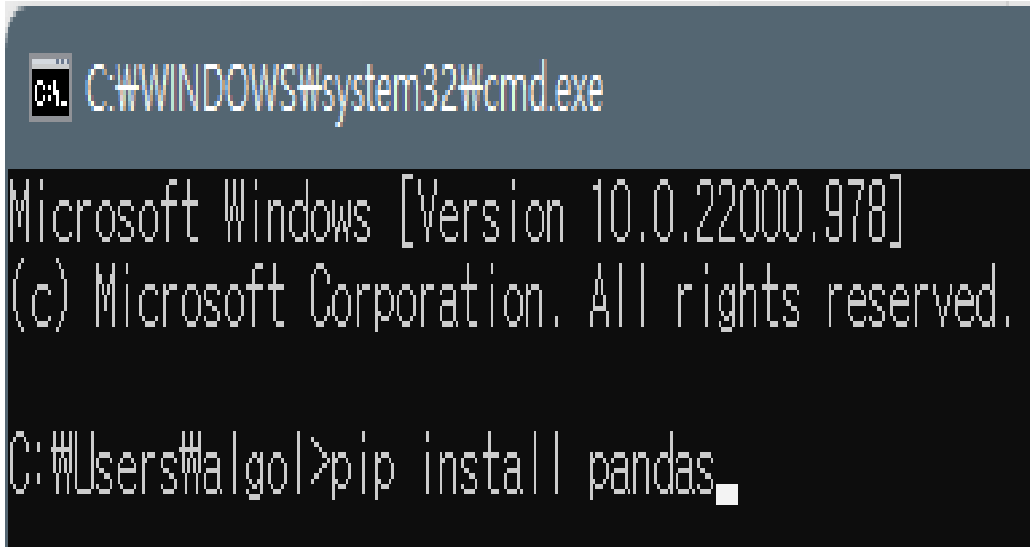
pip 을 이용하여 Pandas 설치하기

명령 프롬프트 실행하기
(Window + R)



cmd 입력 후 확인

명령어 입력하기



pip install pandas

데이터 시각화가 필요한 이유

- 통계 분석은 데이터 집합의 변수가 서로 관련되는 방식과 이러한 관계가 다른 변수에 의존하는 방식을 이해하는 프로세스
- 데이터가 제대로 시각화 되면 인각 시각 시스템에서 관계를 나타내는 추세와 패턴을 확인 가능

데이터 시각화란?

- 특이값을 찾아내거나 데이터 변형이 필요한지 알아보거나 모델에 대한 아이디어를 찾기 위한 과정

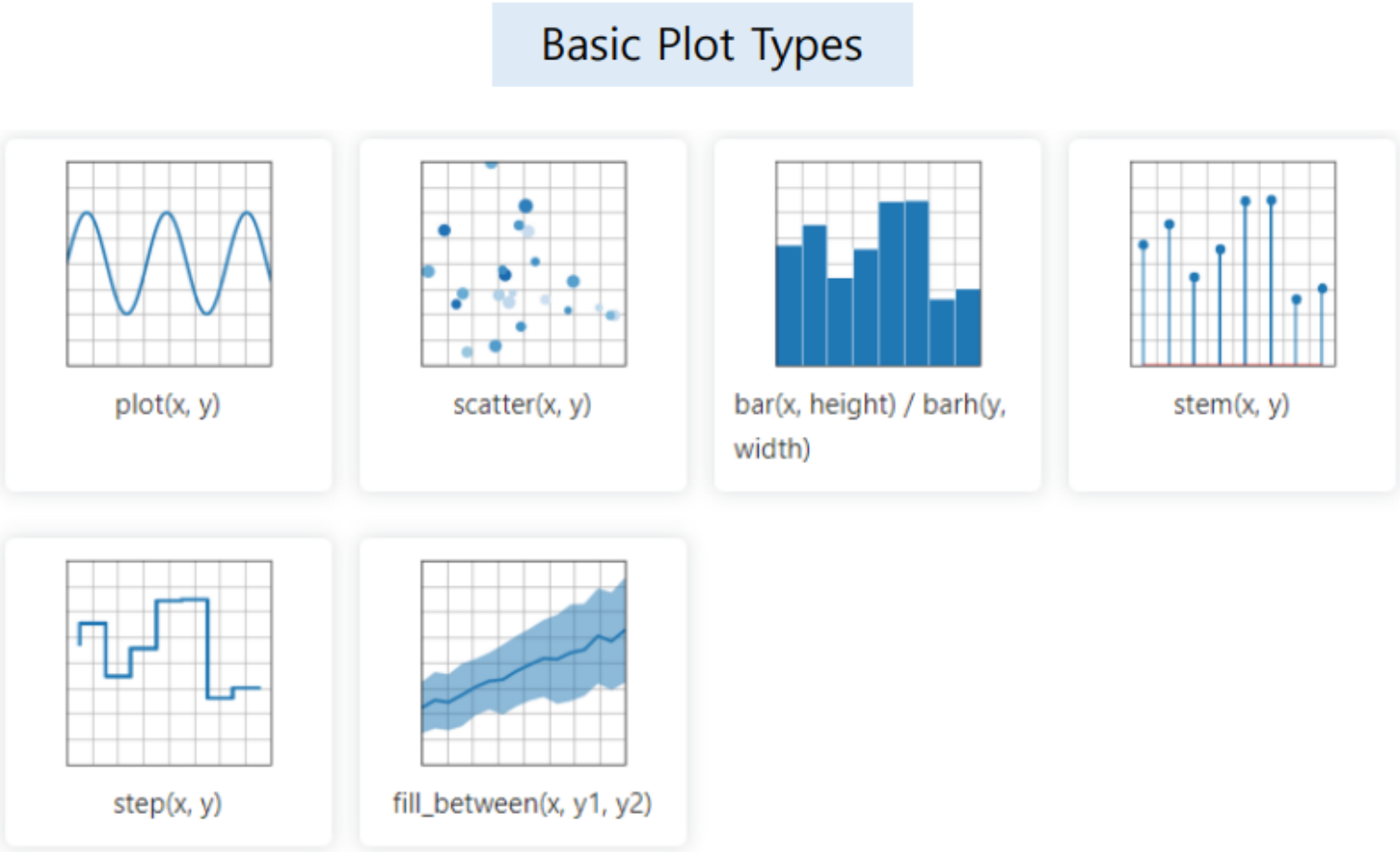
Matplotlib 란?

함수를 이용해 데이터를 간편하게 그래프로 만들고 변화 줄 수 있는 패키지

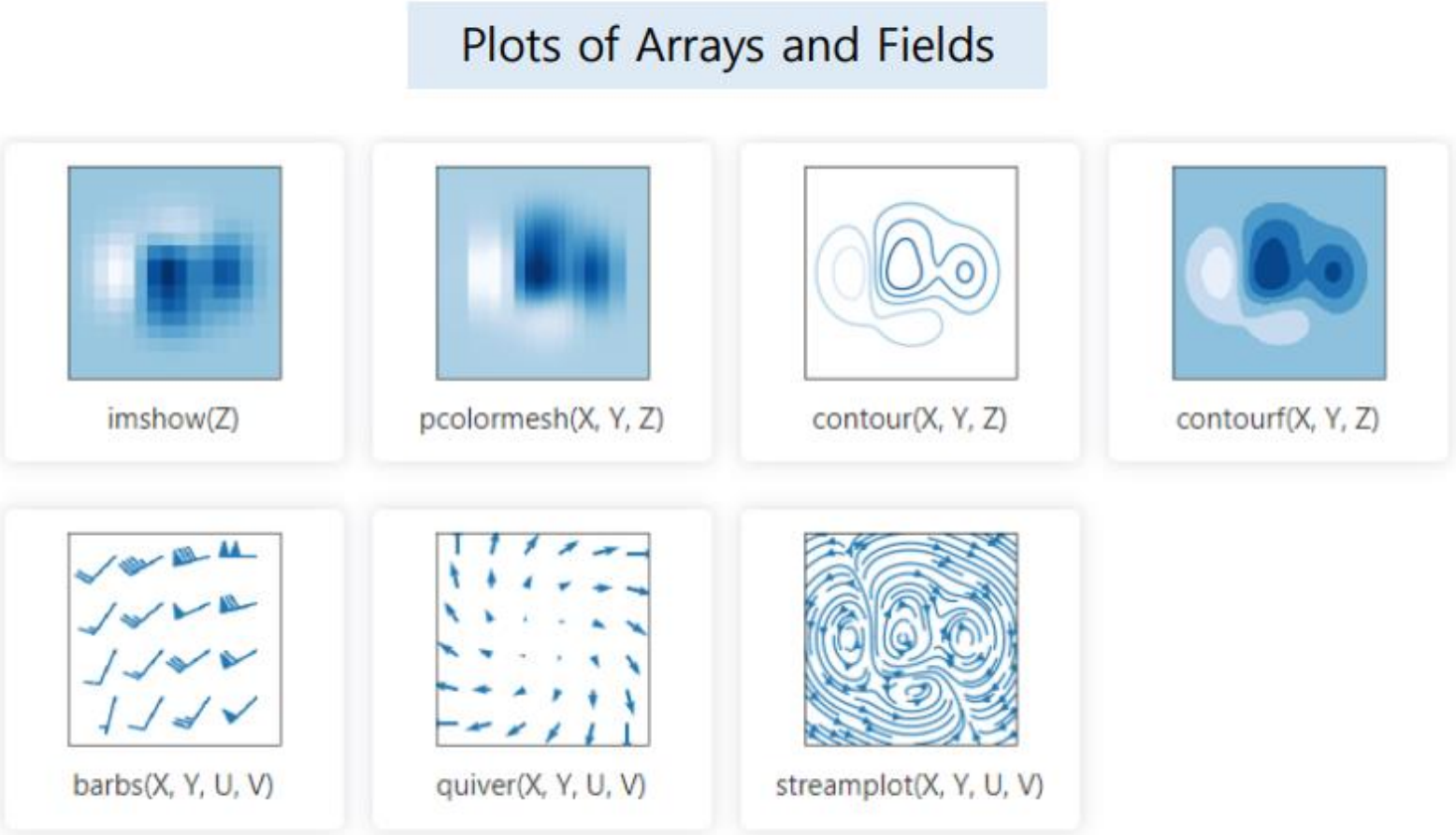
- 데이터를 시각적으로 표현 가능하다.
- 시각적인 자료라 이해하기 쉽다.
- 주로 2D 도표를 위한 데스크톱 패키지이다.
- 저수준(세밀한)의 그래프 그리기 작성이 가능하다.
- import 시 matplotlib.pyplot 로 불러온다.

  ex) import matplotlib.pylot as plt (이름이 너무 길어 plt라는 명으로 바꿔주기)

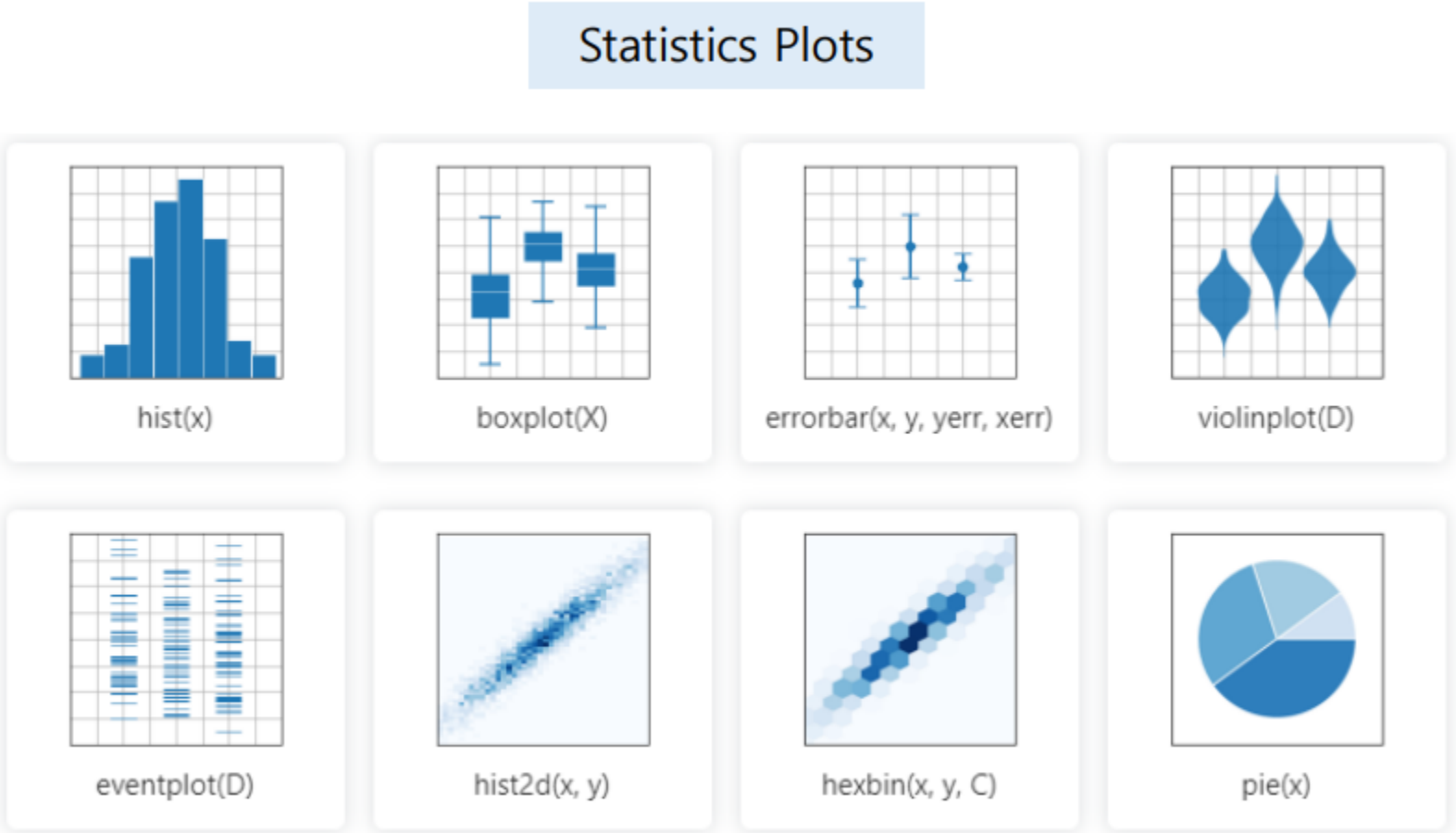
Matplotlib의 그래프 표기



Matplotlib의 그래프 표기



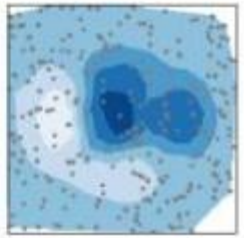
Matplotlib의 그래프 표기



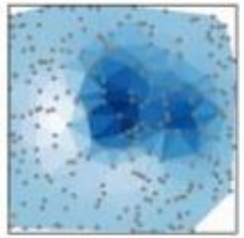
Matplotlib의 그래프 표기

Unstructured Coordinates

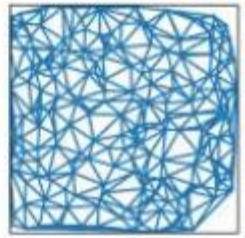
tricontour(x, y, z)



tricontourf(x, y, z)



tripcolor(x, y, z)



triplot(x, y)

패키지 설치하기 (실습)

명령 프롬프트를 활용하여 numpy + matplotlib 패키지 설치하기

1. 명령 프롬프트를 실행한다 : Window + R
2. 명령어를 작성한다 : pip install numpy

```
C:\Users\Chidum.Osobalu>pip3 install numpy
Collecting numpy
  Downloading numpy-1.19.1-cp38-cp38-win32.whl (10.9 MB)
     |████████████████████████████████| 10.9 MB 22 kB/s
Installing collected packages: numpy
Successfully installed numpy-1.19.1
```

3. 설치 여부를 확인한다 : pip show numpy

```
C:\Users\Chidum.Osobalu>pip3 show numpy
Name: numpy
Version: 1.19.1
Summary: NumPy is the fundamental package for array computing with Python.
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email: None
```

1. 명령 프롬프트를 실행한다 : Window + R
2. 명령어를 작성한다 : pip install matplotlib

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.22000.708]
(c) Microsoft Corporation. All rights reserved.

C:₩Users₩bette>pip install matplotlib
```

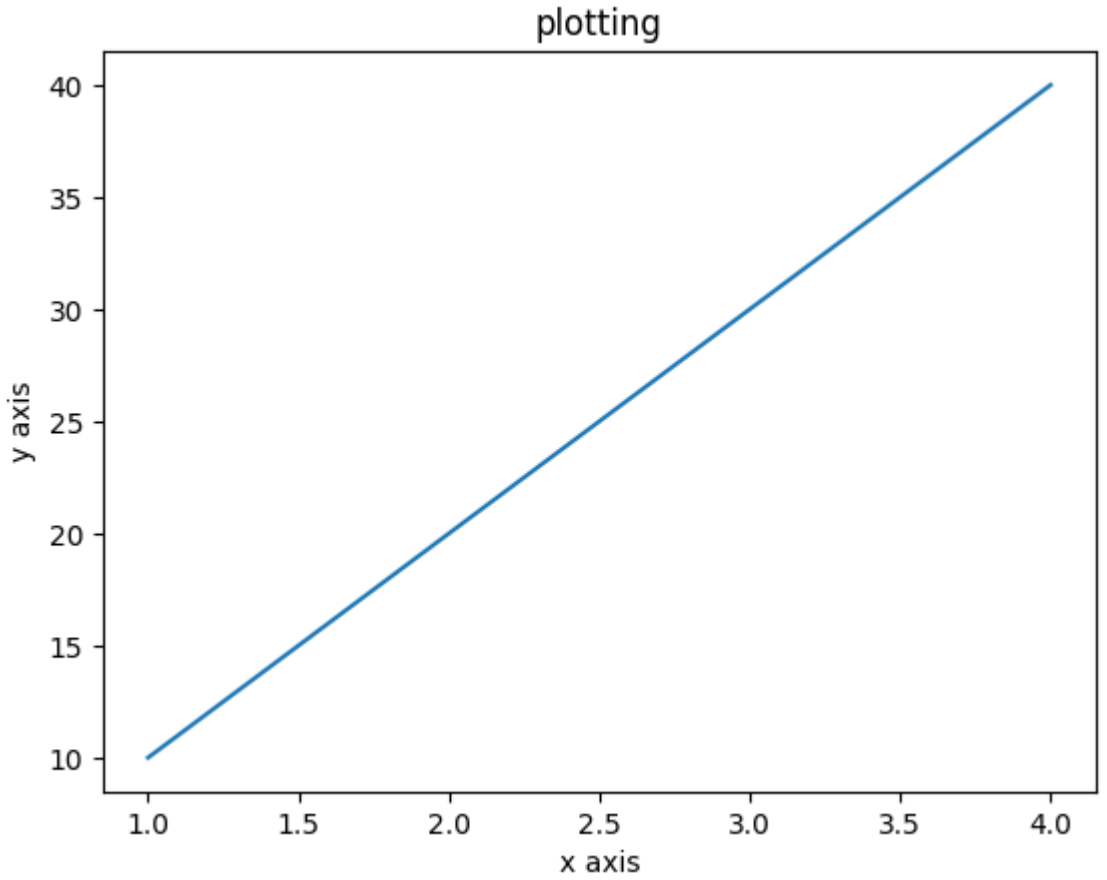
3. 설치 여부를 확인한다 : pip show matplotlib

```
C:\Users\Aryan>pip show matplotlib
Name: matplotlib
Version: 3.5.2
Summary: Python plotting package
Home-page: https://matplotlib.org
```

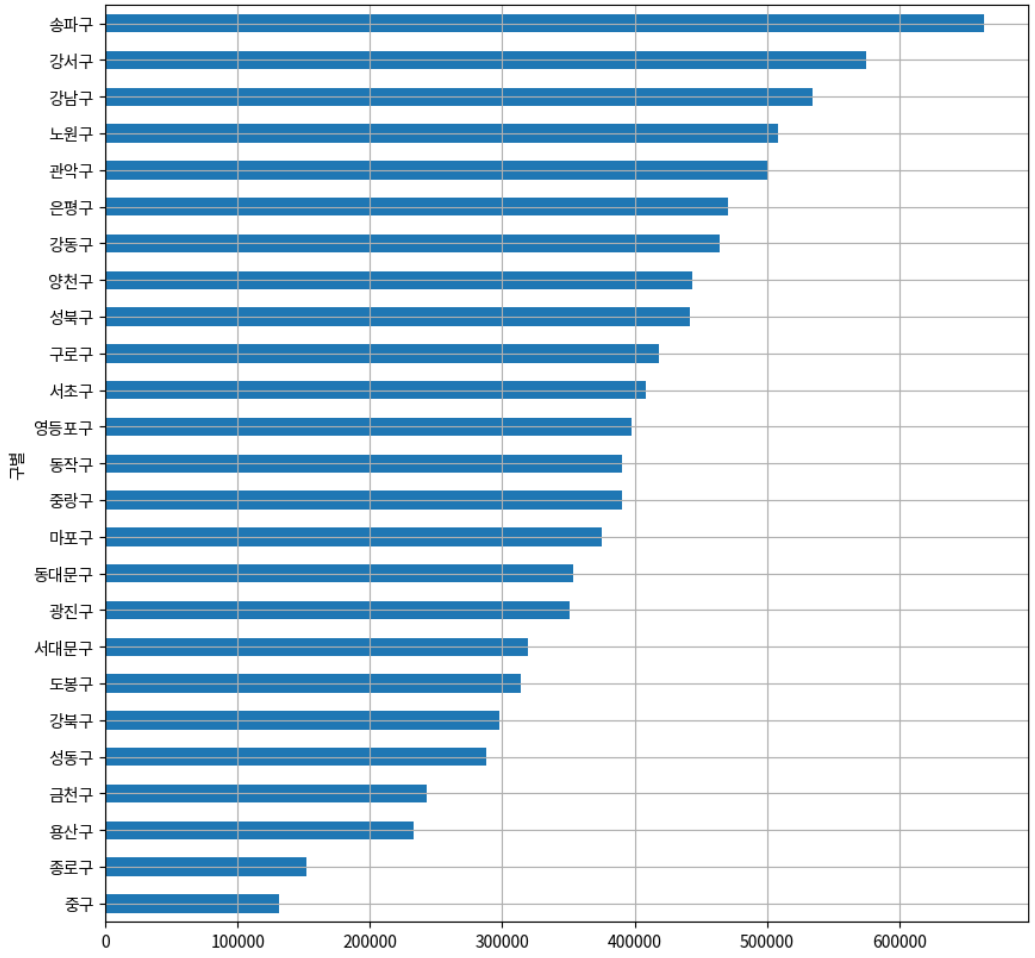
Matplotlib의 그래프 실습

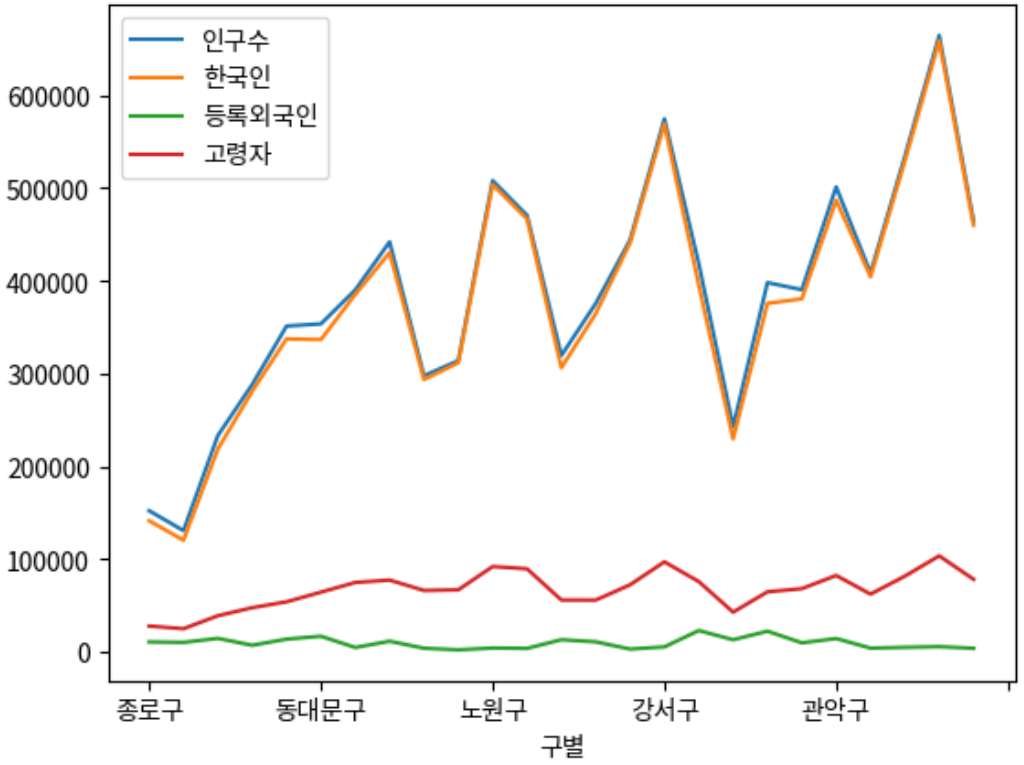
Plot 함수로 선 그래프 그리기

```
import matplotlib.pyplot as plt

plt.plot([1,2,3,4],[10,20,30,40])
plt.title('plotting’) # 타이틀명
plt.xlabel('x axis’)  # x축 이름
plt.ylabel('y axis’)  # y축 이름

plt.show()
```



수업 내용 정리 실습


송파구
강서구
강남구
노원구
관악구
은평구
강동구
양천구
성북구
구로구
서초구
영등포구
동작구
중랑구
마포구
동대문구
광진구
서대문구
도봉구
강북구
성동구
금천구
용산구
종로구
중구
구별
0
100000
200000
300000
400000
500000
600000


인구수
한국인
등록외국인
고령자
600000
500000
400000
300000
200000
100000
0
종로구
동대문구
노원구
강서구
관악구
구별

Python 정규식(Regular Expression)

문자열을 처리하는 방법 중 하나로 "특정 조건 또는 패턴"을 치환하는 과정을 쉽게 처리할 수 있는 방법.

패턴	설명	예제
^	이 패턴으로 시작해야 함	^abc : abc로 시작해야 함 (abcd, abc12 등)
$	이 패턴으로 종료되어야 함	xyz$: xyz로 종료되어야 함 (123xyz, strxyz 등)
[문자들]	문자들 중에 하나이어야 함. 가능한 문자들의 집합을 정의함.	[Pp]ython : "Python" 혹은 "python"
[^문자들]	[문자들]의 반대로 피해야할 문자들의 집합을 정의함.	[^aeiou] : 소문자 모음이 아닌 문자들
|	두 패턴 중 하나이어야 함 (OR 기능)	a | b : a 또는 b 이어야 함
?	앞 패턴이 없거나 하나이어야 함 (Optional 패턴을 정의할 때 사용)	\d? : 숫자가 하나 있거나 없어야 함
+	앞 패턴이 하나 이상이어야 함	\d+ : 숫자가 하나 이상이어야 함
*	앞 패턴이 0개 이상이어야 함	\d* : 숫자가 없거나 하나 이상이어야 함
패턴{n}	앞 패턴이 n번 반복해서 나타나는 경우	\d{3} : 숫자가 3개 있어야 함
패턴{n, m}	앞 패턴이 최소 n번, 최대 m 번 반복해서 나타나는 경우 (n 또는 m 은 생략 가능)	\d{3,5} : 숫자가 3개, 4개 혹은 5개 있어야 함
\d	숫자 0 ~ 9	\d\d\d : 0 ~ 9 범위의 숫자가 3개를 의미 (123, 000 등)
\w	문자를 의미	\w\w\w : 문자가 3개를 의미 (xyz, ABC 등)
\s	화이트 스페이스를 의미하는데, [\t\n\r\f] 와 동일	\s\s : 화이트 스페이스 문자 2개 의미 (\r\n, \t\t 등)
.	뉴라인(\n) 을 제외한 모든 문자를 의미	.{3} : 문자 3개 (F15, 0x0 등)

메타 문자

: 원래 그 문자가 가진 뜻이 아닌 특별한 용도로 사용하는 문자

```
# 메타 문자
. ^ $ * + ? { } [ ] \ | ( )
```

re 라이브러리

파이썬에서 정규표현식을 처리하기 위한 다양한 기능을 제공.

주요 기능 (메서드)

- compile : 정규식을 컴파일. 같은 패턴을 여러 번 사용 시 성능 향상.
- search : 문자열에서 정규식과 일치하는 첫 번째 부분 탐색. 일치하는 부분이 없을 시 None 리턴.
- match : 문자열의 시작 부분에서 패턴과 일치하는 부분 탐색. 일치하는 부분이 없을 시 None 리턴.
- findall : 패턴과 매치되는 모든 부분을 찾아 리스트로 반환.
- sub(pattern, repl, tring, count=0) : string에서 pattern과 일치하는 부분을 replace 텍스트로 교체. count를 사용하여 교체 횟수 제한.
- split(pattern, string, maxsplit=0) : pattern을 기준으로 string을 분리하고 분리된 문자열 리스트를 반환. maxsplit를 사용하여 분할 횟수를 제한.

re 라이브러리 실습

1. 다음 문장을 이용하여 결과값이랑 동일하게 만드시오.

"제 이메일 주소는 example@gmail.com 이고, 제 친구의 이메일 주소는 friend@gmail.com 입니다."

```
['example@gmail.com', 'friend@gmail.com']
```

re 라이브러리 실습

1. 다음 문장을 이용하여 결과값이랑 동일하게 만드시오.

```
import re

text = '제 이메일 주소는 example@gmail.com 이고,
제 친구의 이메일 주소는 friend@gmail.com 입니다.’

email_pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z:a-z]{2,}\b'
email_addresses = re.findall(email_pattern,text);

print(email_addresses)
```