

streamlit 대시보드 강의

with  python™

배포금지

강의 대상 및 목적

with  python™

입문자

국비교육생

취업준비생

포트폴리오

Google Cloud Platform

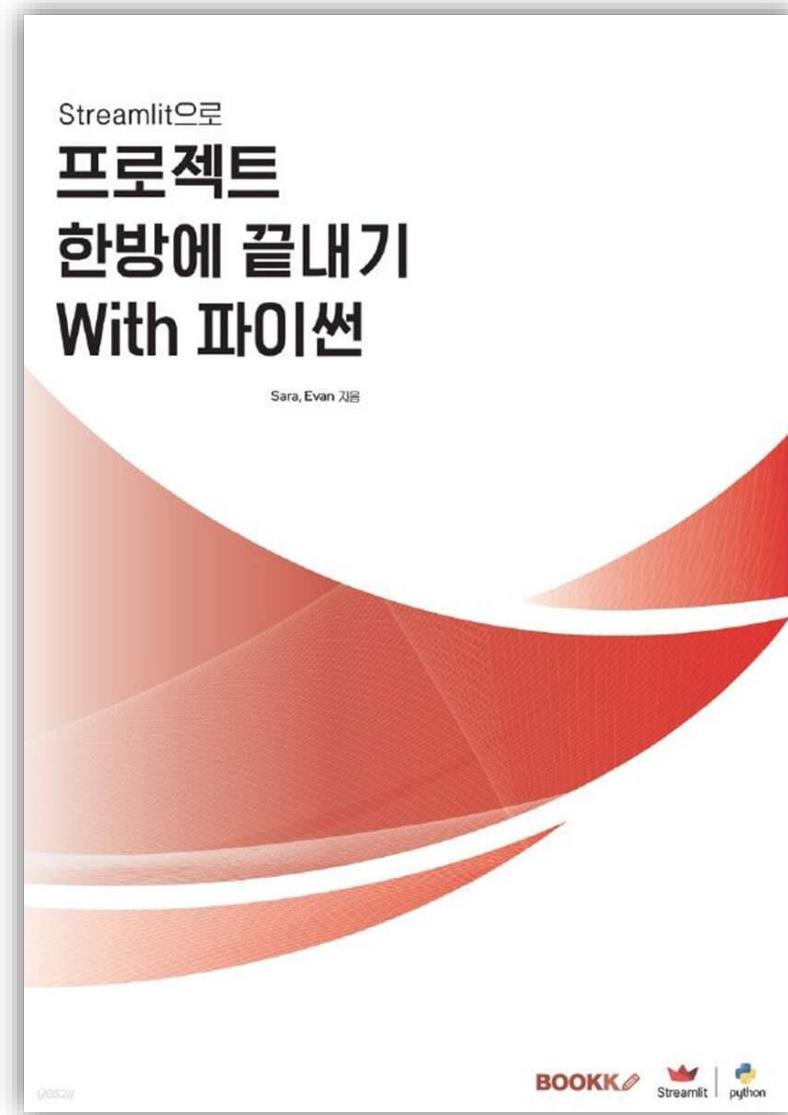
Streamlit 대시보드

기초통계, 머신러닝

파이썬 기초 문법

03. 주요 교재

(2023년 11월 기준)



- ✓ 2023년 5월 30일 출간 (초판)
 - 파이썬 기초문법
 - 기초통계 및 머신러닝
 - Streamlit 대시보드

- ✓ 2024년 1월 개정판
 - Google Cloud Platform 배포
 - Github Actions

강사 소개

with  python™

01. 강사 소개

(2023년 11월 기준)

- ✓ INTJ (혼자 있을 때 힘이 충전됨)
- ✓ 학점은행 경영학사 (2009)
- ✓ 한동대학교 국제개발협력대학원 (2021)
- ✓ 국민대학교 비즈니스IT전문대학원 박사과정 (2023 ~)
- ✓ 2023년 기준 강의경력 4년차
 - 공기관 강의
 - 금융권 강의
 - 재직자 대상 강의
 - 취업준비생 대상 강의 (★ ★ ★)
- ✓ 오프라인 누적 강의 시간(2023년 12월 기준 5000시간 ↑)
 - 월평균 150시간 x 3년

02. 저서

(2023년 11월 기준)

- ✓ (KCI 등재) 필리핀 스타트업의 기업가적 지향성과 기업성과에 관한 연구:
사회적 자본의 매개 효과(2021, 한국벤처창업학회)
 - 주요 분석 방법론 : 구조방정식(R, PLS-SEM)
- ✓ 파이썬 캐글 뽀개기(2021, 비제이퍼블릭)
- ✓ Streamlit으로 프로젝트 한방에 끝내기 with 파이썬(2023, 부크크)

개발환경설정

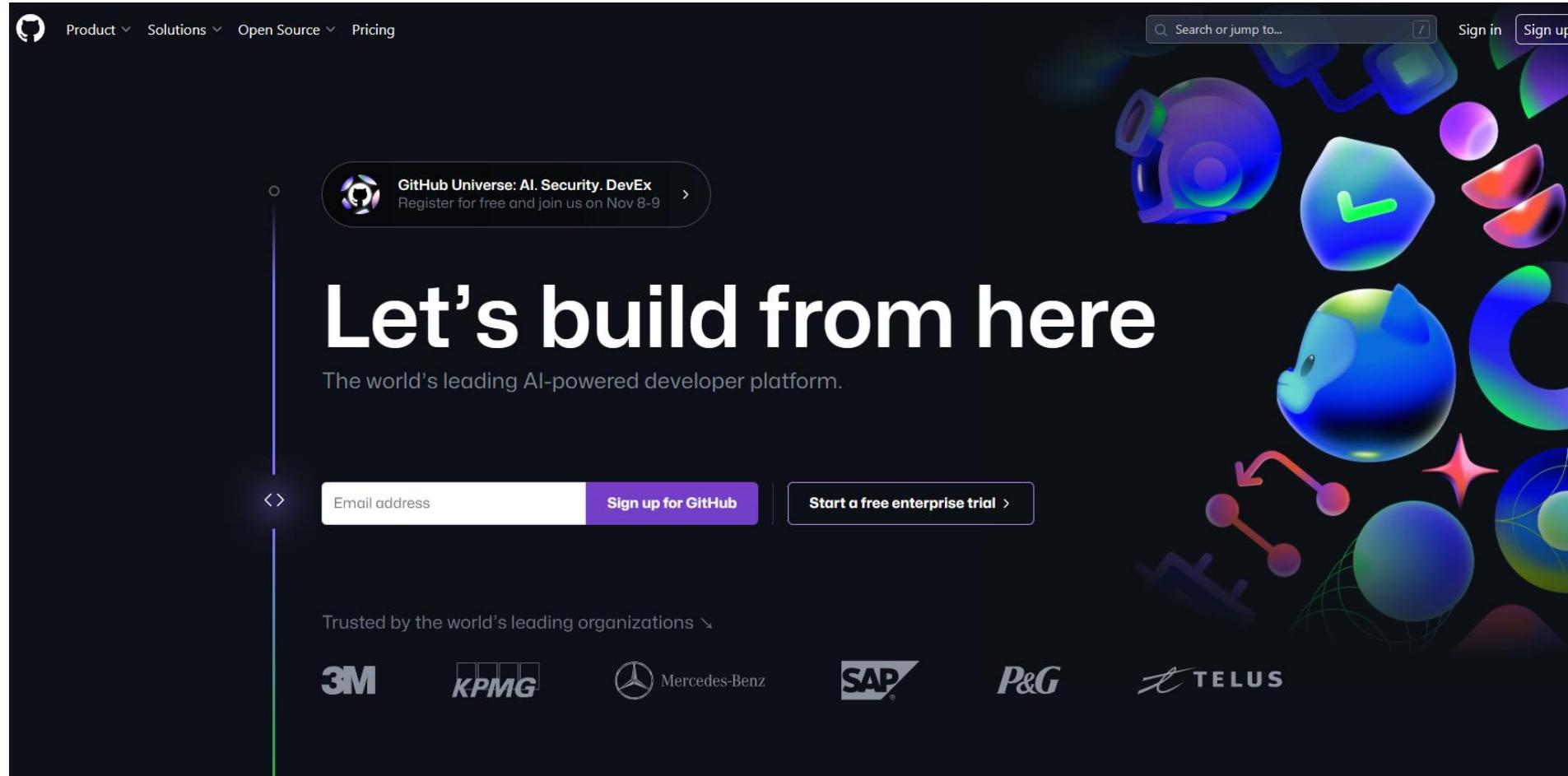
with  python™

01. Github 설정

(2023년 11월 기준)

◆ Github : Git을 사용한 버전 관리 및 협업을 위한 웹 기반 플랫폼

- 사이트 : <https://github.com/>



01. Github 설정

(2023년 11월 기준)

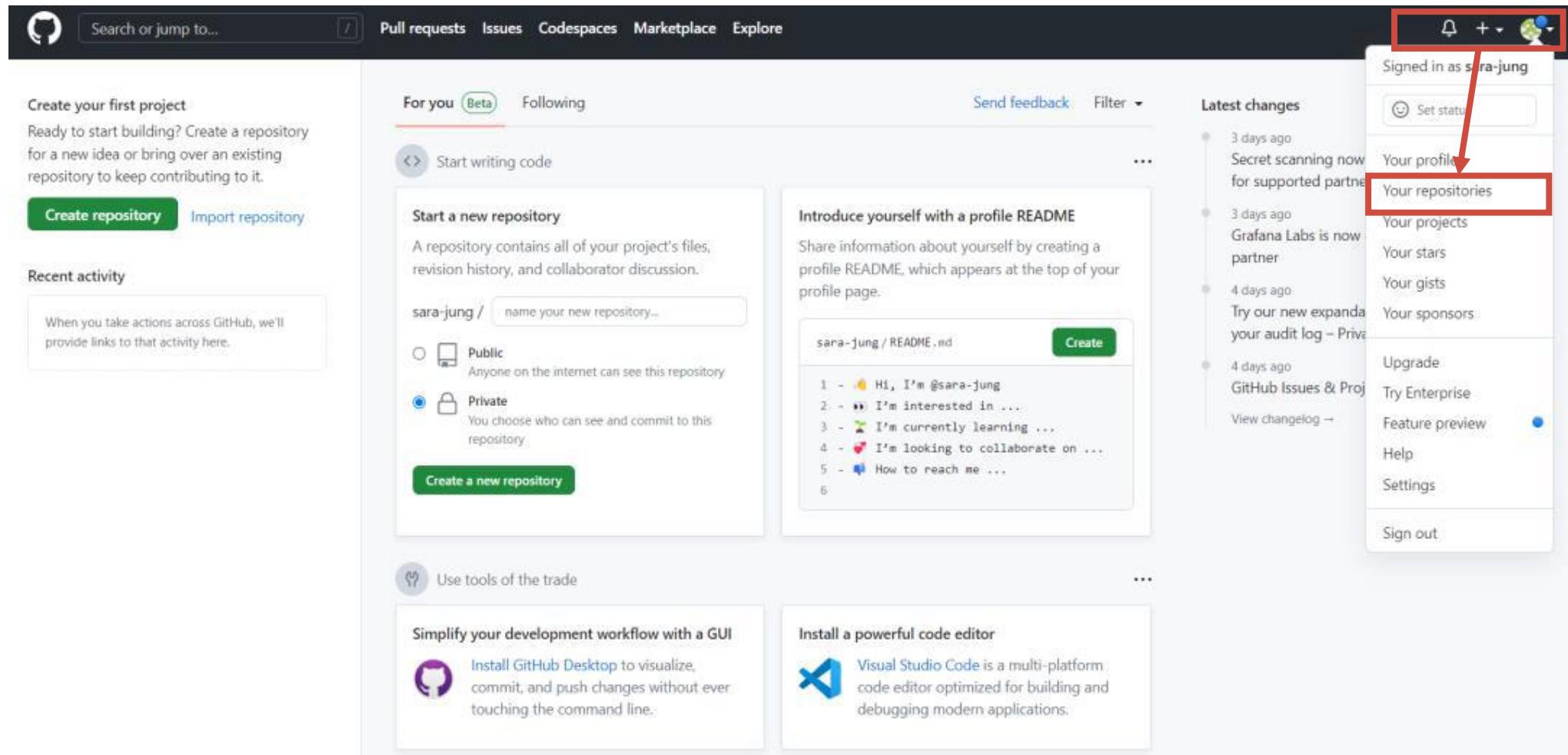
◆ 회원가입이 완료된 후, Create Repository 클릭

The screenshot shows the GitHub homepage with a dark theme. At the top, there is a navigation bar with links for Pull requests, Issues, Codespaces, Marketplace, and Explore. Below the navigation bar, there is a search bar and a user profile icon. On the left side, there is a sidebar with a 'Create your first project' section containing a 'Create repository' button (which is highlighted with a red box) and an 'Import repository' link. Below this, there is a 'Recent activity' section. The main content area features several cards: 'Start a new repository' (with a 'Create a new repository' button highlighted with a red box), 'Introduce yourself with a profile README' (showing a sample README file with a 'Create' button), 'Use tools of the trade' (with cards for 'Simplify your development workflow with a GUI' and 'Install a powerful code editor'), and a 'Latest changes' sidebar on the right.

01. Github 설정

(2023년 11월 기준)

◆ (또는) 우측 상단의 Profile icon 클릭 > Your repositories 클릭



01. Github 설정

(2023년 11월 기준)

- ◆ (또는) 우측 상단의 Profile icon 클릭 > Your repositories 클릭

sara-jung

Edit profile

Joined 5 minutes ago

Overview Repositories Projects Packages Stars

Find a repository... Type Language Sort

New

sara-jung doesn't have any public repositories yet.



© 2023 GitHub, Inc.

Terms

Privacy

Security

Status

Docs

Contact GitHub

Pricing

API

Training

Blog

About

강의 실습 영상 참고

02. Streamlit 회원가입

(2023년 11월 기준)

- ◆ Streamlit은 머신러닝 및 데이터 사이언스 프로젝트 위한 대시보드 오픈소스 라이브러리
 - 사이트 : <https://streamlit.io/>

The screenshot shows the Streamlit homepage. At the top, there is a banner with the text "Join the Streamlit Hackathon for Snowflake Summit — now extended to May 5th! Learn more here." Below the banner, there is a navigation bar with links for "Cloud", "Gallery", "Components", "Community", "Docs", and "Blog". On the right side of the navigation bar are "Sign in" and "Sign up" buttons. The main title "A faster way to build and share data apps" is prominently displayed in large, bold, black font. Below the title, there is a subtitle "Streamlit turns data scripts into shareable web apps in minutes." followed by the text "All in pure Python. No front-end experience required." At the bottom, there are two buttons: a red "Sign up for Community Cloud" button and a white "Install Streamlit" button.

03. Git 설치

(2023년 11월 기준)

◆ Git은 소스 코드의 변경 사항을 추적하기 위한 분산 버전 제어 시스템

- 사이트 : <https://git-scm.com/downloads>

The screenshot shows the official Git website at <https://git-scm.com/>. The main navigation bar includes links for About, Documentation, Downloads, and Community. The Downloads section is highlighted. It features a large "Downloads" heading and three download links for macOS, Windows, and Linux/Unix. A prominent callout box highlights the "Latest source Release" which is version 2.42.1, released on 2023-11-02. A "Download for Windows" button is shown. Below this, there's a note about older releases and a GitHub link. The page also includes sections for "GUI Clients" and "Logos".

git --distributed-is-the-new-centralized

About Documentation Downloads Community

Downloads

macOS Windows

Linux/Unix

Latest source Release
2.42.1
Release Notes (2023-11-02)
Download for Windows

Older releases are available and the Git source repository is on GitHub.

GUI Clients

Git comes with built-in GUI tools (`git-gui`, `gitk`), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

강의 실습 영상 참고

04. Python 설치

(2023년 11월 기준)

- ◆ Python은 웹 개발, 데이터 분석, 머신 러닝 등 다양한 작업에 사용되는 프로그래밍 언어
 - 사이트 : <https://www.python.org/>

The screenshot shows the Python.org homepage with a dark blue header. The navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. The main content area features the Python logo and a search bar with a 'GO' button. A sidebar on the left displays a Python code snippet for calculating Fibonacci numbers:

```
# Python 3: Fib
>>> def fib(n):
>>>     a, b =
>>>     while a
>>>         pri
>>>         a,
>>>         print()
>>>     fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

The sidebar also lists 'All releases', 'Source code', 'Windows', 'macOS', 'Other Platforms', 'License', and 'Alternative Implementations'. The central content area is titled 'Download for Windows' and offers Python 3.12.0. It includes a note that Python 3.9+ cannot be used on Windows 7 or earlier, and links to 'View the full list of downloads' and 'More about'.

Python is a programming language that lets you work quickly
and integrate systems more effectively. [» Learn More](#)

04. Python 설치

(2023년 11월 기준)

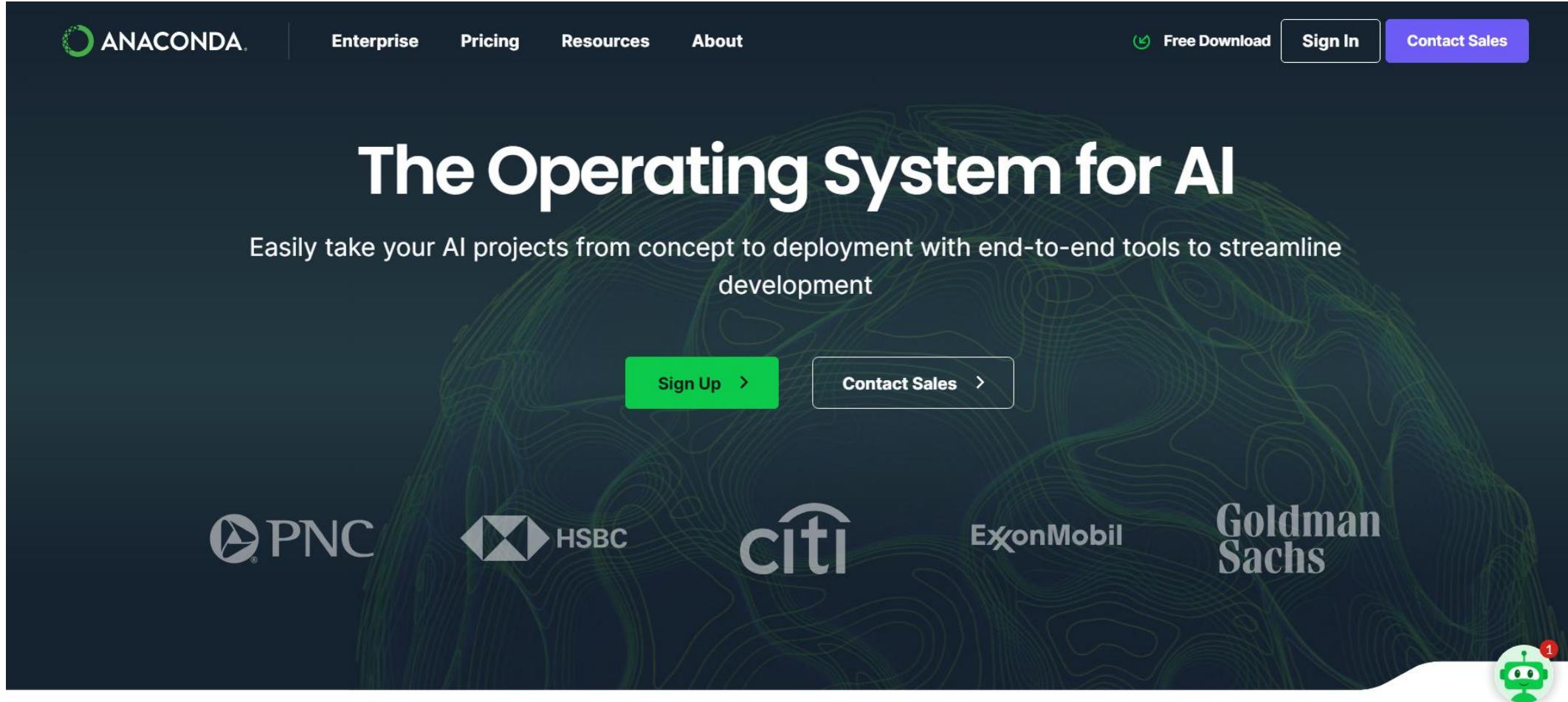
- ◆ Python은 웹 개발, 데이터 분석, 머신 러닝 등 다양한 작업에 사용되는 프로그래밍 언어
 - 사이트 : <https://www.python.org/>

강의 실습 영상 참고

05. Anaconda 설치

(2023년 11월 기준)

- ◆ Anaconda 데이터 과학 및 머신 러닝 작업을 위한 오픈 소스 플랫폼
 - 사이트 : <https://www.anaconda.com/>



05. Anaconda 설치

(2023년 11월 기준)

- ◆ Anaconda 데이터 과학 및 머신 러닝 작업을 위한 오픈 소스 플랫폼
 - 사이트 : <https://www.anaconda.com/>

강의 실습 영상 참고

06. miniconda 설치

(2023년 11월 기준)

- ◆ miniconda는 아나콘다의 경량 버전으로, 데이터 과학 및 패키지 관리를 위한 필수 도구를 제공하는 작은 배포 버전
 - 사이트 : <https://docs.conda.io/projects/miniconda/en/latest/>

The screenshot shows the official Miniconda documentation page for the latest version. The header includes the Miniconda logo and navigation links for 'Search docs', 'System requirements', 'Latest Miniconda installer links by Python version', 'Installing Miniconda', 'Miniconda release notes', and 'Other resources'. The main content area is titled 'Miniconda' and describes it as a free minimal installer for conda, including conda, Python, and other useful packages. It also links to the 'Anaconda or Miniconda' page for installation reasons. Below this is a section titled 'Latest Miniconda installer links' with a table of download links for Windows and macOS.

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	29e008bc...f9250b1
macOS	Miniconda3 macOS Intel x86 64-bit bash	4b60eb49cf...62ed

06. miniconda 설치

(2023년 11월 기준)

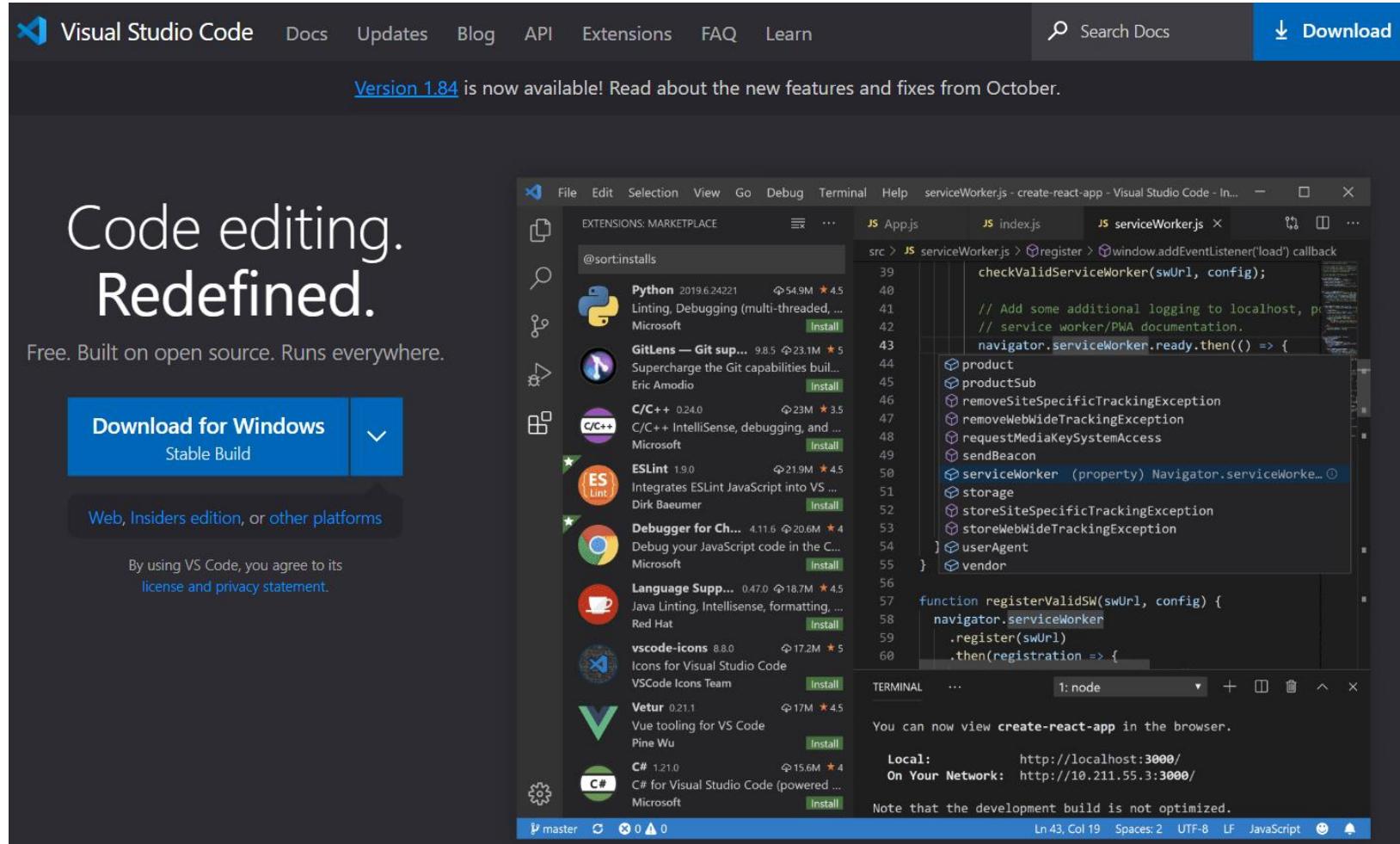
- ◆ miniconda는 아나콘다의 경량 버전으로, 데이터 과학 및 패키지 관리를 위한 필수 도구를 제공하는 작은 배포 버전
 - 사이트 : <https://docs.conda.io/projects/miniconda/en/latest/>

강의 실습 영상 참고

07. Visual Studio Code 설치

(2023년 11월 기준)

- ◆ Microsoft에서 개발한 인기 있는 무료 소스 코드 편집기로, Windows, Linux, MacOS에서 실행 가능
 - 사이트 : <https://code.visualstudio.com/>



07. Visual Studio Code 설치

(2023년 11월 기준)

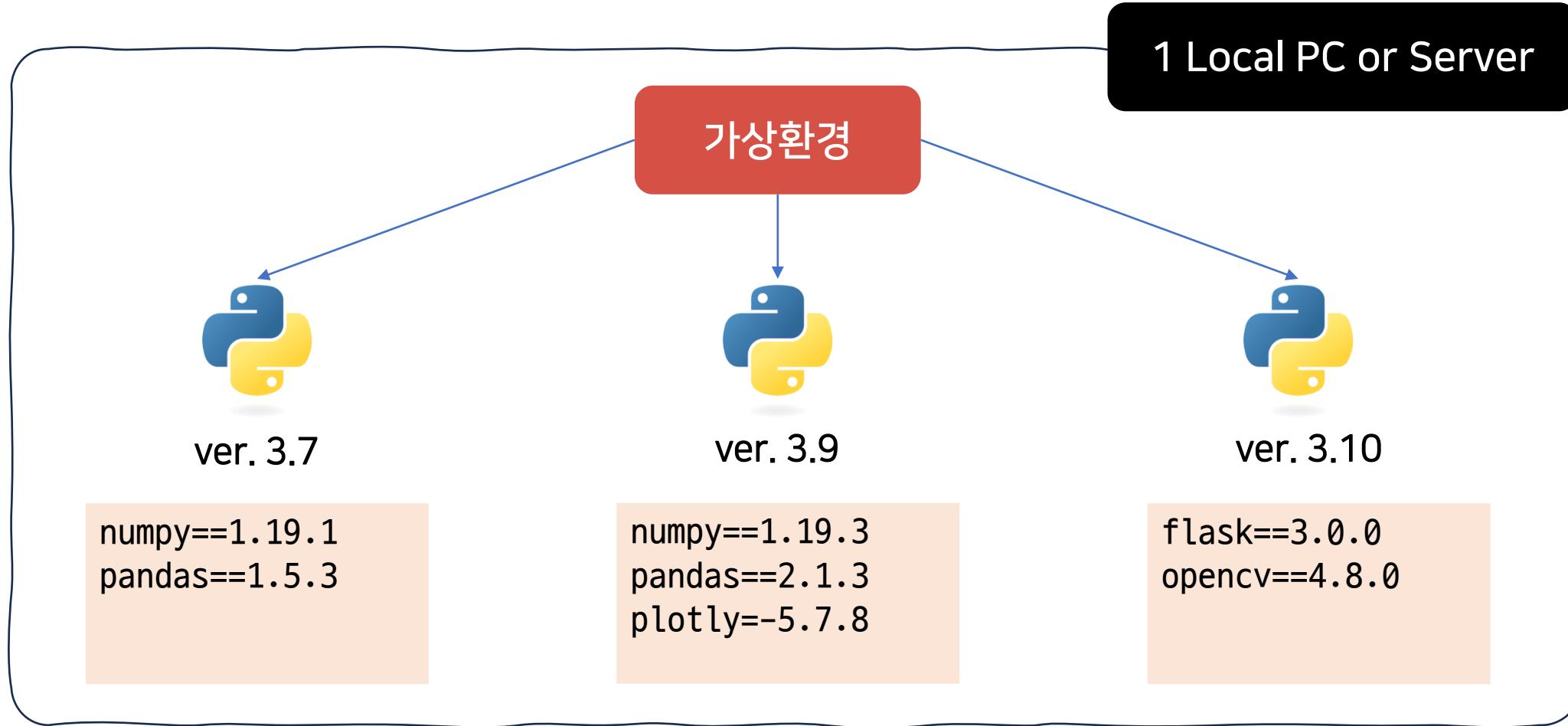
- ◆ Microsoft에서 개발한 인기 있는 무료 소스 코드 편집기로, Windows, Linux, MacOS에서 실행 가능
 - 사이트 : <https://code.visualstudio.com/>

강의 실습 영상 참고

08. Python 가상환경

(2023년 11월 기준)

- ◆ 가상환경은 프로젝트별로 독립적인 환경을 만들어주는 도구(virtualenv, conda, …)
- ◆ 프로젝트 간 라이브러리 버전 충돌을 피하고 라이브러리들을 유연하게 관리할 수 있도록 해줌



- ◆ 가상환경은 프로젝트별로 독립적인 환경을 만들어주는 도구
- ◆ 프로젝트 간 라이브러리 버전 충돌을 피하고 라이브러리들을 유연하게 관리할 수 있도록 해줌

강의 실습 영상 참고

09. github 연동

(2023년 11월 기준)

- ◆ 주요 명령어는 다음과 같다.

```
git config --global user.email "your_email@email.com"
```

```
git config --global user.name "yourusername"
```

- git 설정을 변경하는 것으로 전역으로 사용자의 이름과 이메일 주소를 설정 (일종의 로그인 기능)

```
git add .
```

```
git commit -m "your message"
```

```
git push
```

주요 명령어	설명
git add .	코드 수정 후, 변경 사항을 모두 준비하는 단계를 말하며, `.`은 현재 디렉토리를 의미
git commit	이전 단계에서 변경 사항으로 새로운 변경 사항으로 새 커밋
git push	커밋된 변경 사항을 원격 Git저장소(Github)에 업로드 함

강의 실습 영상 참고

10. Python 라이브러리 설치

(2023년 11월 기준)

◆ Python 라이브러리 설치방법 (pip)

```
pip install name_of_library
```

◆ Python 라이브러리 설치방법 (특정버전)

```
pip install name_of_library==버전번호
```

◆ Python 라이브러리 설치방법 (requirements.txt)

- requirements.txt 파일에 설치할 라이브러리 목록을 작성
-

```
pip install -r requirements.txt
```

강의 실습 영상 참고

파이썬 기초 문법

with  python™

01. 파이썬 자료형

(2023년 11월 기준)

◆ 숫자형(Number)

항목	표현 예시
정수 (int)	1, 2, 3, 123, 1345, 0, -123, -1, -2
실수 (float)	12.34, 456.189, 1.2e+5*, 1.2e-5**

Scientific Notation

$$* \ 1.2\text{e}+5 = 1.2 \times 10^5 = 120000$$

$$** \ 1.2\text{e}-5 = 1.2 \times 10^{-5} = 0.000012$$

◆ 문자열(String)

주요 기능	정의
더하기	더하기 순서대로 문자열을 하나로 연결한다는 것을 의미
곱하기	문자열을 반복해서 연결하는 것을 의미
인덱싱	특정 위치에 있는 문자만 지정하여 일부를 추출함. 인덱스번호는 0번째부터 시작함
슬라이싱	특정 위치에 있는 범위를 지정하여 전체 또는 1개 이상의 문자를 추출함

02. 문자열 주요 메서드

(2023년 11월 기준)

◆ 메서드, a는 임의의 문자열이 저장된 객체

주요 메서드	메서드 설명
a.count('p')	특정 문자 ('p')가 몇 개가 있는지 확인
a.find('p')	특정 문자 ('p')가 첫번째로 등장한 위치(인덱스) 번호를 확인
a.upper()	영어 문자를 대문자로 변환
a.lower()	영어 문자를 소문자로 변환
a.lstrip()	문자열 왼쪽에 있는 공백 제거
a.rstrip()	문자열 오른쪽에 있는 공백 제거
a.strip()	문자열 양쪽에 있는 공백 제거
a.replace("x", "y")	문자열에 포함된 x를 y로 변경
b = ','join(a)	기존 문자열(a)에 특정 문자(예: ,)를 문자열 사이마다 삽입할 수 있음
a.split(조건)	문자열을 특정 조건에 따라 나눠서 다수의 문자열로 구성된 리스트 자료형으로 변경

◆ 리스트는 숫자형, 문자열 자료형을 하나의 집합으로 구성할 수 있음

```
1 : []
2 : [2,4,6,8,10]
3 : ['빅데이터', '분석', '기사']
4 : [1, 2, 3, ['빅데이터', '분석', '기사']] # 중첩리스트(Nested List)
5 : ['빅데이터', '분석', '기사', 1, 2, 3]
```

◆ 문자열과 마찬가지로 더하기, 곱하기 같은 사칙연산을 활용해서 리스트를 합치거나 반복 가능

```
1 : a = [1, 2, 3]
2 : b = [4, 5, 6]
3 : c = a + b
4 : d = c * 3
5 : print(d)
```

03. 리스트

(2023년 11월 기준)

- ◆ 문자열과 마찬가지로 인덱싱과 슬라이싱 가능
- ◆ 중첩된 리스트에 대해서도 인덱싱과 슬라이싱 가능

```
1 : a = [1, 2, 3, ["a", "b", "c", "d"]]
2 : print(a)
3 : print(a[3][0]) # 중첩된 리스트 인덱싱 및 슬라이싱
```

```
[1, 2, 3, ['a', 'b', 'c', 'd']]
a
```

03. 리스트 - 주요 메서드

(2023년 11월 기준)

- ◆ a는 임의의 리스트가 저장된 객체

주요 메서드	메서드 설명
a.append(x)	리스트의 맨 마지막 요소 위치에 x라는 요소를 추가한다.
a.sort()	리스트의 요소들을 정렬하는 함수가 있다.
a.index(x)	리스트 내의 x값의 위치에 해당하는 인덱스 값을 반환한다.
a.insert(x, y)	x값은 인덱스 번호, y값은 특정 값으로 x 인덱스 위치에 y 값을 추가한다.
a.remove(x)	x는 특정값을 의미하며, 리스트에서 x 값을 제거한다.
a.pop(x)	x는 인덱스 번호를 의미, 리스트에서 해당 인덱스의 값을 반환하며, 리스트에서 제거한다.
a.count(x)	특정요소 x의 개수를 계산한다.

04. 튜플(Tuple) 자료형

(2023년 11월 기준)

- ◆ 튜플과 리스트는 비슷한 역할을 수행함. 대괄호가 아닌 소괄호를 사용해야 함.

```
1 : (1, 2, 3)
2 : (1, )
3 : 1, 2, 3
4 : (1, 2, 3, ('빅데이터', '분석', '기사')) # 중첩튜플(Nested Tuple)
5 : ('빅데이터', '분석', '기사', 1, 2, 3)
```

- ◆ 리스트와 마찬가지로 더하기, 곱하기 같은 사칙연산을 활용해서 튜플을 합치거나 반복 가능
- ◆ 인덱싱과 슬라이싱 가능
- ◆ 리스트와의 가장 큰 차이점
 - 리스트 : 요소의 생성, 삭제, 수정 가능
 - 튜플 : **요소 변경 불가능**

05. 딕셔너리(Dictionary 자료형)

(2023년 11월 기준)

◆ 키(Key)와 값(Value)으로 이루어진 자료형

```
1 : a = {'name' : 'evan', 'age' : 30, 'birth' : [4, 30]}
```

◆ Value 값을 구하기 위해서는 Key를 활용해야 함.

```
1 : a = {'name' : 'evan', 'age' : 30, 'birth' : [4, 30]}
2 : print(a['name'])
3 : print(a['birth'])
```

```
evan
[4, 30]
```

05. 딕셔너리(Dictionary 자료형) - 메서드

(2023년 11월 기준)

- ◆ 주요 메서드는 다음과 같음, a는 임의의 딕셔너리로 저장된 객체

주요 메서드	메서드 설명
a.keys()	딕셔너리의 키(Key)의 리스트를 추출할 수 있음
a.values()	딕셔너리의 값(Value)의 리스트를 추출할 수 있음
a.items()	딕셔너리의 Key, Value를 튜플 구조로 묶고, 리스트로 추출할 수 있음
a.get(x, y)	딕셔너리의 x의 키가 존재하지 않을 때 y값 반환하도록 처리

- ◆ If문 : 조건문1을 테스트하여 참이면 if문, 아니면 elif 조건문2를 테스트하여 참이면 코드

```
1  :  if 조건문1:  
2  :      코드 실행 1 # 들여쓰기는 공백(Spacebar) 또는 탭(Tab) 사용  
3  :      코드 실행 2  
4  :  elif 조건문2:  
5  :      코드 실행 1  
6  :      코드 실행 2  
6  :  else:  
7  :      코드 실행 1  
8  :      코드 실행 2  
9  :  A = [1, 2, 3]  
10 :  ...
```

06. 파이썬 제어문

(2023년 11월 기준)

◆ 조건문 작성 방법

조건문 표현 방법	조건문 의미
$x < y$	x가 y보다 작다면
$x > y$	x가 y보다 크다면
$x == y$	x와 y가 같다면
$x != y$	x와 y가 같지 않다면
$x >= y$	x가 y보다 크거나 같다면
$x <= y$	X가 y보다 작거나 같다면

06. 파이썬 제어문

(2023년 11월 기준)

◆ 조건문 비교 연산자 (AND)

X	Y	Result
True	True	True
True	False	False
False	True	False
False	False	False

◆ 조건문 비교 연산자 (OR)

X	Y	Result
True	True	True
True	False	True
False	True	True
False	False	False

◆ not x : x가 거짓이면 참이다

- ◆ while 반복문 : 조건문이 참인 동안에 while 아래의 문장을 반복해서 수행함

```
1  :  a = 0
2  :
3  :  while a < 3:
4  :      print(f'현재 a값은 {a} 입니다.')
5  :      a = a + 1
6  :
7  :  print("종료")
```

현재 a값은 0 입니다.
현재 a값은 1 입니다.
현재 a값은 2 입니다.
종료

06. 파이썬 제어문

(2023년 11월 기준)

- ◆ for 반복문 : 리스트, 문자열, 튜플 등 0번째 인덱스부터 마지막 인덱스까지 차례로 변수에 대입되어 명령문이 실행

```
1  : numbers = [100, 200, 300]
2  :
3  : for num in numbers:
4  :     print(num)
5  :
6  : print("종료")
```

```
100
200
300
종료
```

07. 사용자 정의 함수

(2023년 11월 기준)

- ◆ def문 : 함수 이름을 임의로 만들고, 이름 뒤 괄호 안의 매개변수는 입력으로 전달되는 값을 받는 변수

```
1 : def 함수이름(매개변수1, 매개변수2, ...)  
2 :     코드 1  
3 :     코드 2  
4 :     ...  
5 :     코드 N  
6 :     return 결괏값
```

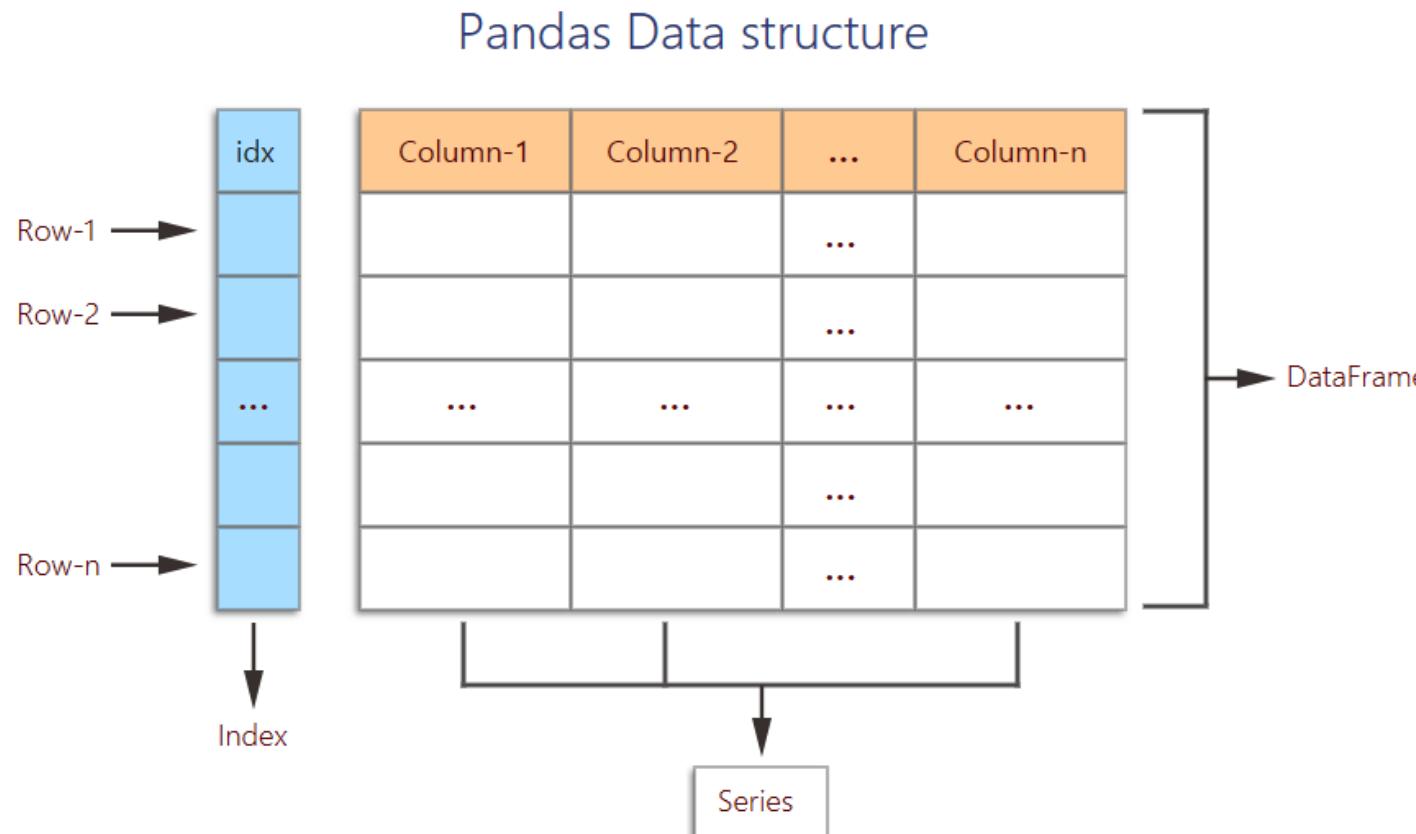
강의 실습 영상 참고

파이썬 데이터 분석 라이브러리

with  python™

◆ Series vs DataFrame

- ✓ Series : 1차원 배열의 형태를 가진 자료 구조, index와 value 확인 가능
- ✓ DataFrame : 2차원 행렬 구조의 테이블 형태로 구성, index와 복수의 컬럼(Column)이 존재함



출처 : https://www.w3resource.com/w3r_images/pandas-data-structure.svg

◆ DataFrame 컬럼의 데이터 확인 및 변경

구분	내용
int	정수형, 소수점을 가지지 않은 숫자
float	실수형, 소수점 이하의 값을 가진 숫자
bool	부울형, True 혹은 False로 이루어짐
datetime	날짜와 시간 표현
category	카테고리, 범주형 변수일 경우 사용
object	문자열 & 복합형, 위의 형식으로 정할 수 없거나 여러 형식이 섞여 있는 경우 사용

◆ Series vs DataFrame

- ✓ Series : 1차원 배열의 형태를 가진 자료 구조, index와 value 확인 가능
- ✓ DataFrame : 2차원 행렬 구조의 테이블 형태로 구성, index와 복수의 컬럼(Column)이 존재함

◆ 주요 메서드 – 데이터 살펴보기, data는 pandas DataFrame을 의미한다.

주요 메서드	설명
data.head(n)	데이터프레임의 첫번째 행부터 순차적으로 n개까지의 행을 반환
data.tail(n)	데이터프레임의 마지막 행부터 역순으로 n개까지의 행을 반환
data.shape	데이터프레임의 행과 컬럼 정보를 튜플 형태로 반환
data.info()	데이터프레임의 컬럼, Non-Null 데이터 개수, 컬럼의 타입
data.describe()	컬럼별 숫자형 데이터의 개수, 평균값, 표준편차, 최솟값, 사분위수값, 최댓값을 표현함
value_counts()	해당 컬럼 값의 유형과 건수를 확인할 수 있음. 데이터의 분포도를 확인하는 데 유용함

◆ Series vs DataFrame

- ✓ Series : 1차원 배열의 형태를 가진 자료 구조, index와 value 확인 가능
- ✓ DataFrame : 2차원 행렬 구조의 테이블 형태로 구성, index와 복수의 컬럼(Column)이 존재함

◆ 주요 메서드 – 데이터 살펴보기, data는 pandas DataFrame을 의미한다.

◆ 주요 작업

- ✓ 컬럼 생성과 수정
- ✓ 데이터프레임 데이터 삭제

```
1 : data.drop("column1", axis=1) # column1 삭제  
2 : data.drop(["column1", "column2"], axis=1) # column1, column2 삭제  
3 : data.drop([0, 1, 2], axis=0) # 행 인덱스 0, 1, 2 삭제
```

◆ Series vs DataFrame

- ✓ Series : 1차원 배열의 형태를 가진 자료 구조, index와 value 확인 가능
- ✓ DataFrame : 2차원 행렬 구조의 테이블 형태로 구성, index와 복수의 컬럼(Column)이 존재함

◆ 주요 메서드 – 데이터 살펴보기, data는 pandas DataFrame을 의미한다.

◆ 주요 작업

- ✓ 컬럼 생성과 수정
- ✓ 데이터프레임 데이터 삭제
- ✓ 데이터 조회 (컬럼명, 슬라이싱, 논리형 인덱싱)

◆ Input/Output

- ✓ 다양한 파일 읽기 및 쓰기 제공(CSV, JSON, HTML 등)
- ✓ SAS, SPSS, SQL, Google BigQuery, STATA 등도 제공

파일구분	Reading Data	Writing Data
CSV	<code>pd.read_csv()</code>	<code>DataFrame.to_csv()</code>
EXCEL	<code>pd.read_excel()</code>	<code>DataFrame.to_excel()</code>
JSON	<code>pd.read_json()</code>	<code>DataFrame.to_json()</code>
HTML	<code>pd.read_html()</code>	<code>DataFrame.to_html()</code>
SQL	<code>pd.read_sql()</code>	<code>DataFrame.to_sql()</code>
Parquet	<code>pd.read_parquet()</code>	<code>DataFrame.to_parquet()</code>
HDF	<code>pd.read_hdf()</code>	<code>DataFrame.to_hdf()</code>

◆ iloc vs loc 차이

iloc	loc
Integer-location based (위치 기반)	Label(s) or Boolean Array based (명칭 기반)
<code>data.iloc[row_index, column_index]</code>	<code>data.loc[row_label, column_label]</code>
input 형태 <ul style="list-style-type: none">- integer- List or Array [4, 3, 0]- Slicing 1:7- Boolean Array	input 형태 <ul style="list-style-type: none">- Single Label 5, 'a'- List or Array of Label, ["a", "b", "c"]- Slicing 'a' : 'c'

◆ 데이터 정렬 및 집계함수

주요 메서드	설명
data.sort_values()	DataFrame, Series의 정렬을 할 때 사용한다. SQL의 order by 키워드와 유사함.
data.sum()	DataFrame의 모든 컬럼 각각 합계가 나타남.

```
1 : Import seaborn as sns  
2 : iris = sns.load_dataset('iris')  
3 : iris[['sepal_length', 'sepal_width', 'petal_length']].sum()
```

```
sepal_length    876.5  
sepal_width     458.6  
petal_length    563.7  
dtype: float64
```

◆ 데이터 정렬 및 집계함수

주요 메서드	설명
data.sort_values()	DataFrame, Series의 정렬을 할 때 사용한다. SQL의 order by 키워드와 유사함.
data.sum()	DataFrame의 모든 컬럼에 대하여 각각 합계가 나타남.
data.min()	DataFrame의 모든 컬럼에 대하여 최솟값이 나타남.
data.max()	DataFrame의 모든 컬럼에 대하여 최댓값이 나타남.
data.count()	DataFrame의 모든 컬럼에 대하여 행의 개수가 나타남
data.mean()	DataFrame의 모든 컬럼에 대하여 평균값이 나타남
data.median()	DataFrame의 모든 컬럼에 대하여 중간값이 나타남

03. pandas

(2023년 11월 기준)

◆ 데이터 요약 및 기술통계량

주요 메서드	설명
describe()	데이터의 기초 통계량(평균, 표준편차 각 컬럼의 사분위수 등) 함수
describe(include=[object])	Object 컬럼에 count, unique, top, freq 값을 출력함

```
1 : import seaborn as sns  
2 : iris = sns.load_dataset("iris")  
3 : iris.describe(include=[object])
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
1 : import seaborn as sns  
2 : iris = sns.load_dataset("iris")  
3 : iris.describe(include=[object])
```

	species
count	150
unique	3
top	setosa
freq	50

03. pandas

(2023년 11월 기준)

◆ rename

주요 메서드	설명
<code>df.rename(columns={'old name': 'new name'})</code>	컬럼명을 변경할 때 사용하는 메서드

```
1 : sample_df = sample_df.rename(columns={'ZN': 'landZone'})  
2 : sample_df.head()
```

	CRIM	ZN	INDUS	CHAS	
0	0.00632	18.0	2.31	0.0	
1	0.02731	0.0	7.07	0.0	
2	0.02729	0.0	7.07	0.0	
3	0.03237	0.0	2.18	0.0	
4	0.06905	0.0	2.18	0.0	

→

	CRIM	landZone	INDUS	CHAS	
0	0.00632	18.0	2.31	0.0	
1	0.02731	0.0	7.07	0.0	
2	0.02729	0.0	7.07	0.0	
3	0.03237	0.0	2.18	0.0	
4	0.06905	0.0	2.18	0.0	

◆ value_counts() : 고유 값의 개수를 반환함

df.value_counts(normalize=False)

- normalize True로 설정 시, 각 객체는 고유값의 상대적인 비율로 조회됨

```
1 : df_boston['RAD'].value_counts()
```

```
24.0    132  
5.0     115  
4.0     110  
3.0      38  
6.0      26  
2.0      24  
8.0      24  
1.0      20  
7.0      17  
Name: RAD, dtype: int64
```

```
1 : df_boston['RAD'].value_counts(normalize=True)
```

```
24.0    0.260870  
5.0     0.227273  
4.0     0.217391  
3.0     0.075099  
6.0     0.051383  
2.0     0.047431  
8.0     0.047431  
1.0     0.039526  
7.0     0.033597  
Name: RAD, dtype: float64
```

◆ isin() : 데이터 필터링, 데이터 프레임의 각 요소가 값에 포함되어 있는지 판단

df.isin(values)

- values iterables(i.e., List), Series(index), DataFrame(index & column labels) or dict(keys)

```
1 : numbers = [1.0, 7.0]
2 : filtered_df = df_boston[df_boston['RAD'].isin(numbers)]
3 : filtered_df[['CRIM', 'RAD']].head(6)
```

CRIM RAD

0	0.00632	1.0
193	0.02187	1.0
194	0.01439	1.0
244	0.20608	7.0
245	0.19133	7.0
246	0.33983	7.0

◆ 날짜 데이터 처리

- ✓ 날짜와 시간을 다루기 위해서는 datetime 라이브러리 활용
- ✓ 날짜 형식으로 변환방법 (pandas DataFrame)

```
df['date'] = pd.to_datetime(df['date'], format="%Y-%m-%d %H:%M:%S")
```

◆ datetime 객체 사용 예시

```
df['date'].dt.year
```

구분	내용
year	객체 datetime의 연도를 추출함
month	객체 datetime의 월을 추출함
day	객체 datetime의 일을 추출함
hour	객체 datetime의 시간을 추출함
weekday	객체 datetime 날짜의 요일을 추출함 (Monday = 0, Sunday=6)

그 외 : <https://pandas.pydata.org/docs/reference/series.html#datetimelike-properties>

◆ Timedelta

- ✓ 두 날짜 또는 시간 간의 차이, 즉 기간을 표현함
- ✓ 두개의 datetime 컬럼(i.e., 출발시간 - 도착시간) 연산 시, **Timedelta 객체**로 자동 변환

```
df['이동시간'] = df['도착시간'] - df['출발시간']
```

```
import pandas as pd

df = pd.DataFrame({
    '출발시간' : [pd.to_datetime('2023-01-01 08:00:00')],
    '도착시간' : [pd.to_datetime('2023-01-01 09:30:00')],
})

df['이동시간'] = df['도착시간'] - df['출발시간']
print(df.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1 entries, 0 to 0
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype    
--- 
 0   출발시간      1 non-null    datetime64[ns]
 1   도착시간      1 non-null    datetime64[ns]
 2   이동시간      1 non-null    timedelta64[ns]
dtypes: datetime64[ns](2), timedelta64[ns](1)
memory usage: 152.0 bytes
None
```

```
df.head(1)
```

	출발시간	도착시간	이동시간
0	2023-01-01 08:00:00	2023-01-01 09:30:00	0 days 01:30:00

◆ Timedelta

- ✓ 두 날짜 또는 시간 간의 차이, 즉 기간을 표현함
- ✓ 두개의 datetime 컬럼(i.e., 출발시간 - 도착시간) 연산 시, **Timedelta 객체**로 자동 변환

```
df.head(1)
```

	출발시간	도착시간	이동시간
0	2023-01-01 08:00:00	2023-01-01 09:30:00	0 days 01:30:00

```
df['이동시간'].dt.days
```

구분	내용
days	0
total_seconds()	$5400(\text{초}) = \text{hours} * 3600 + \text{minutes} * 60 + \text{seconds}$
dt.total_seconds() / 60	$90(\text{분}) = 5400 / 60$
dt.total_seconds() / (60 * 60)	$1.5(\text{시간}) = 5400 / (60 * 60)$

그 외 : <https://pandas.pydata.org/docs/reference/api/pandas.Timedelta.html>

◆ shift()

- ✓ 인덱스는 그대로 두고 데이터만 이동이 가능함
- ✓ 현재 기준 앞으로 또는 지정된 기간만큼 이동함

`df.shift(periods=1, fill_value=object, optional)`

- `periods` 데이터가 shift 이동할 기간의 숫자, 양수 또는 음수가 올 수 있음
 - `fill_value` shift 실행 시, 발생할 결측값을 채워주는 scalar 값
-

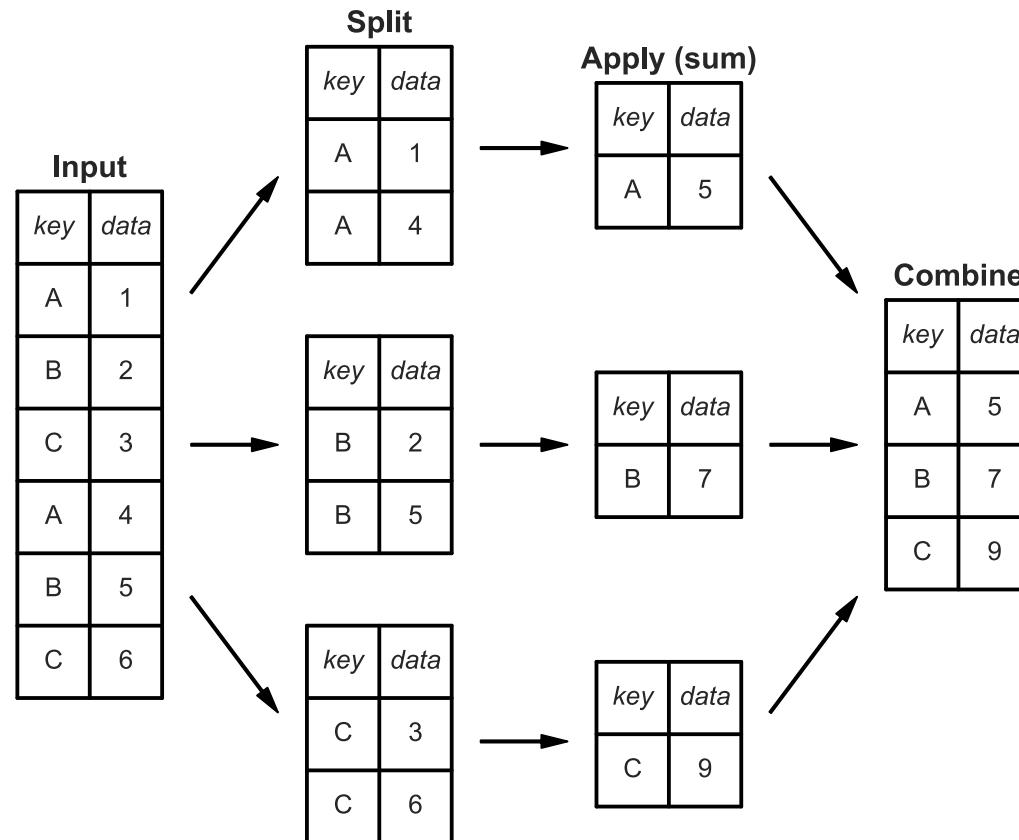
```
temp_df['shifted_v1'] = temp_df['price'].shift(periods=1, fill_value=0).astype(int)
temp_df['shifted_v2'] = temp_df['price'].shift(periods=2, fill_value=0).astype(int)
```

	datesold	price	shifted_v1	shifted_v2
0	2007-02-07	525000	0	0
1	2007-02-27	290000	525000	0
2	2007-03-07	328000	290000	525000
3	2007-03-09	380000	328000	290000
4	2007-03-21	310000	380000	328000

03. pandas

(2023년 11월 기준)

◆ groupby 원리



◆ 집계함수 종류

주요 메서드	설명
<code>count()</code>	값의 개수
<code>sum()</code>	값들의 합
<code>min()</code>	최솟값
<code>max()</code>	최댓값
<code>mean()</code>	평균
<code>median()</code>	중앙값
<code>std()</code>	표준편차
<code>var()</code>	분산
<code>quantile()</code>	분위수
<code>first()</code>	첫번째 값
<code>last()</code>	마지막 값

<https://jakevdp.github.io/blog/2017/03/22/group-by-from-scratch/>

03. pandas

(2023년 11월 기준)

◆ groupby 원리

```
1 : tips.groupby(  
2 :     ['sex', 'smoker'])  
3 :         ).agg(  
4 :             mean_bill = ('total_bill', 'mean'),  
5 :             median_tip = ('tip', 'median'))  
6 :         ).reset_index()
```

sex smoker → mean_bill median_tip

0	Male	Yes	22.284500	3.00
1	Male	No	19.791237	2.74
2	Female	Yes	17.977879	2.88
3	Female	No	18.105185	2.68

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female		No	Sun	Dinner
1	10.34	1.66	Male		No	Sun	Dinner
2	21.01	3.50	Male		No	Sun	Dinner
3	23.68	3.31	Male		No	Sun	Dinner
4	24.59	3.61	Female		No	Sun	Dinner

03. pandas

(2023년 11월 기준)

◆ 데이터프레임 결측치 처리

- ✓ 컬럼에 값이 없는 Null을 의미한다.
- ✓ 결측치를 처리하지 않으면 작동하지 않으므로 이 값을 반드시 다른 값으로 대체해야 함

주요 메서드	설명
data.isna()	결측치 여부를 확인하는 함수, 반환값은 True/False
data.fillna()	결측치를 채우는 함수
data.dropna()	결측값이 포함된 모든 행을 삭제

```
df['몸무게'] = df['몸무게'].fillna(df['몸무게'].mean())
```

	연도	키	몸무게	시력	병결
0	2017	160.0	53.0	1.2	NaN
1	2018	162.0	52.0	NaN	NaN
2	2019	165.0	NaN	1.2	NaN
3	2020	Nan	50.0	1.2	2.0
4	2021	NaN	51.0	1.1	NaN
5	2022	166.0	54.0	0.8	1.0

	연도	키	몸무게	시력	병결
0	2017	160.0	53.0	1.2	NaN
1	2018	162.0	52.0	NaN	NaN
2	2019	165.0	52.0	1.2	NaN
3	2020	Nan	50.0	1.2	2.0
4	2021	NaN	51.0	1.1	NaN
5	2022	166.0	54.0	0.8	1.0

03. pandas

(2023년 11월 기준)

◆ 데이터 재구조화

- ✓ pivot_table : 많은 양의 데이터에서 필요한 자료만을 뽑아 데이터를 재구조화 할 수 있음

```
df.pivot_table(index=["ID"], columns=["반"], values = "성적", aggfunc="sum")
```

- index 피벗테이블에서 인덱스로 지정할 컬럼의 이름(두 개 이상이면 리스트로 입력할 것)
- columns 피벗테이블에서 컬럼으로 지정할 컬럼의 이름(범주형 변수 활용)
- values 피벗테이블에서 columns의 값이 될 컬럼의 이름(수치형 변수 활용)
- aggfunc 집계함수를 사용할 경우 지정

ID	반	성적
0	1	A 100
1	1	B 88
2	1	A 85
3	2	B 75
4	2	A 100
5	2	B 80



	반	A	B
ID			
1		185	88
2		100	155

◆ 데이터 재구조화

- ✓ melt : 피벗테이블의 반대 개념으로 생각하면 된다

```
df.melt(id_vars = ["ID"], var_name = "반", value_name = "성적")
```

- id_vars 피벗 테이블에서 인덱스가 될 컬럼의 이름
- var_name variable 변수의 이름으로 지정할 문자열(선택)
- value_name value 변수의 이름으로 지정할 문자열(선택)

	반	A	B
ID			
1	185	88	
2	100	155	



	ID	반	성적
0	1	A	92.5
1	2	A	100.0
2	1	B	88.0
3	2	B	77.5

◆ 주의

- ✓ 기존 테이블에서 집계된 값을 원래값으로 재분리하는 것은 아님

Chapter 03. pandas

강의 실습 영상 참고

파이썬 시작화

with  python™

01. 데이터 시각화의 중요성

(2023년 11월 기준)



Florence Nightingale (1820 ~ 1910)

01. 데이터 시각화의 중요성

(2023년 11월 기준)



Florence Nightingale (1820 ~ 1910)

간호사
(크림전쟁)

저술활동
(간호를 위하여)

통계학자

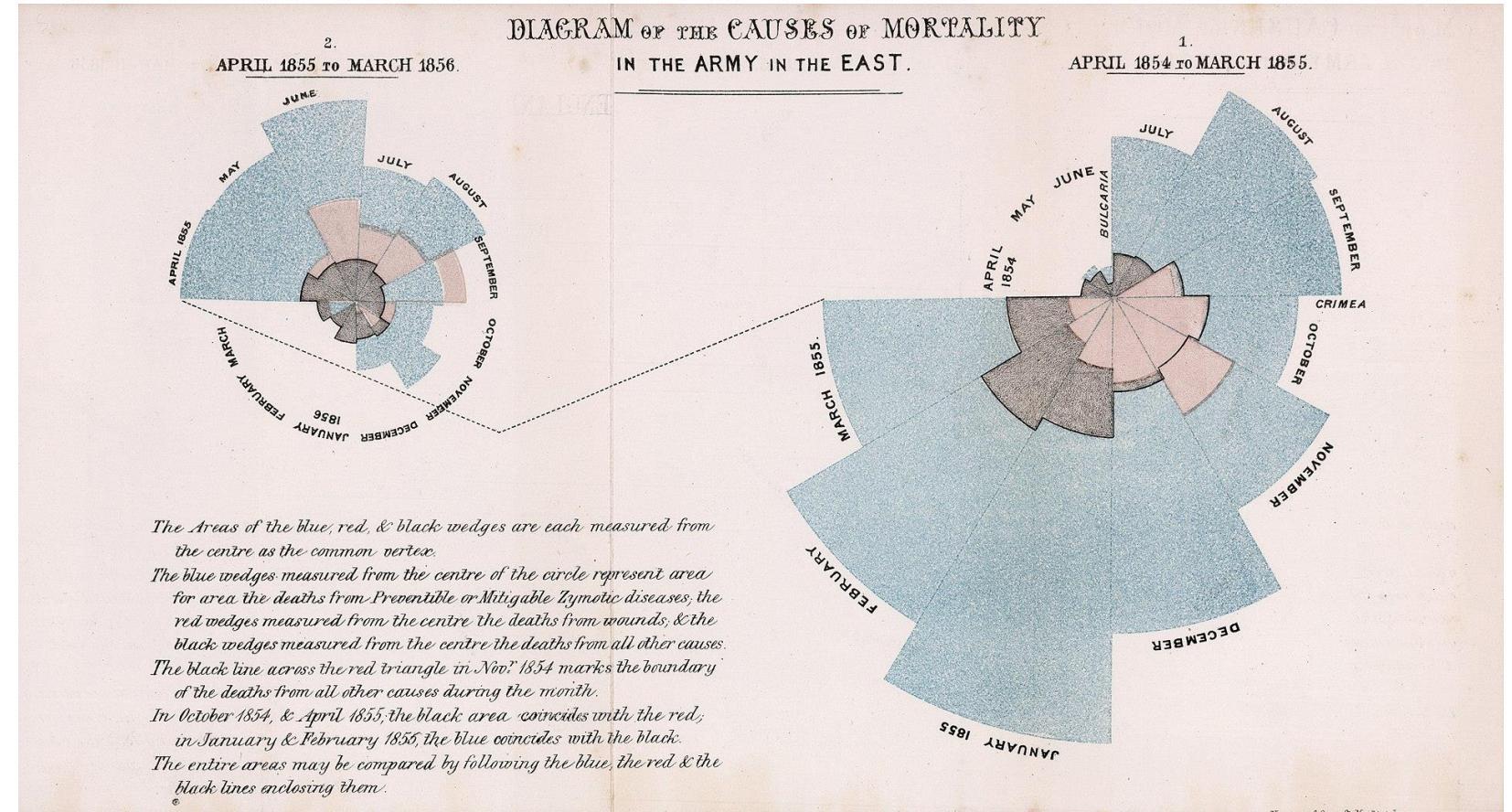
01. 데이터 시각화의 중요성

(2023년 11월 기준)



Florence Nightingale (1820 ~ 1910)

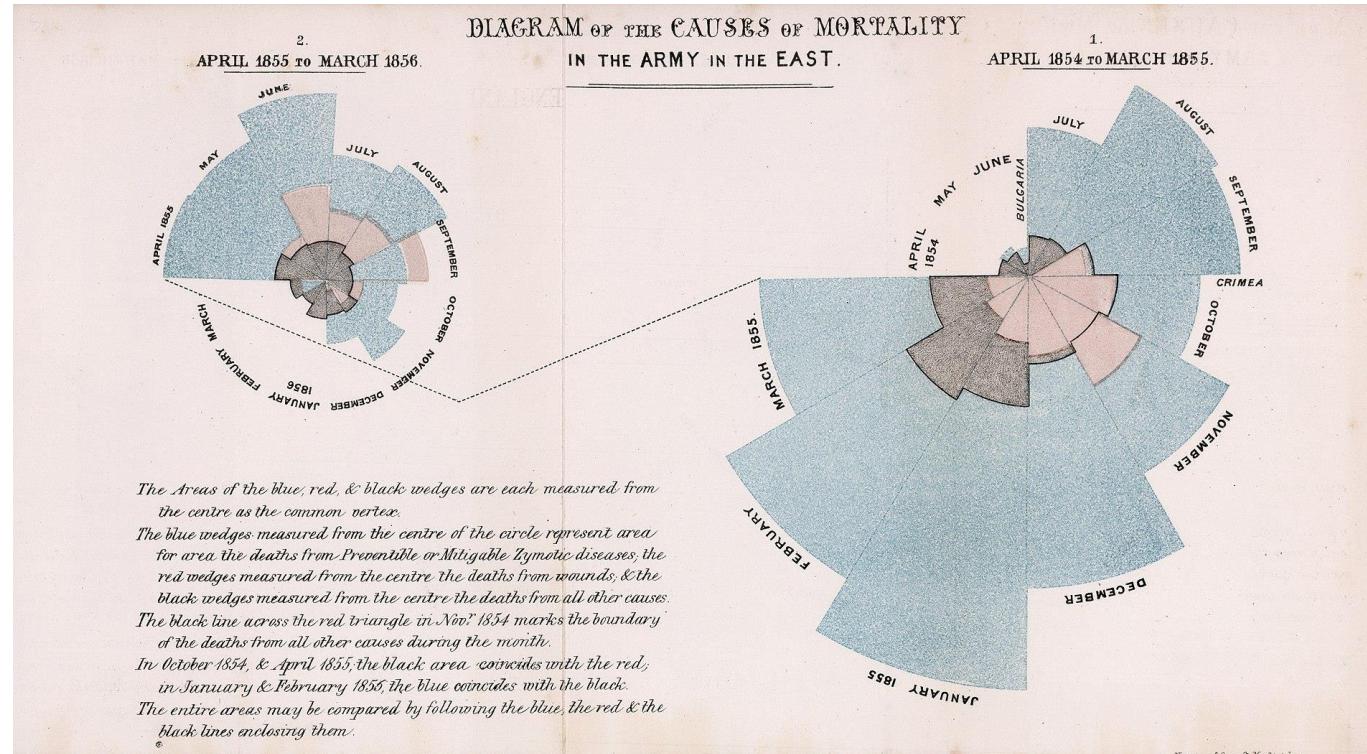
“로즈 다이어그램 Rose Diagram”



출처: <https://commons.wikimedia.org/wiki/File:Nightingale-mortality.jpg>

01. 데이터 시각화의 중요성

(2023년 11월 기준)



“ 전쟁보다 전염병에 의한 사망이 더 많으니 야전병영 위생을 개선해야 합니다! ”

02. 수치 데이터 분석과 시각화 분석의 조합

(2023년 11월 기준)

앤스컴 콰르텟(Anscombe's Quartet)

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.5
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

출처: https://en.wikipedia.org/wiki/Anscombe%27s_quartet

”

02. 수치 데이터 분석과 시각화 분석의 조합

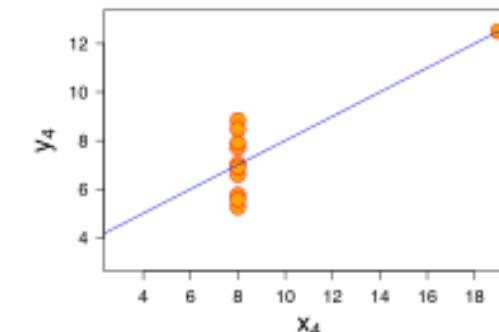
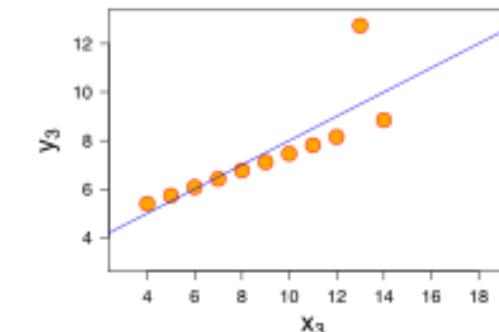
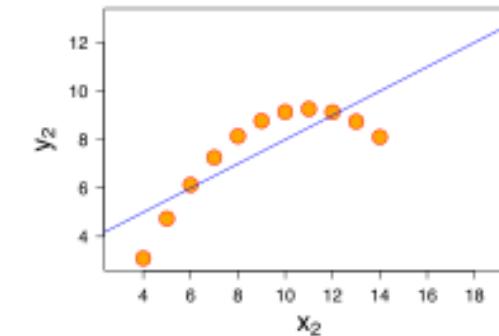
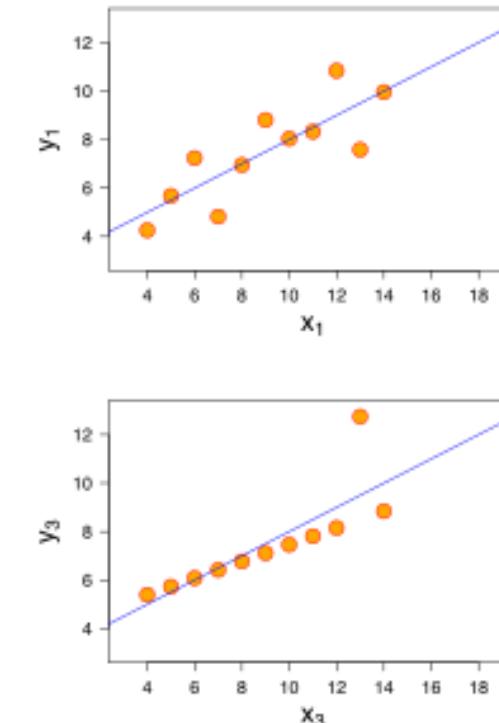
(2023년 11월 기준)

앤스컴 콰르텟(Anscombe's Quartet)

동일한 기술 통계량

항목	값
x 평균	9
x 표본분산	11
y 평균	7.50
y 표본분산	4.125
x 와 y 의 상관	0.816
선형회귀선	$y = 3.00 + 0.500x$
선형회귀 결정계수	0.67

4개의 다른 시각화



”

03. 데이터 시각화의 원리

(2023년 11월 기준)

삭제

(Delete)

분리

(Divide)

강조

(Highlight)

배열

(Arrange)

03. 데이터 시각화의 원리

(2023년 11월 기준)

- ◆ 시각화의 기본 원리 : 삭제 > 분리 > 강조 > 배열

삭제
(Delete)

분리
(Divide)

강조
(Highlight)

배열
(Arrange)

시각화 원리	설명
삭제 (cut, Delete)	필수적인 데이터 남기고, 의미 없는 Chart는 삭제
분리 (Divide)	데이터 분리한 뒤, 별도 Chart로 만든 후, 논리 순서로 배열
강조 (Highlight)	필수 데이터는 강조, 부가적 데이터는 약하게 또는 숨김 처리 (대조)
배열 (Arrange)	데이터를 영역 별로 그룹핑 한 후, 영역 간 및 영역 내 데이터 간 논리 순서로 구조화

Gene Zelazny, <Say it with Charts>

04. 데이터 차트의 종류

(2023년 11월 기준)

◆ Chart 종류 선택

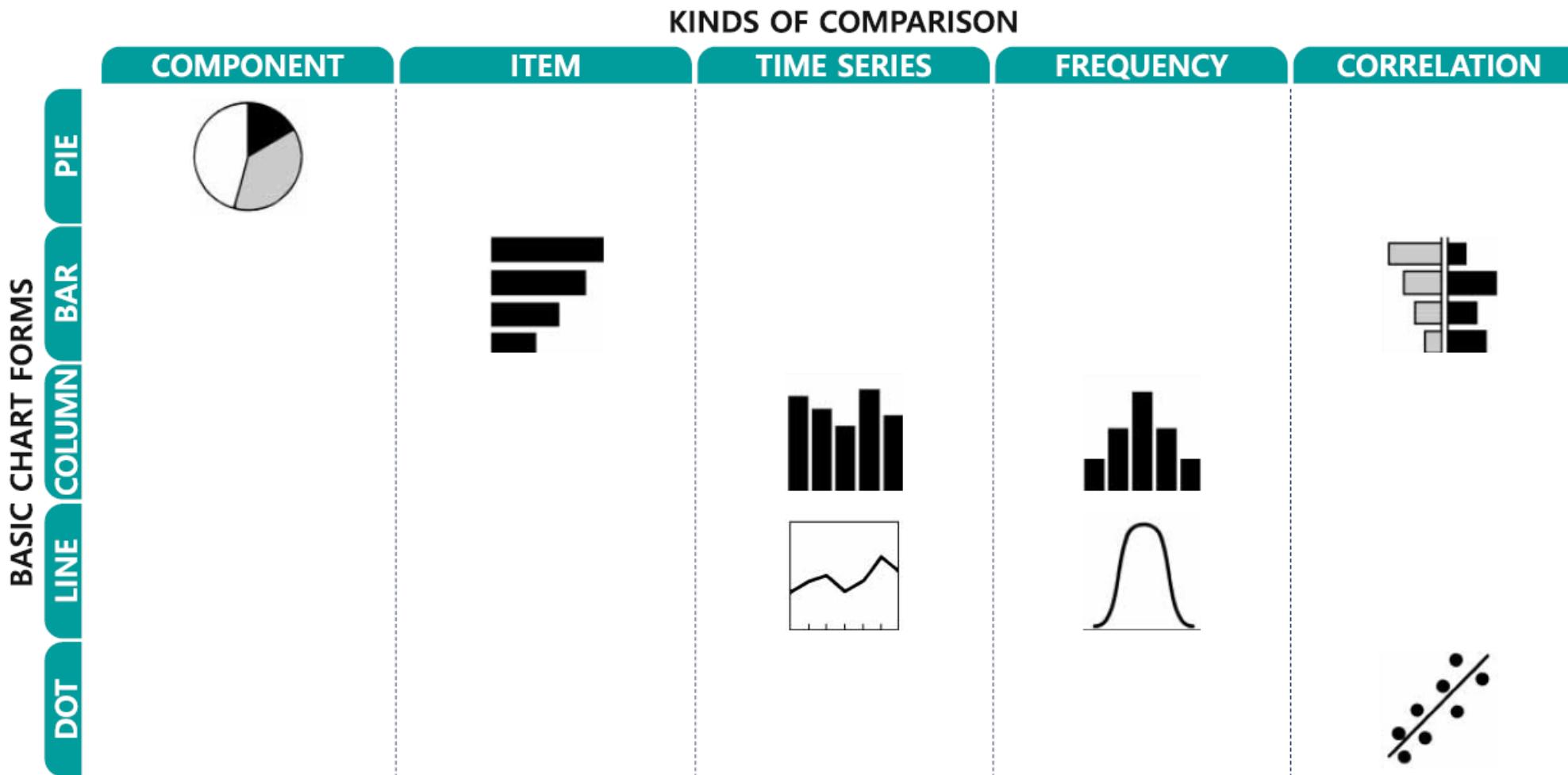
메시지	데이터의 유형	차트 형식
'비율', '퍼센트', '비중'	구성 요소(Component) 비교 : 백분율	Pie, Stacked Column
'~보다 많음', '~보다 적음'	항복(Item) 비교 : 항목의 순위	Bar, Waterfall
'변화', '성장', '변동'	시간적 추이(Time Series) 비교	Column or Line 차트
'분포는~'	빈도분포(Frequency) 비교	Column or Line 차트
'~에 관련된다', '~에 따라 변화'	상관관계 (Correlation) 비교	Scatter or Paired Bar Chart

Gene Zelazny, <Say it with Charts>

04. 데이터 차트의 종류

(2023년 11월 기준)

◆ Chart 종류 선택

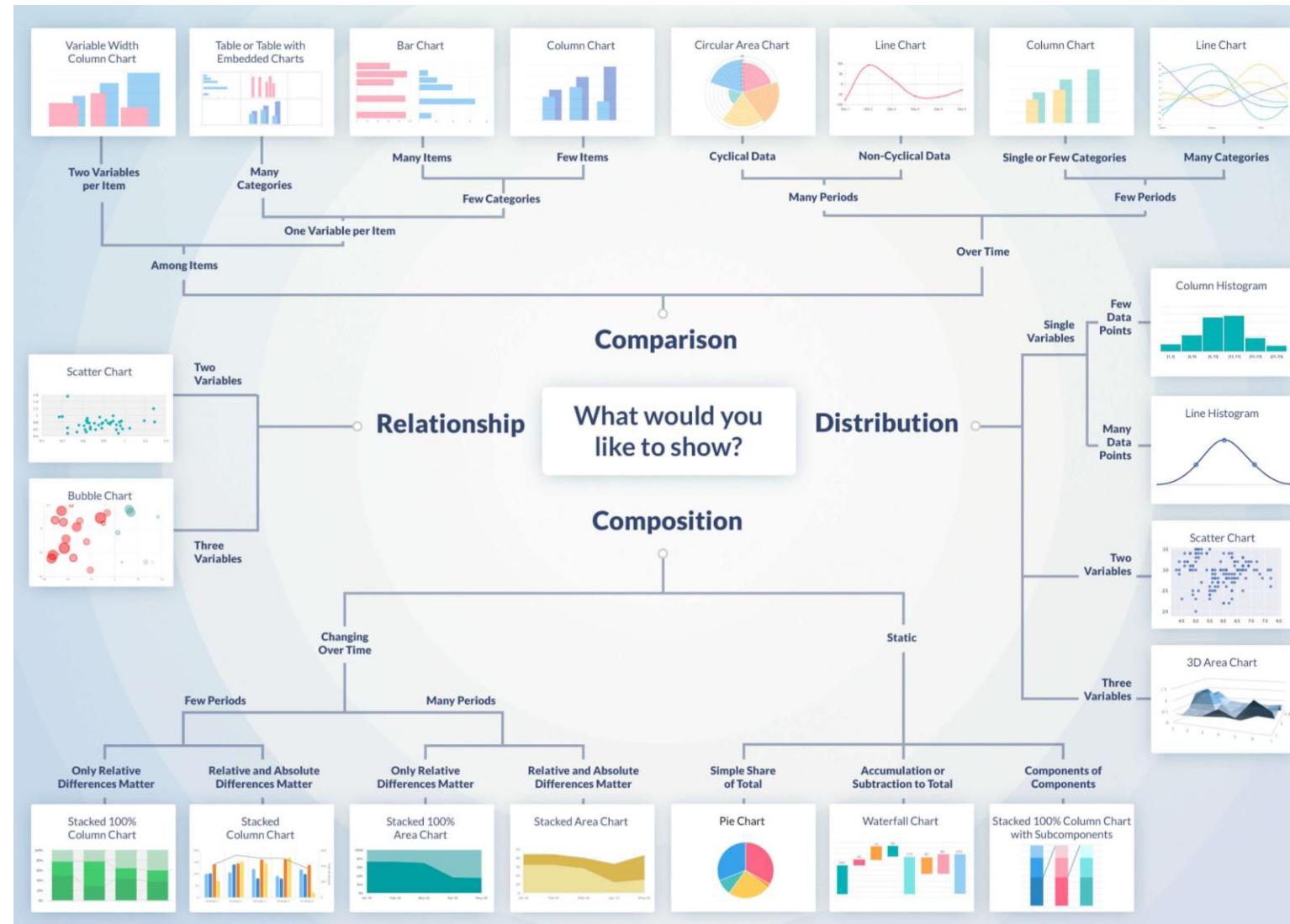


Gene Zelazny, <Say it with Charts>

04. 데이터 차트의 종류

(2023년 11월 기준)

◆ Chart 종류 선택



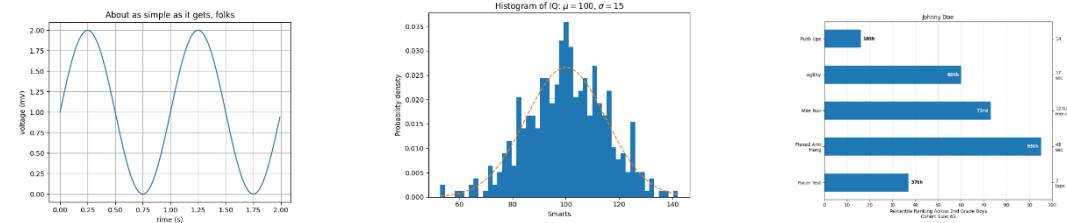
출처 : <https://www.tapclicks.com/wp-content/uploads/How-to-Visualize-your-Data-with-Charts-and-Graphs.jpg>

05. matplotlib + seaborn

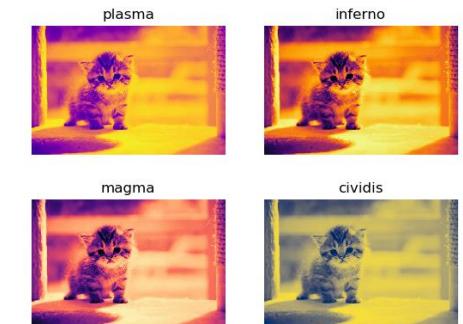
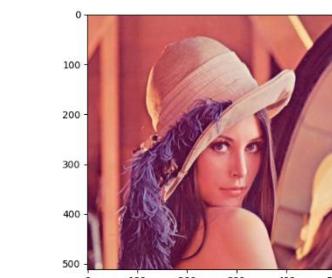
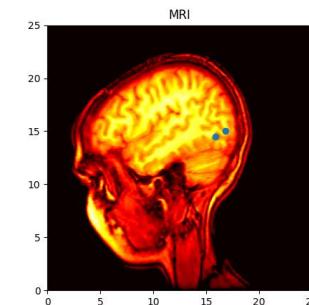
(2023년 11월 기준)



정형 데이터 시각화



이미지 데이터 시각화

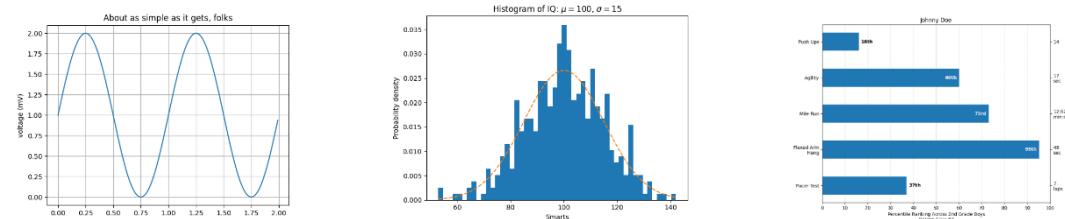


05. matplotlib + seaborn

(2023년 11월 기준)

matplotlib

정형 데이터 시각화



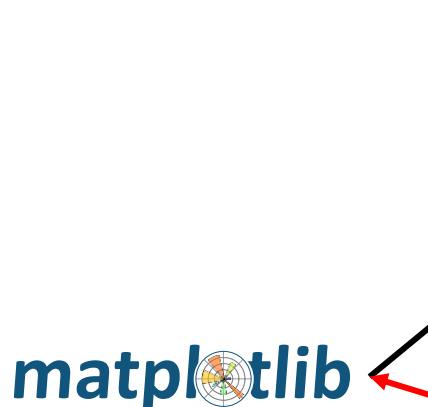
- 보완 및 장점

- ① 비 전공자들에게 matplotlib 시각화 문법은 조금 어렵다.
- ② pandas 데이터 프레임에서 쉽게 시각화 구현 가능하도록 한다.
- ③ 통계 (회귀선) 그래프 등을 쉽게 구현할 수 있도록 한다.

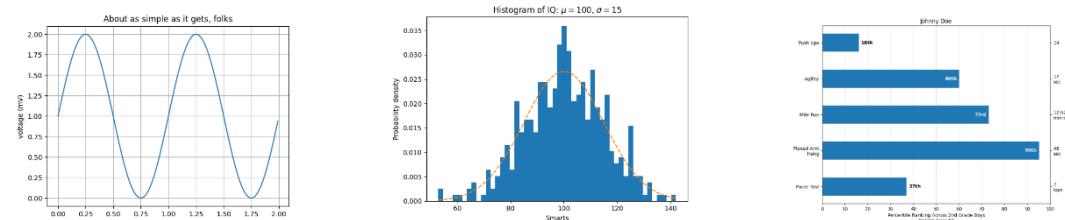


05. matplotlib + seaborn

(2023년 11월 기준)



정형 데이터 시각화



- 보완 및 장점

- ① 비 전공자들에게 matplotlib 시각화 문법은 조금 어렵다.
- ② pandas 데이터 프레임에서 쉽게 시각화 구현 가능하도록 한다.
- ③ 통계 (회귀선) 그래프 등을 쉽게 구현할 수 있도록 한다.



- 단점

- ① 세부 옵션을 수정 하려면 Matplotlib을 알아야 한다.

05. matplotlib + seaborn

(2023년 11월 기준)

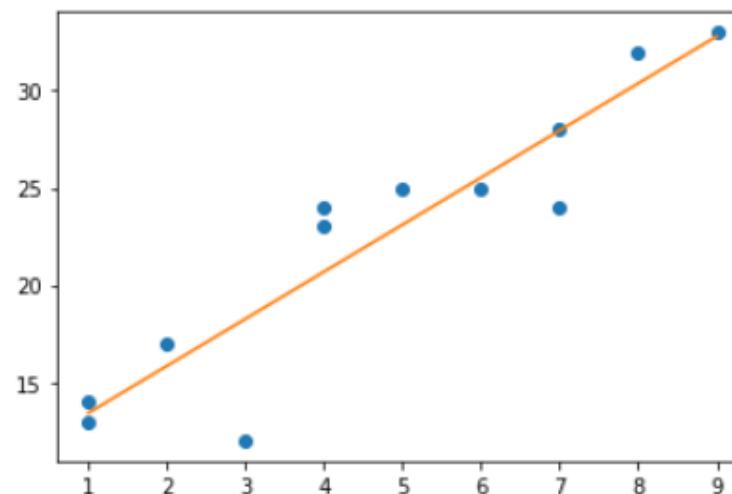


```
[1]: import numpy as np
import matplotlib.pyplot as plt

# 데이터 생성
x = np.array([1, 1, 2, 3, 4, 4, 5, 6, 7, 7, 8, 9])
y = np.array([13, 14, 17, 12, 23, 24, 25, 25, 24, 28, 32, 33])

m, b = np.polyfit(x, y, 1)
plt.plot(x, y, 'o')
plt.plot(x, m*x+b)
```

```
[1]: [matplotlib.lines.Line2D at 0x226212833d0>]
```

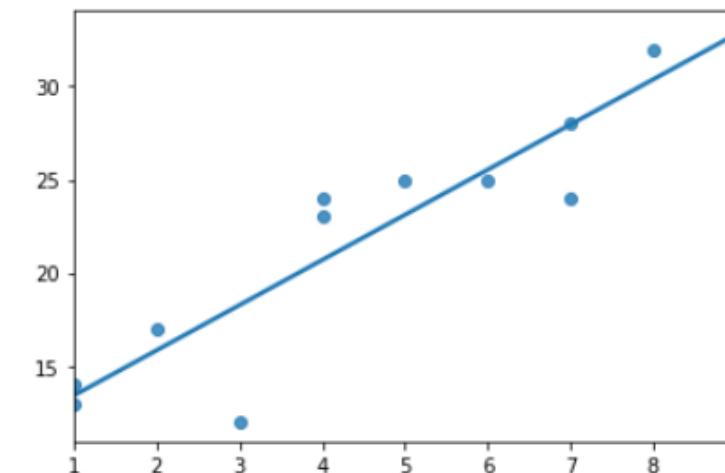


```
[1]: import numpy as np
import seaborn as sns

# 데이터 생성
x = np.array([1, 1, 2, 3, 4, 4, 5, 6, 7, 7, 8, 9])
y = np.array([13, 14, 17, 12, 23, 24, 25, 25, 24, 28, 32, 33])

sns.regplot(x, y, ci=None)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:51
    only valid positional argument will be `data`, and passing other arguments
    warnings.warn(
[1]: <AxesSubplot:>
```



05. matplotlib + seaborn

(2023년 11월 기준)

◆ 두 라이브러리 장점 위주 혼용

- ✓ 시각화 라이브러리 `seaborn`으로 기본 색상 + 글꼴 설정
- ✓ 시각화 라이브러리 `matplotlib`으로 시각화 틀 잡고 시각화 수행
- ✓ 각 라이브러리 특성에 맞추어서 선택적으로 활용하는 것 추천



★
★★★★

간단하고 깔끔하게 vs 구석구석 섬세하게
통계전문 시각화 vs 아무거나 시각화
밀도 함수 etc. vs 영상 해석 etc.

일단 그릴 때 명령어 난이도
손을 많이 댈 때 명령어 난이도

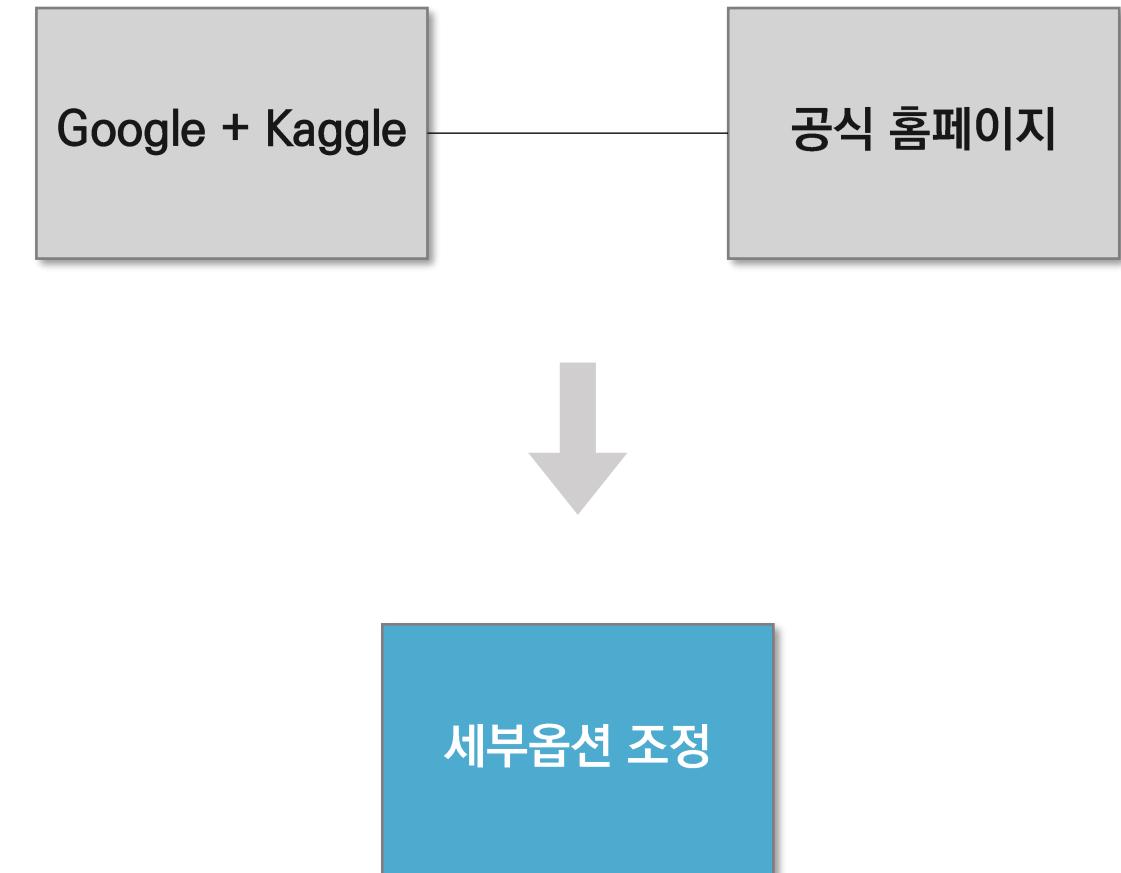
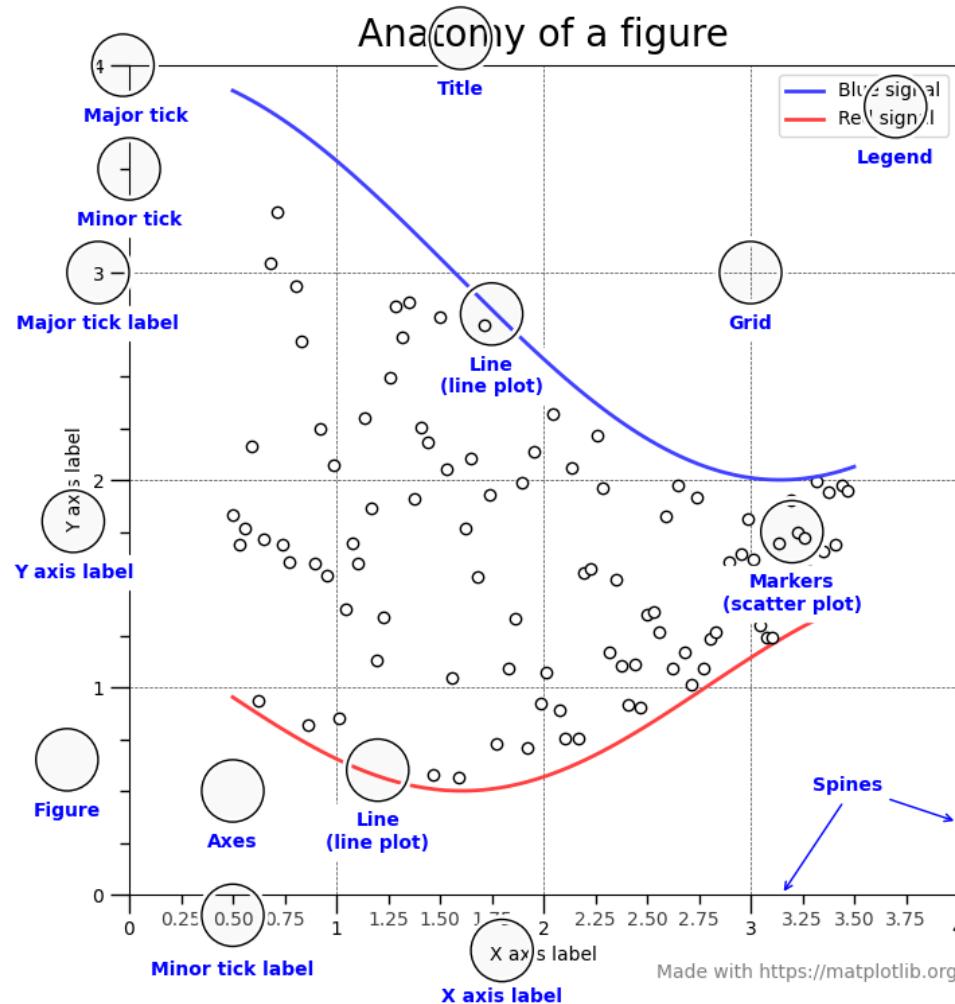


★ ★
★ ★ ★

추천 블로그 : https://jehyunlee.github.io/2020/10/10/Python-DS-37-seaborn_matplotlib4/

06. 세부 옵션

(2023년 11월 기준)



출처: <https://matplotlib.org/stable/gallery/showcase/anatomy.html>

◆ 객체 지향 방식으로 코드 작성 시작 추천

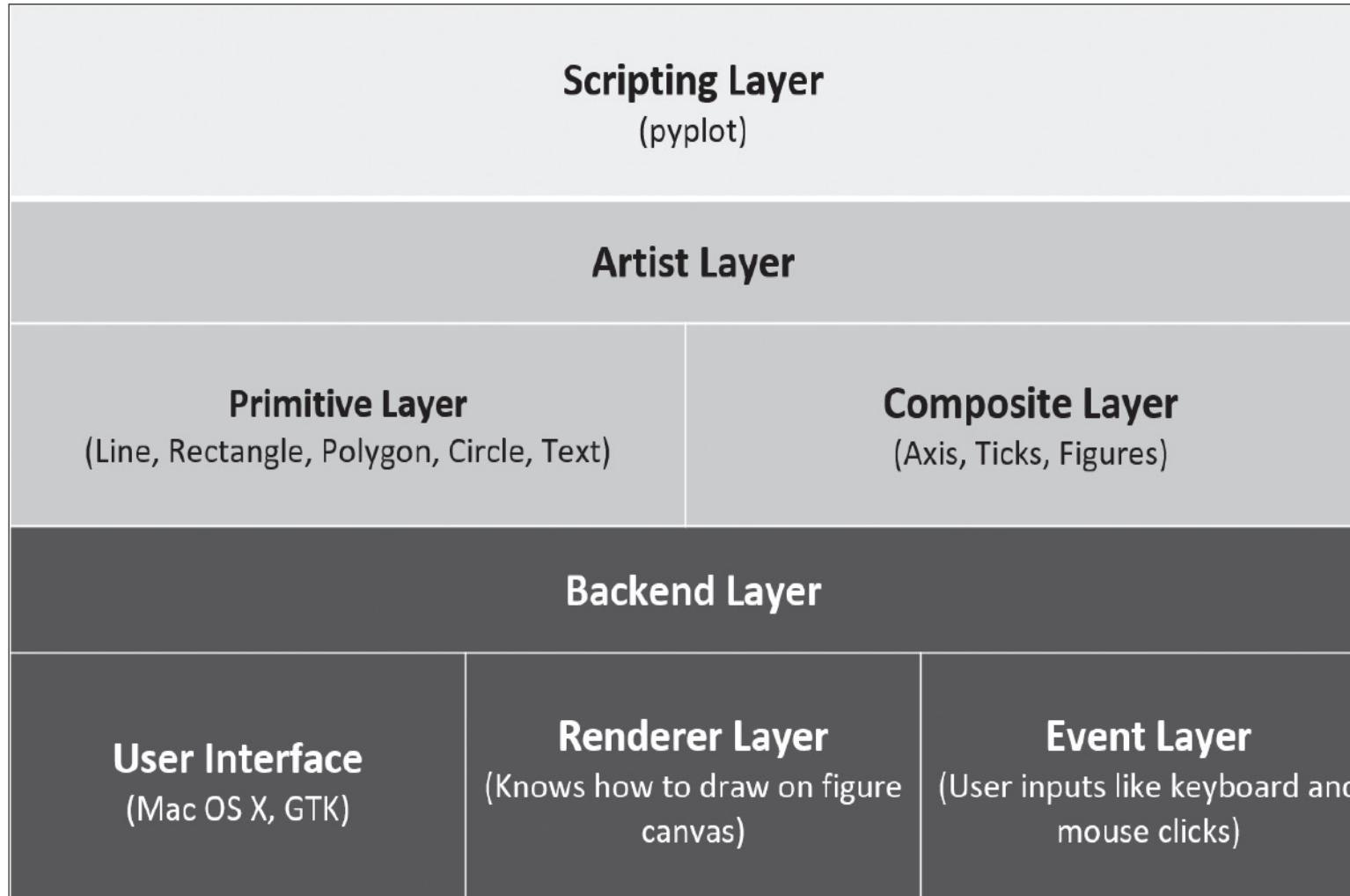
```
1 : import matplotlib.pyplot as plt  
2 : fig, ax = plt.subplots(figsize=(10, 6))  
3 : plt.show()  
  
(Graph)
```

주요 용어	설명
fig(figure의 약어)	<ul style="list-style-type: none">matplotlib에서 최상위 컨테이너 또는 창이며, plot을 만드는 일종의 캔버스하나의 figure에 하나 이상의 plot()을 포함할 수 있고, 다중 패널 또는 하위 plot 레이아웃을 그릴 수 있음
ax(axes의 약어)	<ul style="list-style-type: none">데이터를 그릴 수 있는 figure내의 subplot 또는 특정 영역제목(title), 레이블(labels), 격자선(gridlines) 등 같은 속성을 독립적으로 사용자 지정할 수 있음

08. matplotlib architecture

(2023년 11월 기준)

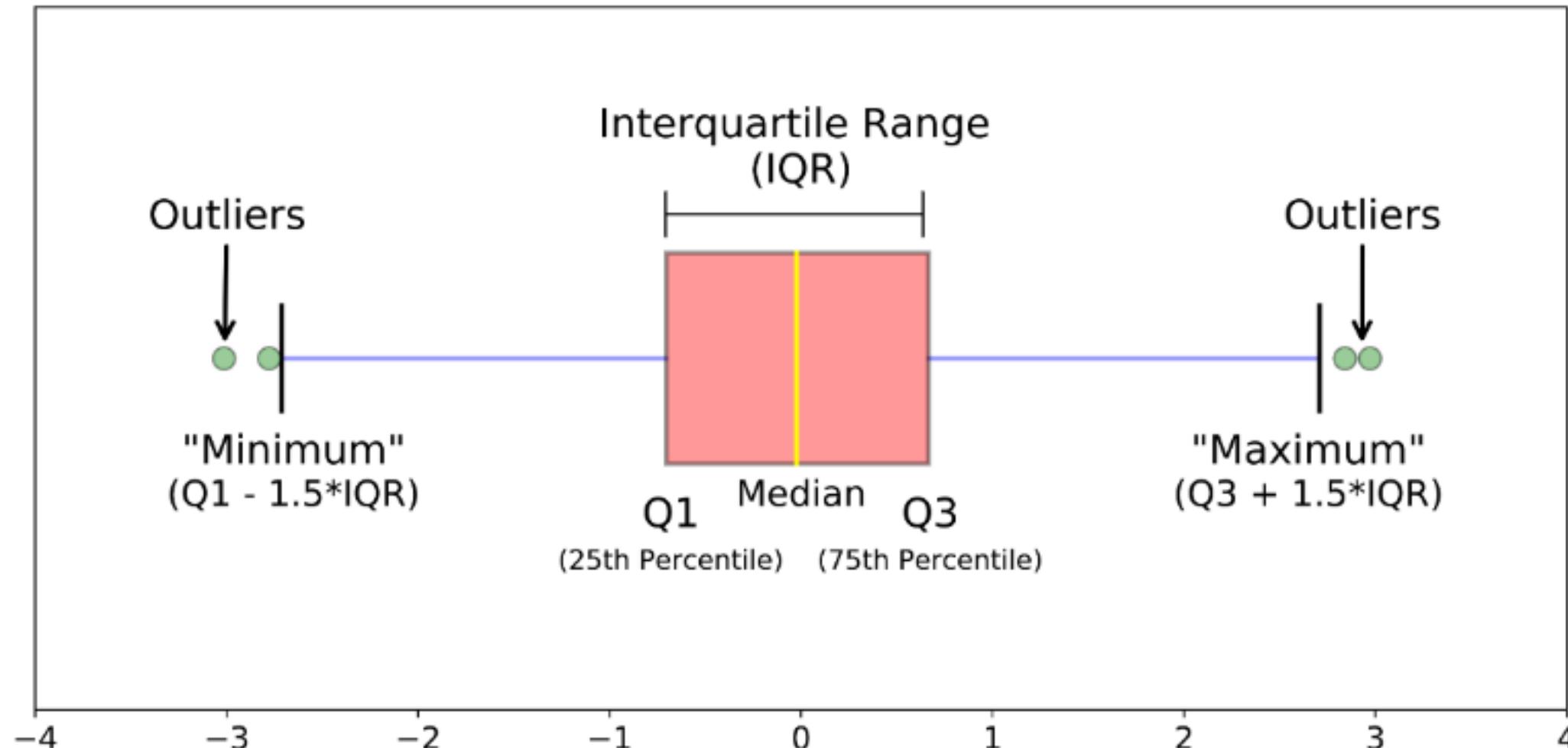
◆ matplotlib 라이브러리의 아키텍쳐



09. boxplot

(2023년 11월 기준)

- ◆ 박스플롯(Box Plot) : 데이터의 분포와 이상치(outlier)를 동시에 보여주는 시각화

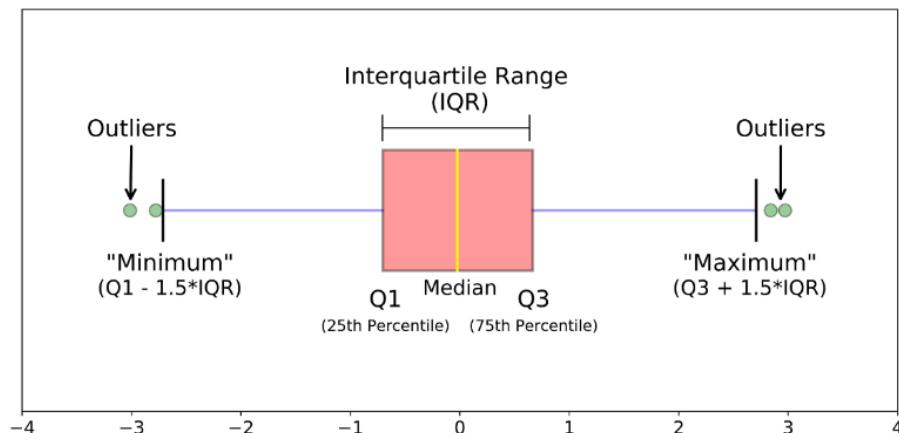


09. boxplot

(2023년 11월 기준)

- ◆ 박스플롯(Box Plot) : 데이터의 분포와 이상치(outlier)를 동시에 보여주는 시각화

구분	내용
Q1(제1사분위수)	데이터의 하위 25%에 해당하는 값
Q3(제3사분위수)	데이터의 상위 25%에 해당하는 값
IQR(사분위 범위)	상자의 길이는 사분위 범위를 나타내며 IQR은 데이터의 중간 50% 범위를 의미
중앙값(Median)	데이터의 중간 값을 나타냄
수염(Whiskers)	데이터의 변동 범위를 나타내며, $1.5 * \text{IQR}$ 규칙을 사용하여 그려짐
이상치(Outliers)	수염의 바깥에 위치하는 데이터 포인트, 일반적인 분포에서 벗어난 값들을 나타냄



Chapter 03. Matplotlib & Seaborn

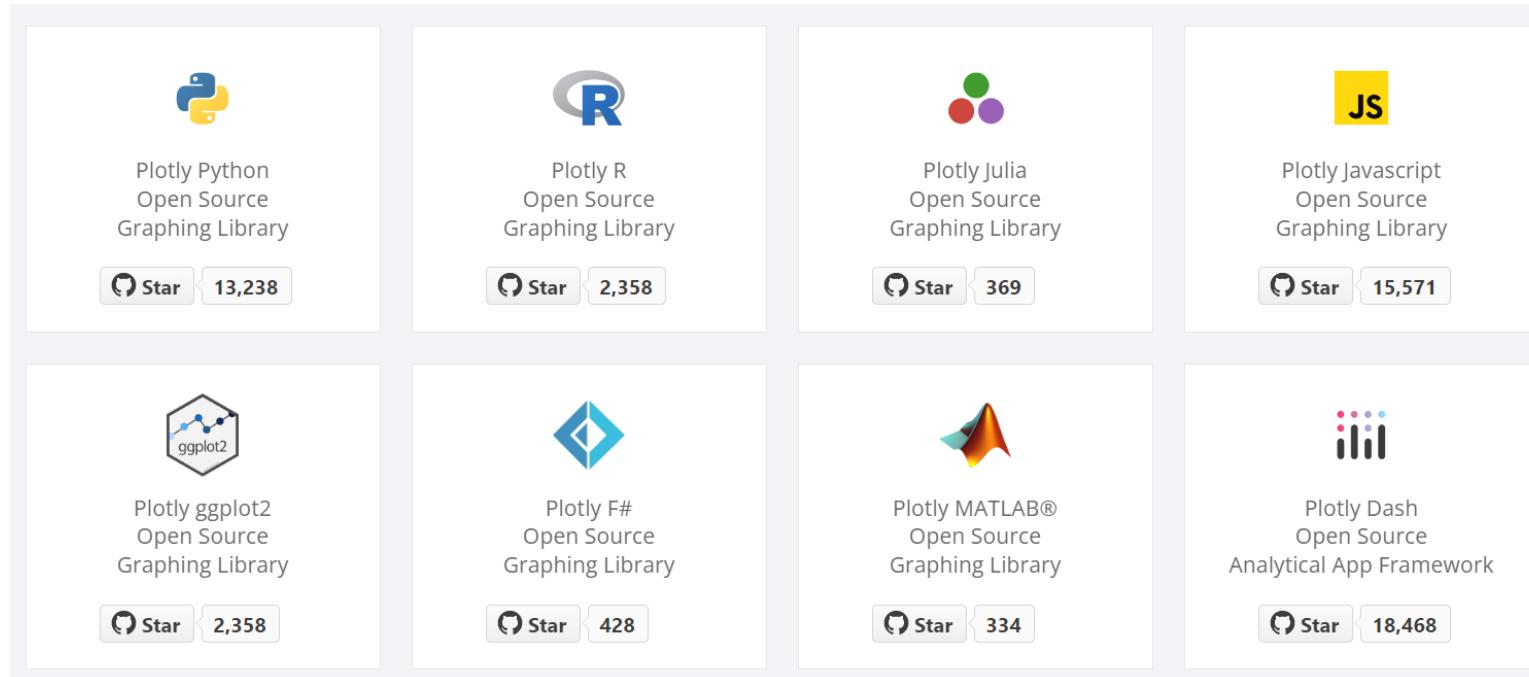
강의 실습 영상 참고

10. plotly

(2023년 11월 기준)

◆ plotly 소개

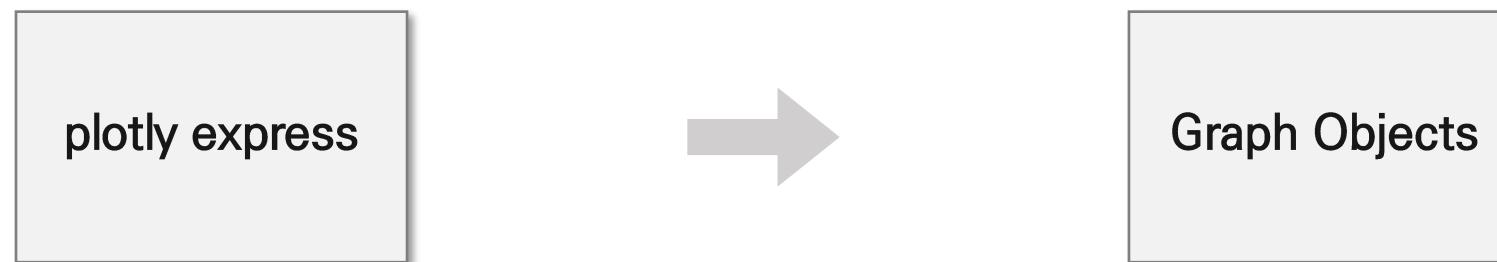
- ✓ 대화형 차트, 그래프 및 대시보드를 생성하기 위한 다양한 도구 제공
- ✓ 2013년, 캐나다에 회사 설립
- ✓ Dash Enterprise 제공, 다양한 언어에서 활용 가능



출처 : <https://plotly.com/graphing-libraries/>

◆ Graph Objects vs Plotly Express

Graph Objects	Plotly Express
Low Level Interface	High Level Interface
<ul style="list-style-type: none">세부적인 커스터마이징이 가능함복잡한 상호작용과 레이아웃 만들기 가능배울 것이 많음고급 사용자	<ul style="list-style-type: none">간단하고 직관적인 문법으로 쉽게 시각화 생성 가능다양한 표준 차트세밀한 커스터마이징은 제한됨입문자



출처 : <https://plotly.com/graphing-libraries/>

- 97 -

Chapter 03. plotly

강의 실습 영상 참고

파이썬 머신러닝

with  python™

01. scikit-learn 소개

(2023년 11월 기준)

- ◆ Python 머신러닝의 대표적인 라이브러리
 - ✓ 출처 : <https://scikit-learn.org/stable/>

Classification

Regression

Clustering

Dimensionality
Reduction

Model Selection

Preprocessing

01. scikit-learn 소개

(2023년 11월 기준)

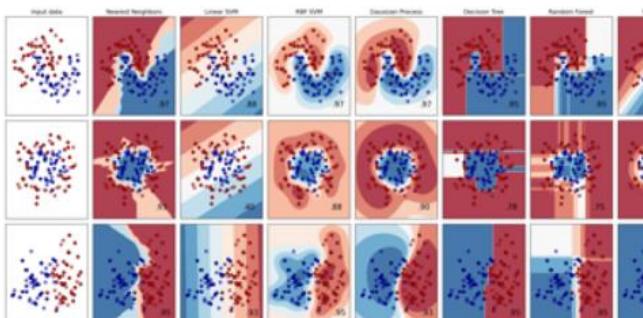
◆ Python 머신러닝의 대표적인 라이브러리

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

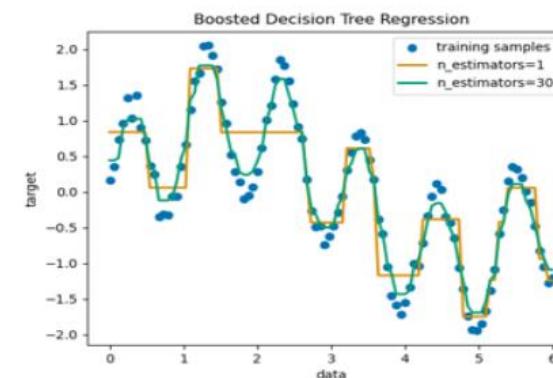


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

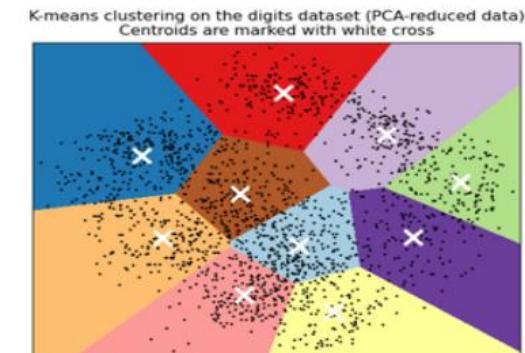


Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



01. scikit-learn 소개

(2023년 11월 기준)

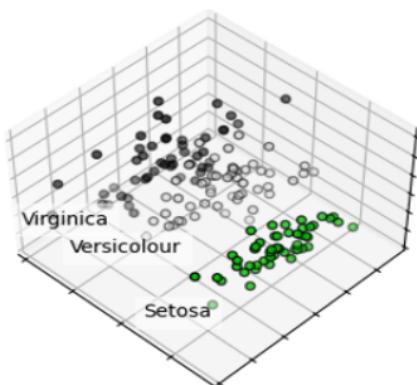
◆ Python 머신러닝의 대표적인 라이브러리

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: PCA, feature selection, non-negative matrix factorization, and more...

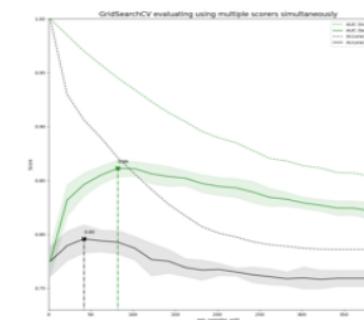


Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning

Algorithms: grid search, cross validation, metrics, and more...

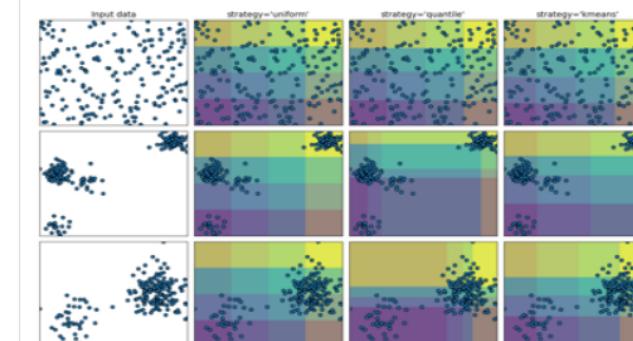


Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.

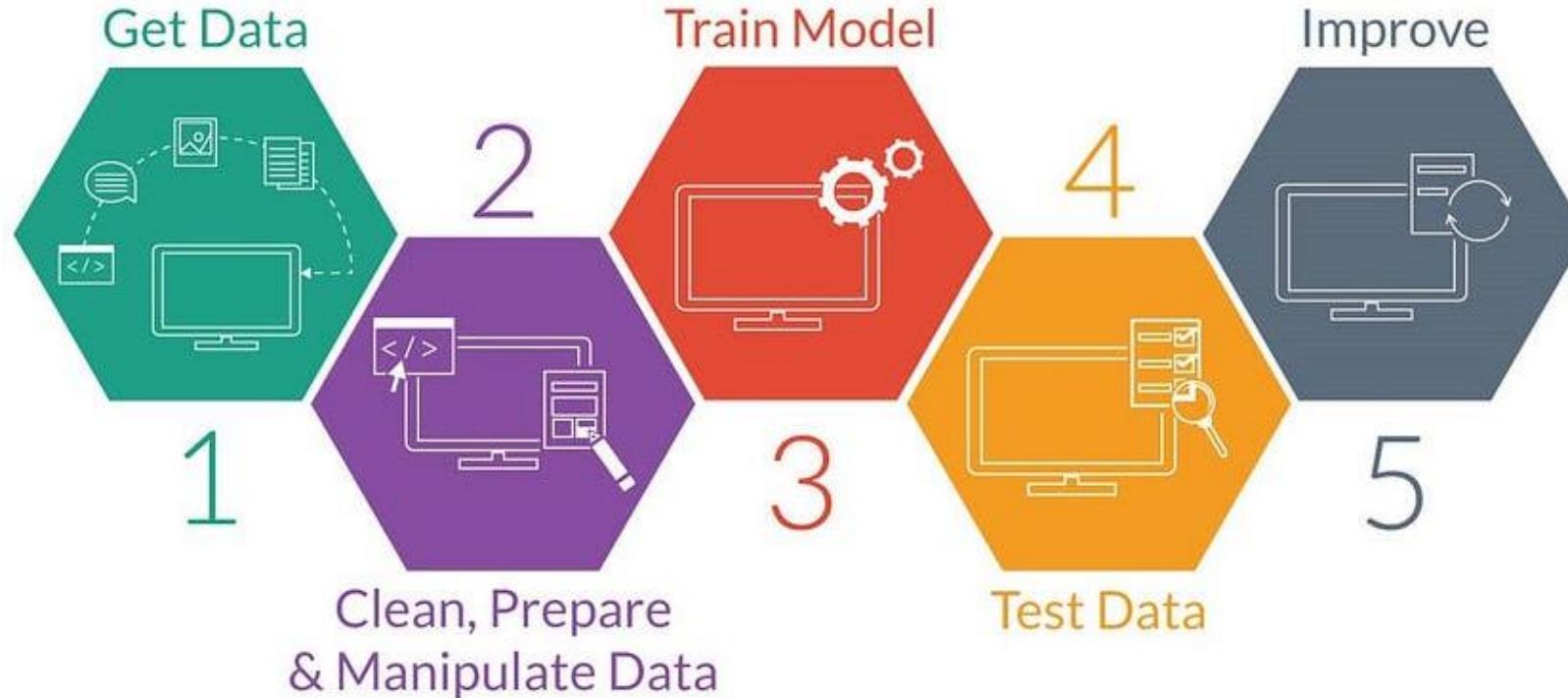
Algorithms: preprocessing, feature extraction, and more...



02. 머신러닝 프로세스

(2023년 11월 기준)

- ◆ 데이터 수집 > 데이터 가공 > 모델 학습 > 모델 평가 > 모델 성능 개선



출처 : <https://shorturl.at/zAEM4>

03. scikit-learn 주요 모듈

(2023년 11월 기준)

◆ 개요

- ✓ 머신러닝 분석 수행할 때 가장 유용하게 사용할 수 있는 파이썬 라이브러리

주요 업무	주요 모듈	설명
데이터 분리	sklearn.model_selection	<ul style="list-style-type: none">• 훈련, 검증, 테스트 데이터로 데이터를 분리하기 위해 활용• train_test_split 메서드 주로 활용
데이터 처리	sklearn.preprocessing	<ul style="list-style-type: none">• Feature Scaling<ul style="list-style-type: none">- StandardScaler : 평균 0, 분산 1로 기준으로 변경- MinMaxScaler : 0과 1 사이에 위치하도록 데이터 재조정• Binarization : 수치 데이터를 0과 1로 변환• Encoding : 문자 데이터를 수치로 변환• Imputer : 결측치를 채움
데이터 축소	sklearn.decomposition	<ul style="list-style-type: none">• PCA 등을 통해 차원축소 지원 가능
모형 학습	sklearn.linear_model	<ul style="list-style-type: none">• 선형회귀, 로지스틱 회귀 등의 알고리즘 지원
	sklearn.tree	<ul style="list-style-type: none">• 트리 알고리즘 제공
	sklearn.ensemble	<ul style="list-style-type: none">• 랜덤포레스트 알고리즘 제공
모형 평가	sklearn.metrics	<ul style="list-style-type: none">• 분류, 회귀, 클러스터링 등에 관한 모형 평가 지표 제공

03. scikit-learn 주요 모듈

(2023년 11월 기준)

◆ 머신러닝 모형 학습 전, 데이터 전처리 이유

- ✓ 머신러닝 알고리즘 적용 전, 결측치 미 허용 → 다른 고정 값 입력 필요
- ✓ 문자열 값을 허용하지 않는다. → 숫자형으로 변환
- ✓ 서로 다른 변수의 값 범위를 일정한 수준으로 변환함

종류	모듈	설명
표준화 Standardization	RobustScaler	<ul style="list-style-type: none">중간값을 제거하고, 사분위수 범위에 따라 데이터의 배율 조정1사분위 3사분위 사이의 범위이상치 제거에 좋음quantile_range 파라미터(default [0.25, 0.75])에서 조정 가능
	StandardScaler	<ul style="list-style-type: none">평균이 0이고 분산이 1이 정규분포를 가진 값으로 변환SVM, 선형회귀, 로지스틱 모델 사용 시, 필수회귀보다는 분류분석에 유용
정규화 Normalization	MinMaxScaler	<ul style="list-style-type: none">모든 수치 데이터를 [0, 1], 음수가 존재하면 [-1, 1] 사이 변환분류보다는 회귀에 유용
	MaxAbsScaler	<ul style="list-style-type: none">최대절댓값과, 0이 각각 1, 0이 되도록 스케일링 하는 정규화모든 값은 -1과 1사이에 표현이상치에 매우 민감,분류보다는 회귀에 유용

03. scikit-learn 주요 모듈

(2023년 11월 기준)

◆ 머신러닝 모형 학습 전, 데이터 전처리 이유

- ✓ 머신러닝 알고리즘 적용 전, 결측치 미 허용 → 다른 고정 값 입력 필요
- ✓ 문자열 값을 허용하지 않는다. → 숫자형으로 변환
- ✓ 서로 다른 변수의 값 범위를 일정한 수준으로 변환함

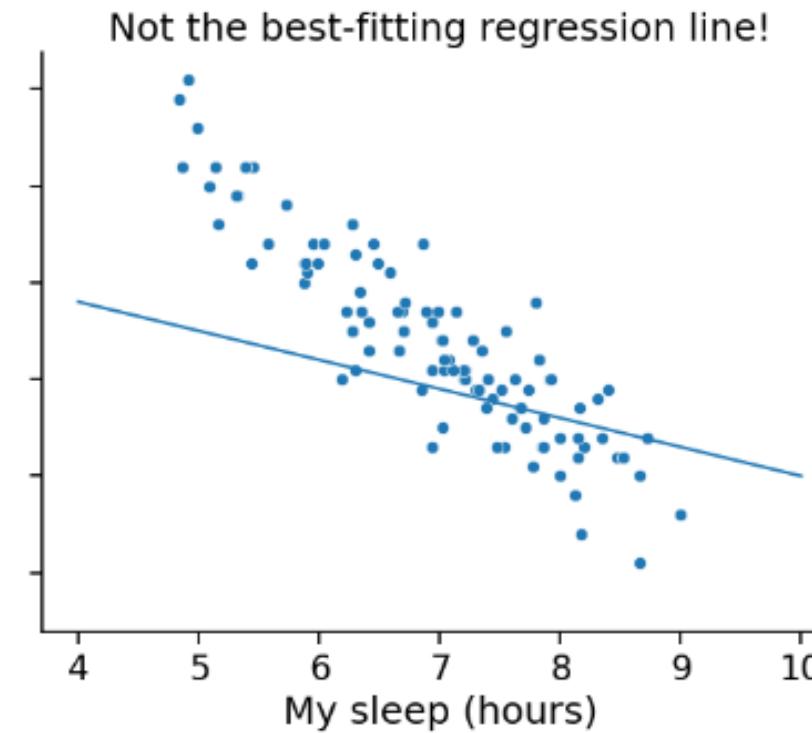
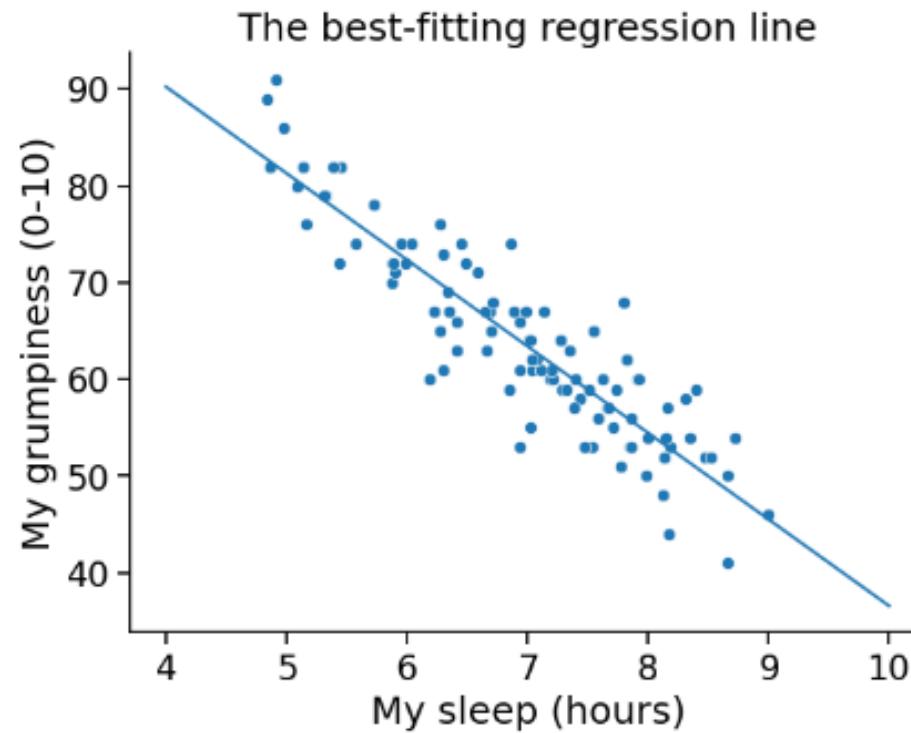
종류	모듈	설명
Ordinal Encoding	OrdinalEncoder	<ul style="list-style-type: none">• 서열(Ordinal) 척도를 숫자로 변환함 (독립변수만)
Label Encoding	LabelEncoder	<ul style="list-style-type: none">• 서열(Nominal) 척도를 숫자로 변환함 (종속변수만)
One-Hot Encoding	OneHotEncoder	<ul style="list-style-type: none">• 명목(Nominal) 척도를 숫자로 변환함 (독립변수만)

04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 회귀 모형 오차의 개념

- ✓ 오차는 실제값과 예측값의 차이를 말함, 양(+)의 값과 음(-)의 값 발생
- ✓ 양과 음의 값 한산 시, 오차는 0에 가까워짐, 따라서, 제곱 또는 절댓값을 취함
- ✓ 회귀 모형은 오차의 제곱 혹은 절댓값의 합이 최소화되는 라인을 찾는 것이 목표임

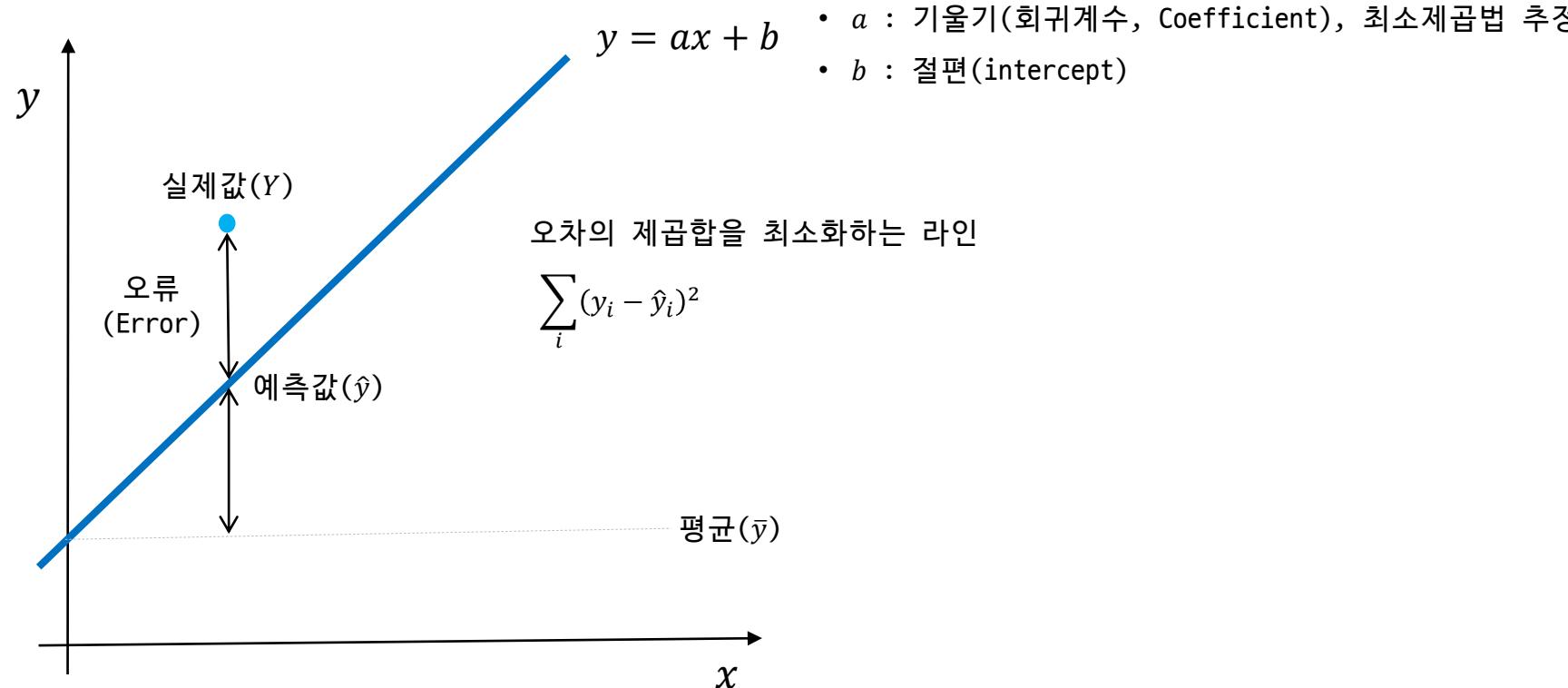


04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 회귀 모형 오차의 개념

- ✓ 오차는 실제값과 예측값의 차이를 말함, 양(+)의 값과 음(-)의 값 발생
- ✓ 양과 음의 값 한산 시, 오차는 0에 가까워짐, 따라서, 제곱 또는 절댓값을 취함
- ✓ 회귀 모형은 오차의 제곱 혹은 절댓값의 합이 최소화되는 라인을 찾는 것이 목표임



◆ 성능평가 - 회귀 : MAE(Mean Absolute Error)

- ✓ 실젯값과 예측값의 차이 활용

정의	<ul style="list-style-type: none">실젯값과 예측값의 차이를 절댓값으로 변환해 평균한 것
수식	$MAE = \frac{1}{n} \sum_{i=1}^n Y_i - \hat{Y}_i $
특징	<ul style="list-style-type: none">에러의 크기가 그대로 반영이상치에 영향 받음

◆ Python Code

```
1 : from sklearn.metrics import mean_absolute_error  
2 : mae = mean_absolute_error(y_test, y_pred)  
    0.00
```

04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 회귀 : MSE(Mean Squared Error)

- ✓ 실젯값과 예측값의 차이 활용

정의	<ul style="list-style-type: none">• 실젯값과 예측값의 차이를 제곱해 평균한 것
수식	$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$
특징	<ul style="list-style-type: none">• 실젯값과 예측값 차이의 면적 합을 의미• 특이값이 존재하면 수치가 증가

◆ Python Code

```
1 : from sklearn.metrics import mean_squared_error  
2 : mae = mean_squared_error (y_test, y_pred)  
    0.00
```

◆ 성능평가 - 회귀 : RMSE(Root Mean Squared Error)

- ✓ 실젯값과 예측값의 차이 활용

정의	<ul style="list-style-type: none">실젯값과 예측값의 차이를 제곱해 평균한 것에 루트를 쓴 것
수식	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$
특징	<ul style="list-style-type: none">오차에 제곱을 하면 오차가 클수록 그에 따른 가중치가 높게 반영이 때, 손실이 기하급수적으로 증가하는 상황에서 실제 오류값의 평균보다 값이 더 커지지 않도록 상쇄하기 위해 사용

◆ Python Code

```
1 : from sklearn.metrics import mean_squared_error  
2 : import numpy as np  
3 : mse = mean_squared_error(y_test, y_pred)  
4 : rmse = np.sqrt(mse)
```

◆ 성능평가 - 회귀 : MSLE(Mean Squared Log Error)

- ✓ 실젯값과 예측값의 차이 활용

정의	<ul style="list-style-type: none">실젯값과 예측값의 차이를 제곱해 평균한 것에 로그를 적용
수식	$MSLE = \log \left(\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \right)$
특징	<ul style="list-style-type: none">RMSE와 같이 오차가 기하급수적으로 증가하는 상황에서 실제 오차의 평균보다 값이 더 커지지 않도록 상쇄

◆ Python Code

```
1 : from sklearn.metrics import mean_squared_log_error  
2 : msle = mean_squared_log_error(y_test, y_pred)  
3 : msle  
0.00
```

◆ 성능평가 - 회귀 : MAPE(Mean Absolute Error)

- ✓ 실젯값과 예측값의 차이 활용

정의	<ul style="list-style-type: none">MAE를 퍼센트로 반환한 것
수식	$MAPE = \frac{n}{100} \sum_{i=1}^n \left \frac{y_i - \hat{f}(x_i)}{y_i} \right $
특징	<ul style="list-style-type: none">오차가 예측값에서 차지하는 정도를 나타냄

◆ Python Code

```
1 : import numpy as np
2 : def MAPE(y_test, y_pred):
3 :     mape = np.mean(np.abs((y_test - y_pred) / y_test)) * 100
4 :     return mape
5 : mape = MAPE(y_test, y_pred)
6 : mape
```

04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 회귀 : 결정계수(R^2)

- ✓ 실젯값과 예측값의 차이 활용

정의	<ul style="list-style-type: none">회귀 모형이 선형인 경우에는 결정계수를 평가 지표로 활용결정계수는 0에서 1사이의 값을 가지며 1에 가까울수록 모형의 설명력이 좋음
수식	$R^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST} = 1 - \frac{SSE}{SST}$

지표	설명	수식
SST (Total Sum of Squared)	<ul style="list-style-type: none">전체제곱합실제 관측치(y_i)와 y값들의 평균(\bar{y})의 차이를 제곱하여 합한 값y가 가지는 전체 변동	$\sum_{i=1}^n (y_i - \bar{y})^2$
SSR (Regression Sum of Squared)	<ul style="list-style-type: none">회귀제곱합모형 예측치(\hat{y}_i)와 y값들의 평균(\bar{y})의 차이를 제곱하여 합한 값y가 가지는 전체 변동성 중 회귀 모형으로 설명할 수 있는 변동	$\sum_{i=1}^n (\hat{y}_i - \bar{y})^2$
SSE (Error Sum of Squared)	<ul style="list-style-type: none">오차제곱합실제 관측치(y_i)와 모형 예측치(\hat{y}_i)의 차이를 제곱하여 합한 값y가 가지는 전체 변동성 중 회귀 모형으로 설명할 수 없는 변동	$\sum_{i=1}^n (y_i - \hat{y}_i)^2$

04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 회귀 : 결정계수(R^2)

- ✓ 실젯값과 예측값의 차이 활용

정의	<ul style="list-style-type: none">회귀 모형이 선형인 경우에는 결정계수를 평가 지표로 활용결정계수는 0에서 1사이의 값을 가지며 1에 가까울수록 모형의 설명력이 좋음
수식	$R^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST} = 1 - \frac{SSE}{SST}$

◆ Python Code

```
1 : from sklearn.metrics import r2_score  
2 : r2 = r2_score(y_test, y_pred)  
3 : r2  
0.92
```

04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 분류 : 혼동행렬

		예측 결과	
		TRUE	FALSE
실제 정답	TRUE	TP (True Positive)	FN (False Negative)
	FALSE	FP (False Positive)	TN (True Negative)

◆ 용어 정리

- TN(True Negative) : 음성을 음성이라고 예측
- FP(False Positive) : 음성을 양성이라고 예측
- FN(False Negative): 양성을 음성이라고 예측
- TP(True Positive) : 양성을 양성이라고 예측

04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 분류 : 혼동행렬

- ✓ 정확도의 한계점 보완하기 위해 혼동행렬 활용

정 의	<ul style="list-style-type: none">모델의 성능을 평가할 때 사용되는 지표, 예측값과, 실제값을 보여주는 행렬
특 징	<ul style="list-style-type: none">4분면 행렬에서 실제 레이블 클래스 값과 예측 레이블 클래스 값이 어떤 유형을 가지고 매핑되는지 나타남

◆ Python Code

```
1  :  from sklearn.metrics import confusion_matrix  
2  :  cm = confusion_matrix(y_test, y_pred)  
3  :  cm
```

04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 분류 : 정확도(Accuracy)

- ✓ 실제분류와 예측분류가 얼마나 일치했는가를 기반으로 알고리즘의 성능을 평가

정 의	• 실제 데이터에서 예측 데이터가 얼마나 같은지 판단하는 지표
수 식	$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$
특 징	• 데이터 구성에 따라 머신러닝 모델의 성능을 왜곡할 가능성 있음

◆ Python Code

```
1 : from sklearn.metrics import accuracy_score  
2 : acc = accuracy_score(y_test, y_pred)  
3 : acc
```

		예측 결과	
		TRUE	FALSE
실제 정답	TRUE	TP (True Positive)	FN (False Negative)
	FALSE	FP (False Positive)	TN (True Negative)

◆ 성능평가 - 분류 : 정밀도(Precision)

- ✓ Positive 데이터 예측에 집중한 평가지표

정의	<ul style="list-style-type: none"> Positive로 예측한 것들 중 실제로도 Positive인 것들의 비율
수식	$Precision = \frac{TP}{FN + TP}$
특징	<ul style="list-style-type: none"> Positive 예측성능을 더욱 정밀하게 측정하기 위한 평가지표 양성 예측도라 불리움 정밀도가 상대적인 중요성을 가지는 경우 <ul style="list-style-type: none"> - 실제 Negative인 데이터를 Positive로 잘못 예측했을 때 업무상 큰 영향 발생 시

◆ Python Code

```

1 : from sklearn.metrics import precision_score
2 : acc = precision_score(y_test, y_pred)
3 : acc

```

		예측 결과	
		TRUE	FALSE
실제 정답	TRUE	TP (True Positive)	FN (False Negative)
	FALSE	FP (False Positive)	TN (True Negative)

04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 분류 : 재현율(Recall)

- ✓ Positive 데이터 예측에 집중한 평가지표

정의	<ul style="list-style-type: none">실제 Positive인 것들 중 Positive로 예측한 것들의 비율
수식	$Recall = \frac{TP}{FP + TP}$
특징	<ul style="list-style-type: none">민감도(Sensitivity) 또는 TPR(True Positive Rate)라고 불림재현율이 상대적인 중요성을 가지는 경우<ul style="list-style-type: none">- 실제 Positive인 데이터를 Negative로 잘못 예측 했을 때 업무상 큰 영향이 발생할 때

◆ Python Code

```
1 : from sklearn.metrics import recall_score  
2 : recall = recall_score(y_test, y_pred)  
3 : recall
```

		예측 결과	
		TRUE	FALSE
실제 정답	TRUE	TP (True Positive)	FN (False Negative)
	FALSE	FP (False Positive)	TN (True Negative)

04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 분류 : F1 스코어

- ✓ 정밀도와 재현율을 결합한 성능지표

정의	<ul style="list-style-type: none">실제 Positive인 것들 중 Positive로 예측한 것들의 비율
수식	$F1\ score = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = 2 \times \frac{precision \times recall}{precision + recall}$
특징	<ul style="list-style-type: none">정밀도와 재현율이 어느 한쪽으로 치우치지 않고 적절한 조화를 이룰 때 상대적으로 높은 수치를 나타냄

◆ Python Code

```
1 : from sklearn.metrics import f1_score  
2 : f1 = f1_score(y_test, y_pred)  
3 : f1
```

		예측 결과	
		TRUE	FALSE
실제 정답	TRUE	TP (True Positive)	FN (False Negative)
	FALSE	FP (False Positive)	TN (True Negative)

04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 분류 : ROC-Curve

- ✓ ROC 곡선과 이를 기반으로 하는 AUC 스코어는 이진 분류모델의 주요 성능평가지표임

정의	<ul style="list-style-type: none">FPR(False Positive Rate)이 변할 때, TPR(True Positive Rate)이 변하는 것을 나타내는 곡선(ROC)
수식	$TNR = \frac{TN}{FP + TN}$ $FPR = 1 - TNR = \frac{FP}{FP + TN}$
특징	<ul style="list-style-type: none">TPR을 y축으로, FPR을 x축으로 하는 그래프분류 결정 임곗값을 조절하면서 FPROI 0부터 1까지 변할 때, TPR의 변화값을 그래프에 나타냄우상향 그래프로 그려짐

		예측 결과	
		TRUE	FALSE
실제 정답	TRUE	TP (True Positive)	FN (False Negative)
	FALSE	FP (False Positive)	TN (True Negative)

04. 머신러닝 성능평가

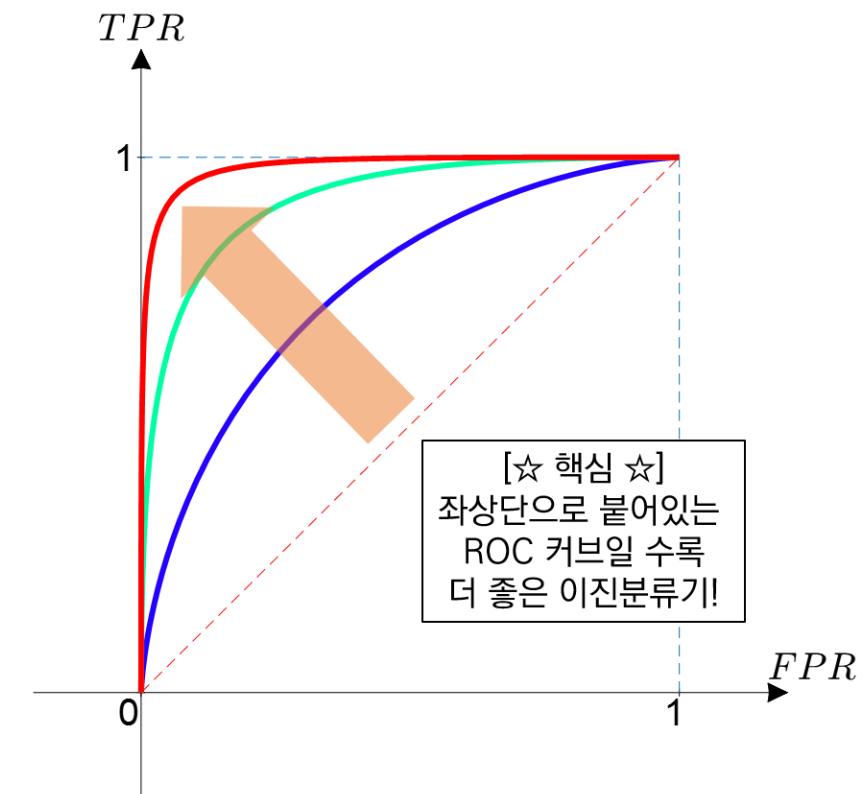
(2023년 11월 기준)

◆ 성능평가 - 분류 : ROC-Curve

- ✓ ROC 곡선과 이를 기반으로 하는 AUC 스코어는 이진 분류모델의 주요 성능평가지표임

◆ Python Code

```
1  : from sklearn.metrics import roc_curve  
2  : import matplotlib.pyplot as plt  
3  : fpr, tpr, thres = roc_curve(y_test, y_pred, pos_label = 1)  
4  : plt.plot(fpr, tpr)  
5  : plt.show()
```



04. 머신러닝 성능평가

(2023년 11월 기준)

◆ 성능평가 - 분류 : AUC 스코어

- ✓ ROC 곡선 아래의 면적 값을 분류 성능지표로 사용

정의	<ul style="list-style-type: none">• Area Under the ROC Curve• ROC 곡선 아래의 면적• 1에 가까울수록 예측성능이 우수하다고 판단
특징	<ul style="list-style-type: none">• AUC 값이 커지려면 FPR이 작을 때 TPR 값이 커야 함• 우상향 직선에서 멀어지고 왼쪽 상단의 모서리 쪽으로 가파르게 곡선이 이동할수록 AUC가 1에 가까워짐• 랜덤 수준의 AUC값은 0.5

◆ Python Code

```
1 : from sklearn.metrics import roc_curve, auc
2 : fpr, tpr, thres = roc_curve(y_test, y_pred, pos_label = 1)
3 : auc = auc(fpr, tpr)
5 : auc
```

◆ 시계열 데이터(time series data)는 일정한 시간 간격으로 순차적으로 기록된 데이터의 집합

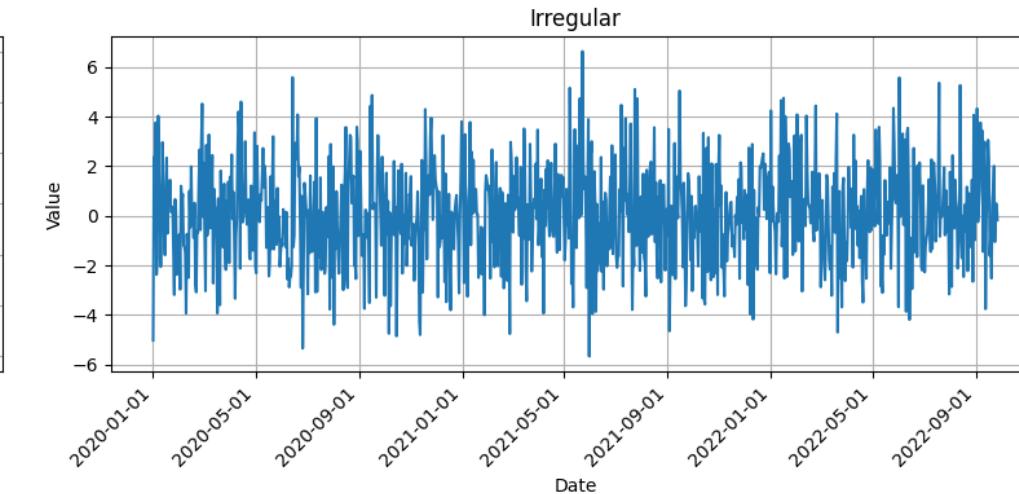
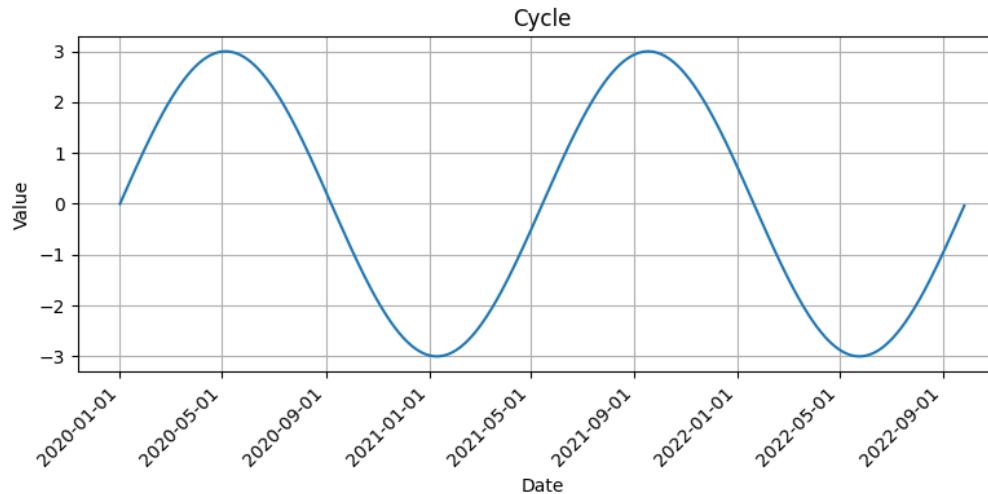
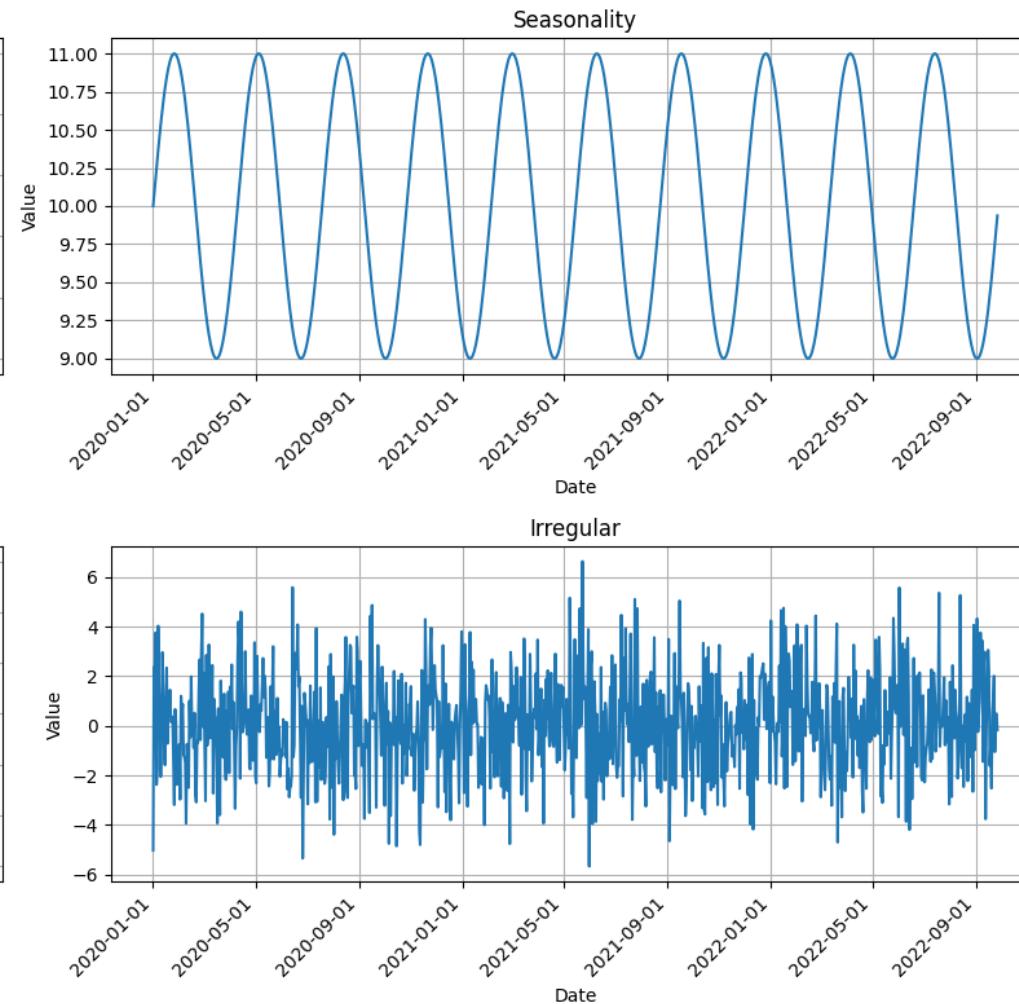
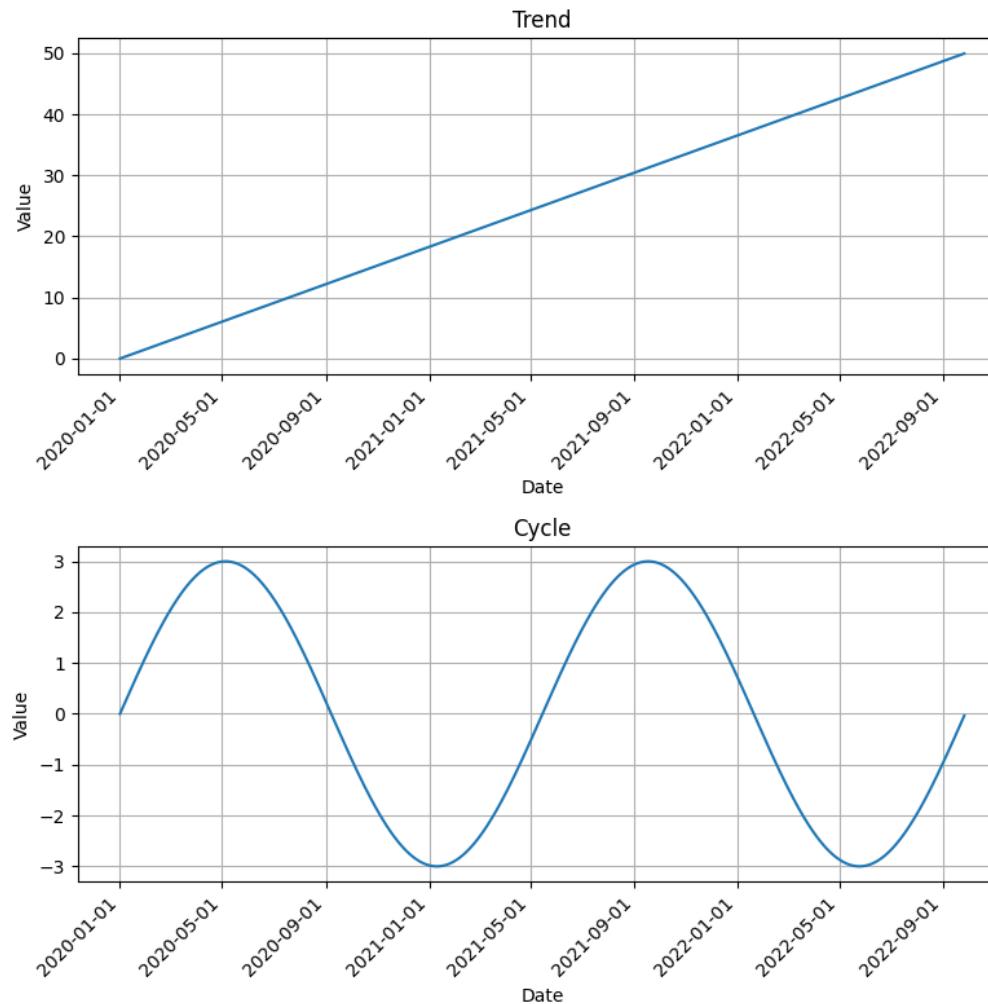
- ✓ 각 데이터 포인트들은 시간 순서대로 연결되어 있음
- ✓ 과거의 데이터가 미래의 데이터에 영향을 미칠 수 있음

주요 특성	설명
계절요인 (Seasonal Factor)	<ul style="list-style-type: none">• 연간, 월간, 주간 등 일정한 주기를 가지고 반복되는 패턴이 있을 수 있음• 예: 추석, 설, 크리스마스
추세요인 (Trend Factor)	<ul style="list-style-type: none">• 장기적인 시간 동안 증가, 감소 또는 안정적인 경향을 보일 수 있음• 예: GDP, 인구증가율, 출산율
순환요인 (Cycle Factor)	<ul style="list-style-type: none">• 특정 주기 혹은 수년 간의 간격으로 발생하는 주기적인 패턴• 예: 경기 변동성
불규칙요인 (Irregular Factor)	<ul style="list-style-type: none">• 예측할 수 없는 또는 설명할 수 없는 요인에 의한 우연한 패턴(예측 불가), noise으로 간주됨• 예: 전쟁, 홍수, 전염병

05. 시계열 데이터 개요

(2023년 11월 기준)

- ◆ 시계열 데이터(time series data)는 일정한 시간 간격으로 순차적으로 기록된 데이터의 집합



◆ 정상성(Stationary)

✓ 시계열분석을 하기 위해서는 기본적으로 평균, 분산, 공분산 및 기타 모든 분포적 특성이 일정한 성질인 정상성 만족

구분	설명
기본 조건	<ul style="list-style-type: none">평균(Mean)은 시점에 관계 없이 일정분산(Variance)은 시점에 관계 없이 일정공분산(Covariance)은 시계열 내의 특정 시점에 의존하지 않음, 다만 시차에만 의존계절성 또는 주기성이 없음
정상성 가정 효과	<ul style="list-style-type: none">모델링 단순화, ARIMA 모델은 정상성을 가정함. 즉 정상성을 만족하지 못하면 ARIMA 모델 사용 불가신뢰할 수 있는 통계적 추론, 모델에 의해 추정된 파라미터들이 시간에 따라 일관된 신뢰 보장
정상성 검정	<ul style="list-style-type: none">Augmented Dickey-Fuller(ADF) 검정 : 대표적인 정상성 검정
정상성 변환방법	<ul style="list-style-type: none">추세가 안 보이거나 평균이 일정하지 않을 시: 차분(Difference)을 통해서 가공분산이 일정하지 않은 경우: 로그 변환, 제곱근 변환, 역 변환 등을 통해 시계열 가공추세 제거 : 데이터에서 추세 성분을 제거분해 : 시계열에서 계절성과 추세 성분을 분리하고 제거

06. 시계열 데이터 예측

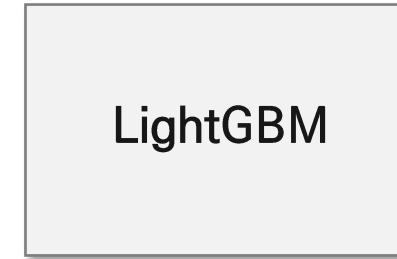
(2023년 11월 기준)

◆ 시계열 데이터 예측 주요 알고리즘 : 알고리즘 코드 적용하기

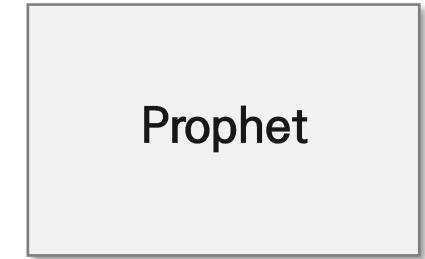
< 전통 시계열 >



< Tree-Boosting >



< Facebook >



◆ ARIMA(AutoRegressive Integrated Moving Average)

- ✓ AR(자기회귀), I(차분), MA(이동평균)의 합성, 기존 AR, MA, ARMA 모델의 경우 데이터가 정상성이어야 함
- ✓ 올바른 AR 및 MA 모델의 차수, I(차분)의 횟수 결정하는 것이 ARIMA 모델의 핵심
- ✓ 모델 평가 : Akaike 정보 기준 활용
- ✓ sktime 라이브러리 : 각 차수 및 횟수를 자동으로 찾아줌
 - <https://github.com/sktime/sktime>

Welcome to sktime

A unified interface for machine learning with time series

🚀 Version 0.24.1 out now! [Check out the release notes here.](#)

sktime is a library for time series analysis in Python. It provides a unified interface for multiple time series learning tasks. Currently, this includes time series classification, regression, clustering, annotation, and forecasting. It comes with [time series algorithms](#) and [scikit-learn](#) compatible tools to build, tune and validate time series models.



06. 시계열 데이터 예측 - ARIMA

강의 실습 영상 참고

◆ LightGBM(Light Gradient Boosting Machine)

- ✓ MS에서 개발한 Gradient Boosting Framework, 대규모 데이터셋을 빠르고 효율적으로 처리
- ✓ 기존 Boosting 모델보다 메모리 사용량이 적고, 실행 속도가 빠르며, 더 높은 정확도를 달성함
- ✓ 분류 및 다중분류 문제, 수치 예측 문제에 효과적
- ✓ 소규모 데이터셋에서의 과적합(Overfitting), 파라미터 튜닝 설계 시, 복잡할 수 있음
- ✓ 그 외 상세 내용 : <https://lightgbm.readthedocs.io/en/stable/>

Welcome to LightGBM's documentation!

LightGBM is a gradient boosting framework that uses tree based learning algorithms.
advantages:

- Faster training speed and higher efficiency.
- Lower memory usage.
- Better accuracy.
- Support of parallel, distributed, and GPU learning.
- Capable of handling large-scale data.



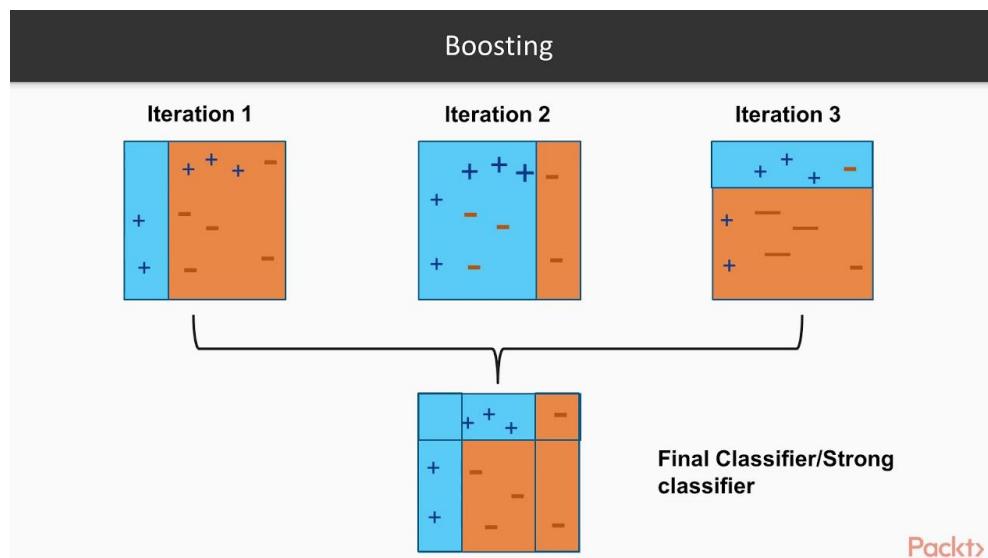
06. 시계열 데이터 예측

(2023년 11월 기준)

- ◆ 머신러닝 모형 : LightGBM(Light Gradient Boosting Machine)

- ◆ GBM의 기본 원리

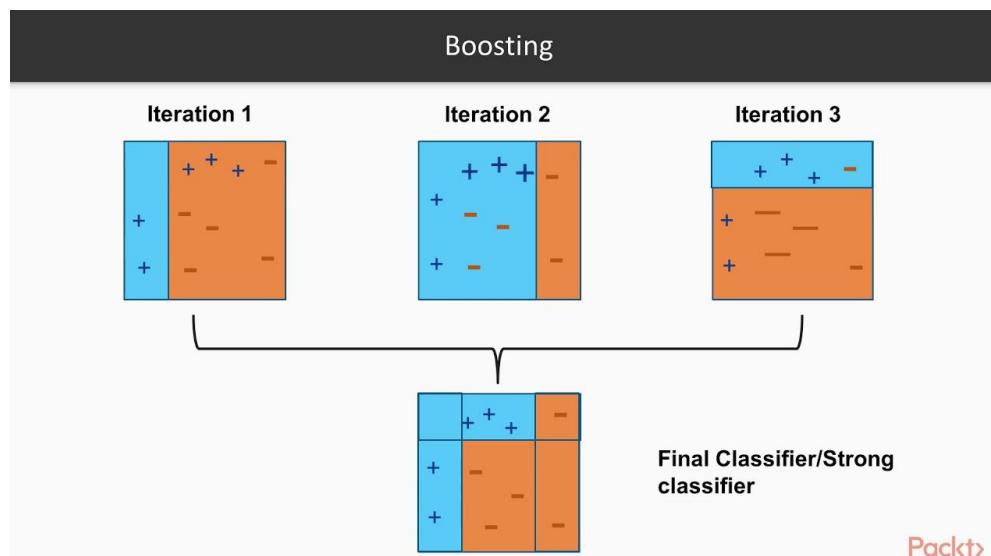
- ✓ 첫번째 단계의 Iteration 1을 활용하여 Y 예측
- ✓ 발생한 잔차(residual)을 다시 Iteration 2의 input으로 넣어주고 다시 예측
- ✓ 발생한 잔차(residual)을 다시 Iteration 3의 input으로 넣어주고 다시 예측
- ✓ residual은 점차 계속 작아질 것임



- ◆ 머신러닝 모형 : LightGBM(Light Gradient Boosting Machine)

- ◆ GBM의 기본 원리

- ✓ 첫번째 단계의 Iteration 1을 활용하여 Y 예측
- ✓ 발생한 잔차(residual)을 다시 Iteration 2의 input으로 넣어주고 다시 예측
- ✓ 발생한 잔차(residual)을 다시 Iteration 3의 input으로 넣어주고 다시 예측
- ✓ residual은 점차 계속 작아질 것임
- ✓ 이렇게 만들어진 Iteration 1 + Iteration 2 + Iteration 3이 우리의 GBM이 됨

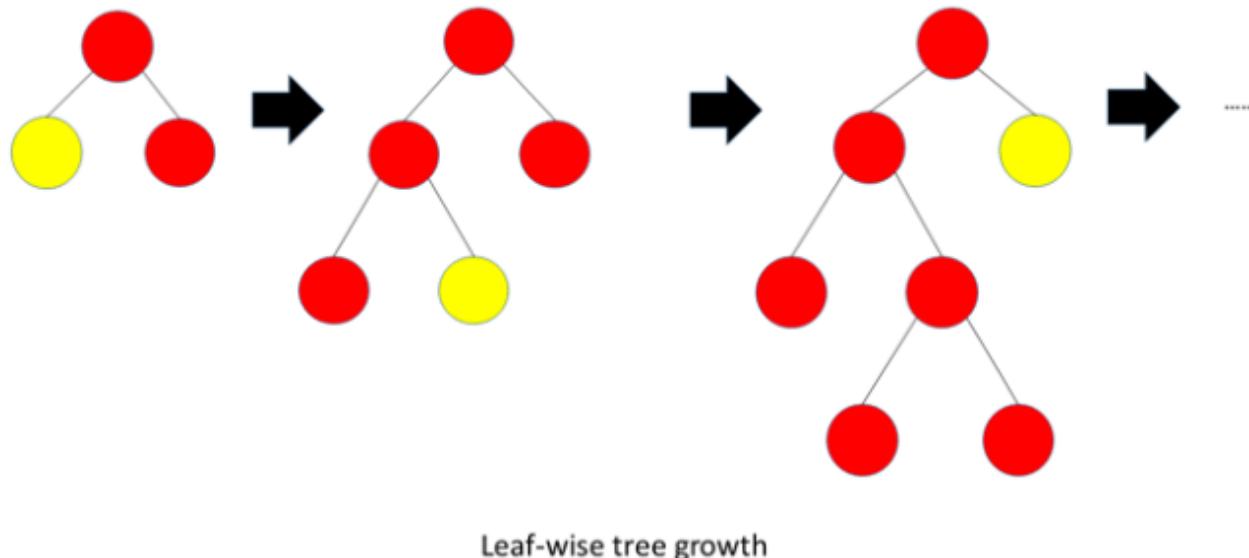


Packt

- ◆ 머신러닝 모형 : LightGBM(Light Gradient Boosting Machine)

- ◆ 기본원리

- ✓ 높은 성능 : LightGBM은 대용량 데이터셋에서도 빠른 학습과 예측 제공, 적은 메모리 사용
- ✓ Leaf-Wise 트리 분할 : 트리의 균형을 유지하지 않고 최대손실을 갖는 리프 노드 우선 분할하여 더 정확한 예측 가능



- ◆ 머신러닝 모형 : LightGBM(Light Gradient Boosting Machine)
- ◆ 하이퍼 파라미터 튜닝 주요 매개변수

주요 매개변수	설명
learning_rate	<ul style="list-style-type: none"> • 데이터 타입(기본값) : float(default = 0.1), • 각 부스팅의 반복 시, 분류기에 적용되는 가중치
n_estimators	<ul style="list-style-type: none"> • 데이터 타입(기본값) : int(default=100) • 설명 : LightGBM에서 부스팅 반복 횟수.
max_depth	<ul style="list-style-type: none"> • 데이터 타입(기본값) : int(default=-1) • 트리 모델의 최대 깊이를 제한. 이는 데이터가 작을 때 과적합을 처리하는 데 사용됨.
num_leaves	<ul style="list-style-type: none"> • 데이터 타입(기본값) : int(default=31) • 한 트리의 최대 잎 수
min_split_gain	<ul style="list-style-type: none"> • 데이터 타입(기본값) : double(default = 0.0) • 분할을 수행하기 위한 최소 이득 / 훈련 속도를 높이는데 사용
subsample	<ul style="list-style-type: none"> • 데이터 타입(기본값) : double(default = 1.0) • 리샘플링 없이 데이터의 일부를 무작위로 선택 • 모형학습 시간 및 과적합을 제어하는데 사용함
random_state	<ul style="list-style-type: none"> • 데이터 타입(기본값) : int(default=None) • 추정량의 임의성을 제어함

06. 시계열 데이터 예측 - LightGBM

강의 실습 영상 참고

06. 시계열 데이터 예측

(2023년 11월 기준)

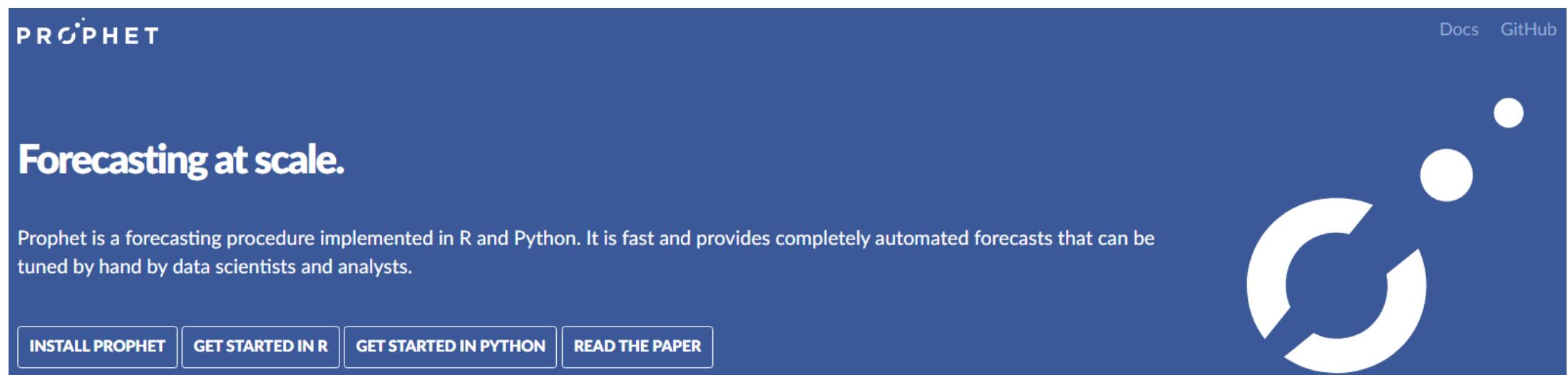
◆ Prophet

- ✓ Facebook에서 개발한 오픈 소스 시계열 예측 라이브러리
- ✓ 계절성을 갖는 비즈니스 시계열 데이터에 대한 예측 수행 시 강점을 가지고 있음
- ✓ 모델의 주요 구성 요소 : Trend, Seasonality, Holiday

주요 특징	설명
사용의 용이성	Prophet은 복잡한 시계열 모델링에 대한 전문 지식이 없는 사용자도 쉽게 고품질의 예측 설계 가능
계절성 인식	연간, 주간, 일간 계절성을 자동으로 감지하고 모델링
휴일 및 이벤트 처리	휴일과 특별 이벤트의 영향을 모델에 포함시킬 수 있음
추세 변화 포인트 감지	자동으로 시계열 데이터 내의 추세 변화 포인트를 감지하고 이를 모델에 반영
결과 해석 용이성	Prophet의 결과는 해석하기 쉽게 설계되어 있어, 비전문가도 결과를 이해하고 비즈니스 결정에 활용

◆ Prophet

- ✓ Facebook에서 개발한 오픈 소스 시계열 예측 라이브러리
- ✓ 계절성을 갖는 비즈니스 시계열 데이터에 대한 예측 수행 시 강점을 가지고 있음
- ✓ 모델의 주요 구성 요소 : Trend, Seasonality, Holiday
- ✓ 그 외 상세 내용 : <https://facebook.github.io/prophet/>



06. 시계열 데이터 예측 - LightGBM

강의 실습 영상 참고

07. 머신러닝 수행 시 고려사항

(2023년 11월 기준)

◆ 이상치의 의미

- ✓ 값이 크게 차이가 나는 경우

◆ 이상치 발생하는 주요 원인

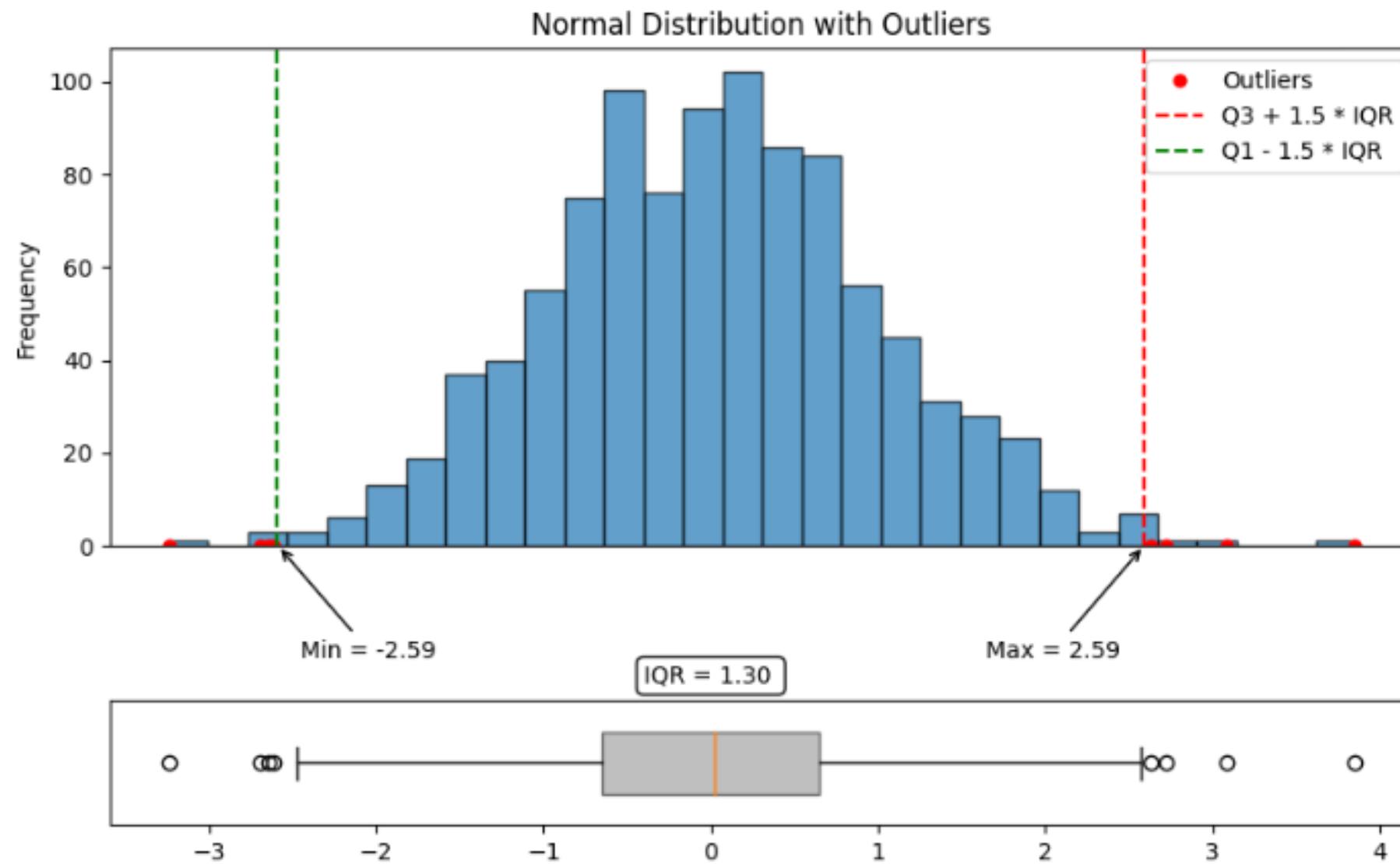
- ✓ 결측치로 표시할 때 (예: error, 9999)
- ✓ 자료 수집의 오류 시, (실내 온도 측정 예: 40000)
- ✓ 실제 관측치, (월급 예: 100만원, 1억원)

◆ 이상치 처리

- ✓ 실무에서 표준화된 정답은 없음
- ✓ 통계적인 방법 : IQR(Inter Quantile Range) 방식

07. 머신러닝 수행 시 고려사항

(2023년 11월 기준)

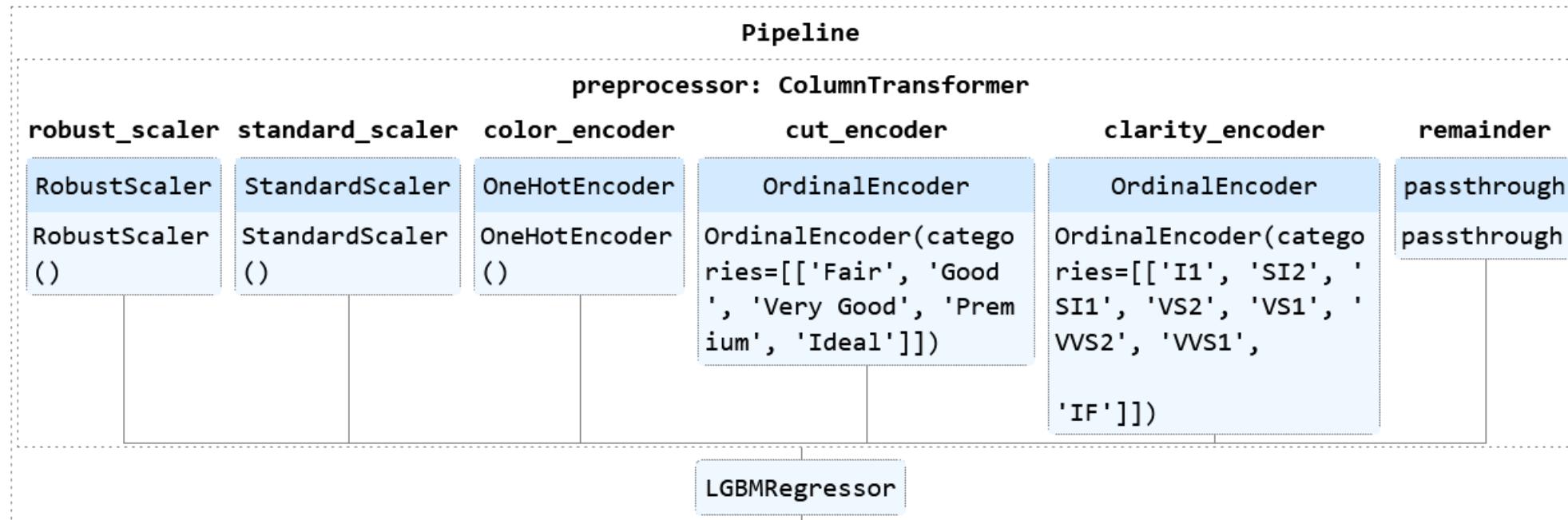


07. 머신러닝 시 고려사항

(2023년 11월 기준)

◆ 모형 자동화

- ✓ `sklearn.pipeline` 과 `sklearn.compose` 모듈을 적극 활용



파이썬 지리공간 데이터

with  python™

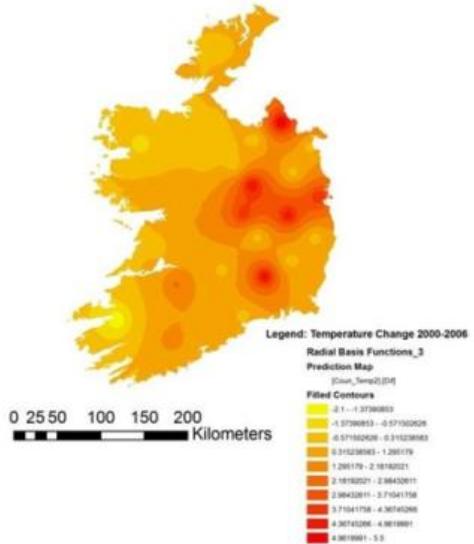
01. Geospatial Analysis

(2023년 11월 기준)

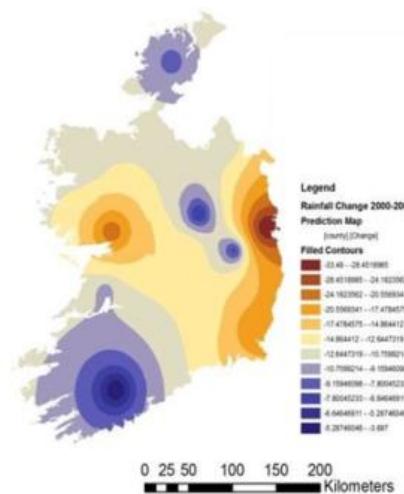
◆ 지리공간 분석의 개요

- ✓ 데이터의 지리적 패턴 또는 추세 파악
- ✓ 인구 밀도 또는 자원의 가용성 추정
- ✓ 질병의 확산 또는 자연재해 영향 모델링
- ✓ 비즈니스, 도시 계획, 공중 보건 등의 분야에서 의사 결정 지원

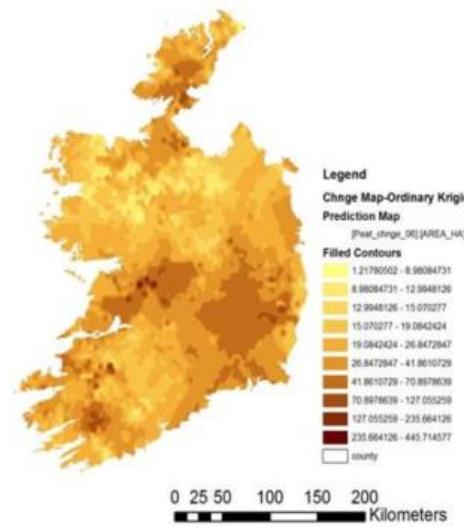
Temperature Change



Rainfall Change



Land-use Change

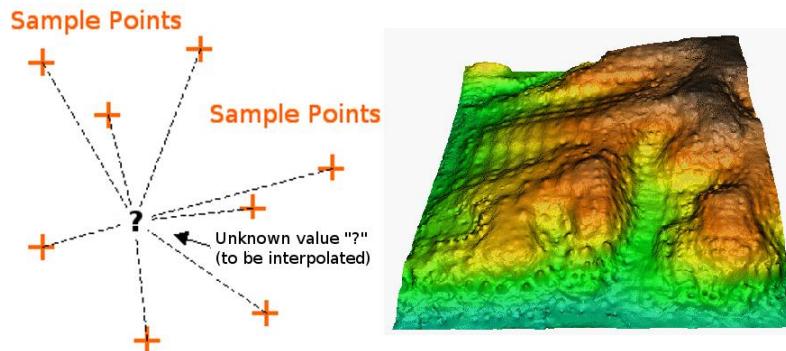


01. Geospatial Analysis

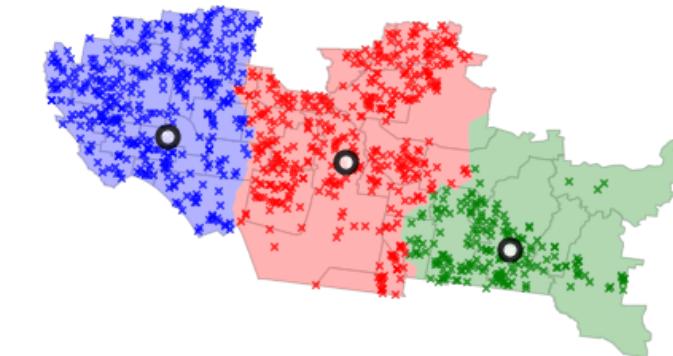
(2023년 11월 기준)

◆ 지리공간 분석 기법

주요 특징	설명
공간 보간	주변 데이터 포인트를 기반으로 특정 위치의 값을 추정
공간 클러스터링	공간에서 서로 가까운 데이터 포인트 그룹을 식별
공간 회귀	종속 변수와 하나 이상의 독립 변수 간의 관계를 모델링하는 동시에 공간적 의존성을 고려
네트워크 분석	도로망이나 공급망과 같이 연결된 위치의 네트워크를 따라 이동 또는 흐름을 모델링



공간 보간 예시



공간 클러스터링 예시

02. 주요 라이브러리

(2023년 11월 기준)

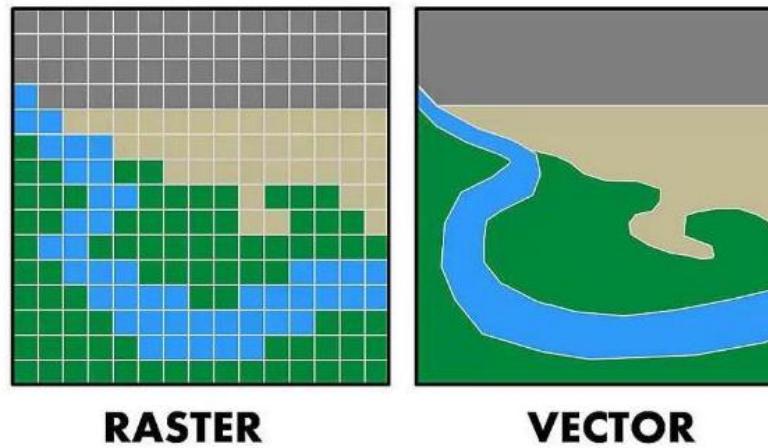
◆ GIS(Geographic Information System)을 다뤄야 하는 도구 필요

- ✓ 주요 솔루션 : ArcGIS, QGIS
- ✓ Python 주요 라이브러리

주요 특징	설명
GeoPandas	pandas 데이터프레임을 기반인 지리공간 데이터 작업을 위한 Python 라이브러리, 공간 작업과 분석 수행
Shapely	점, 선, 다각형과 같은 2D 기하 도형 개체로 작업하고 조작하기 위한 Python 라이브러리
Fiona	Shapefile가 GeoJSON과 같은 지리공간 데이터 형식을 읽고 쓰기 위한 라이브러리
PyProj	지리 좌표계와 투영 좌표계간 변환 위한 Python 라이브러리
GDAL/OGR	Raster 및 Vector 데이터를 포함한 데이터를 포함한 지리공간 데이터 형식으로 작업할 수 있는 강력한 라이브러리

◆ Raster Vs Vector

Raster	Vector
<ul style="list-style-type: none">✓ 셀 or 픽셀 그리드로 표현✓ 각 셀에는 색상 등 특정 속성 포함✓ 연속적인 데이터에 가장 적합✓ 지도, 위성 이미지, 항공사진 시각화에 적합✓ 파일명 : .svg, .shp	<ul style="list-style-type: none">✓ 점, 선, 다각형으로 표현<ul style="list-style-type: none">• 위치, 크기 모양과 같은 속성을 의미• 모든 점은 x, y 좌표(coordinate) 표현✓ 선은 도로나 강을 표현✓ 다각형은 빌딩의 경계, 호수 등을 표현✓ shapefiles 형태로 저장✓ 파일명 : .jpg, .png, .tif



◆ Shapefile

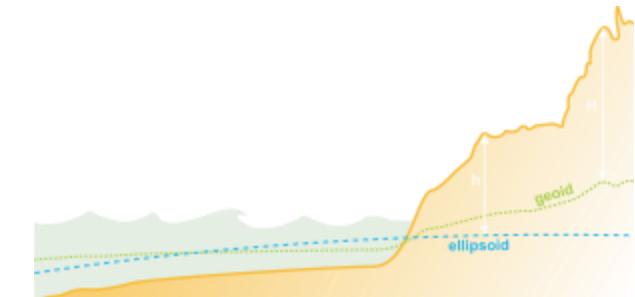
- ✓ 지리 정보 시스템(GIS) 소프트웨어에서 널리 사용되는 지리 공간 벡터 데이터 형식
- ✓ 확장자가 다른 세가지 파일로 구성 (.shp, .shx, .dbf)
 - 공간데이터 (Data) : .shp, .shx
 - 속성정보 (Information) : .dbf

주요 특징	설명
.shp	점, 선, 다각형과 같은 벡터 피처의 geometry가 포함
.shx	GIS 소프트웨어가 .shp 파일에 빠르게 검색 및 액세스할 수 있도록 하는 인덱스 파일
.dbf	파일에는 각 벡터 피처와 관련된 속성 데이터, 예) 인구, 면적 등의 정보가 담긴 데이터가 파일에 저장됨

◆ CRS(Coordinate Reference System)

- ✓ 위도와 경도의 조합을 나타내는 좌표체계 의미
- ✓ 좌표 참조 시스템 유형

GCS(Geographic Coordinate Systems)	PCS(Projected Coordinate Systems)	VCS(Vertical Coordinate Systems)
<ul style="list-style-type: none">✓ 위도 및 경도 좌표 사용✓ 지구 표면의 지형지물의 위치 정의✓ WGS84 or NAD83 등 다양한 GCS 존재	<ul style="list-style-type: none">✓ 투영 좌표계✓ 지도나 컴퓨터 화면과 같은 평평한 표면에 지리적 특징 표현	<ul style="list-style-type: none">✓ 해수면과 같은 기준 표면✓ 고도 높이 등 측정 시 사용✓ 홍수 모델링, 항공 애플리케이션 사용

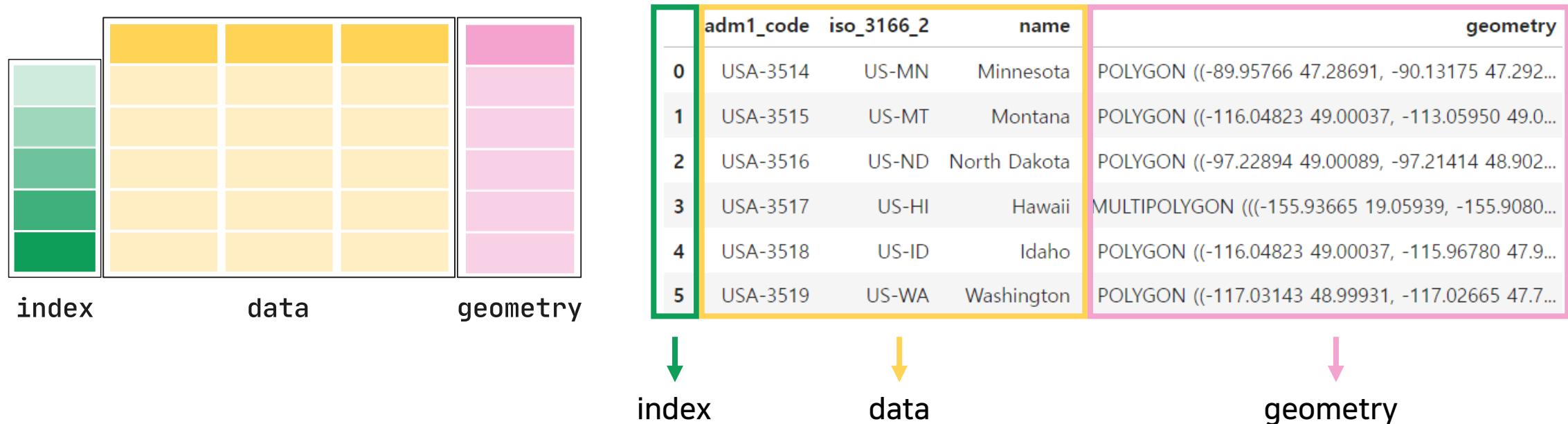


04. GeoPandas

(2023년 11월 기준)

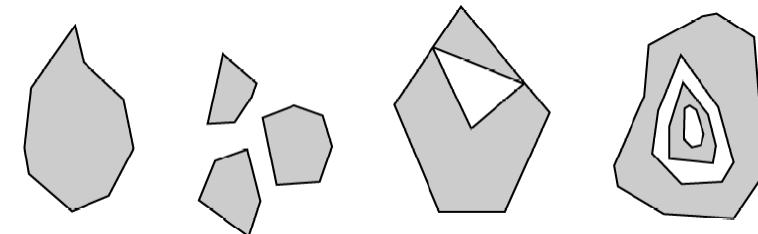
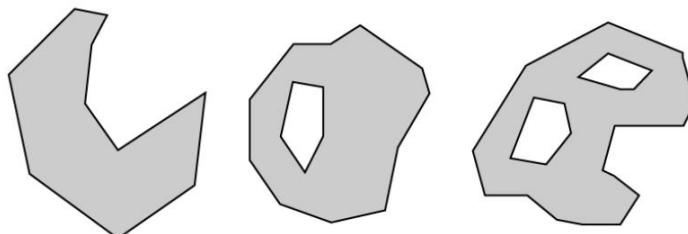
◆ Structure

- ✓ 기존 pandas DataFrame의 index + data에서 확장한 것으로 geometry라는 특별한 열을 지님.
- ✓ 추가된 컬럼 geometry는 shapely geometry 객체를 담고 있음.



◆ Polygon vs MultiPolygon

Polygon	MultiPolygon
<ul style="list-style-type: none">✓ 다각형은 1개의 외부 경계와 0개 이상의 내부 경계<ul style="list-style-type: none">• 평면형 지표(Surface)✓ 각 내부 경계는 다각형의 Hole을 정의✓ X, Y 좌표 집합으로 정의된 2차원 또는 3차원 도형✓ 예: 하나의 섬, 호수, 하나의 행정 구역 정의	<ul style="list-style-type: none">✓ 복수의 Polygon이 존재하는 형태✓ 각 Polygon은 겹쳐지거나 분리 또는 중첩될 수 있음✓ 예: 하와이 같은 군도



Map Visualization

강의 실습 영상 참고

Streamlit

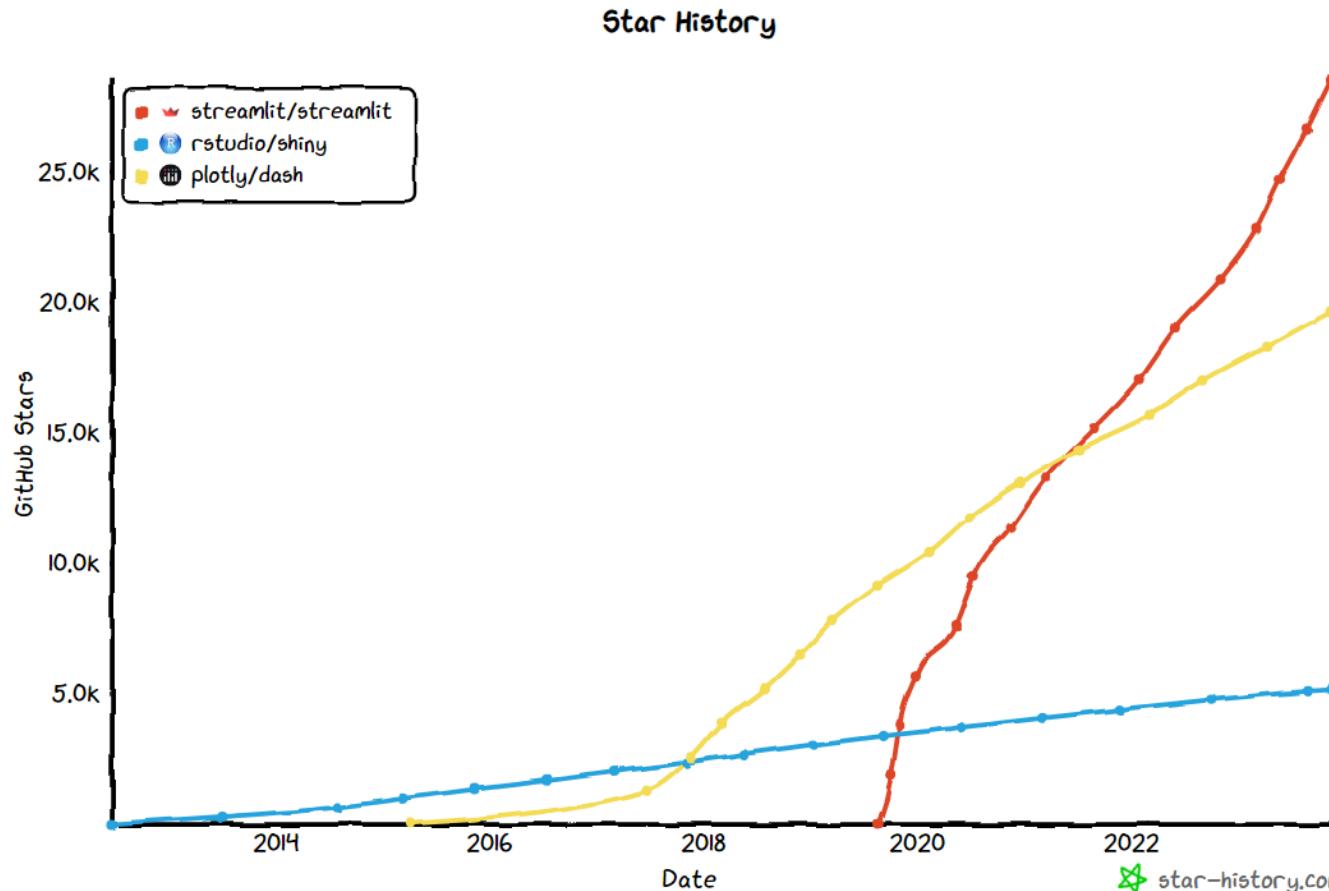
with  python™

01. Popularity

(2023년 11월 기준)

◆ Release

- ✓ 2019년 10월, 가장 빠르게 성장하는 Dashboard 프레임워크



02. Streamlit 탄생 스토리

(2023년 11월 기준)

◆ Streamlit 처음 설계한 창업자



- **CEO of Streamlit**
- **2022년 3월, Snowflake와 합병**
- **Computer Science Prof. at Carnegie Mellon**
- **Google X Project, VP at Zoox**

03. Streamlit 탄생 스토리

(2023년 11월 기준)

- ◆ Streamlit 처음 설계한 창업자



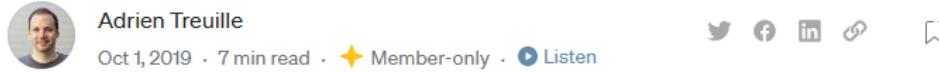
*What if we could make building tools
as easy as writing Python scripts?*

“Adrien Treuille”

03. Streamlit 탄생 스토리

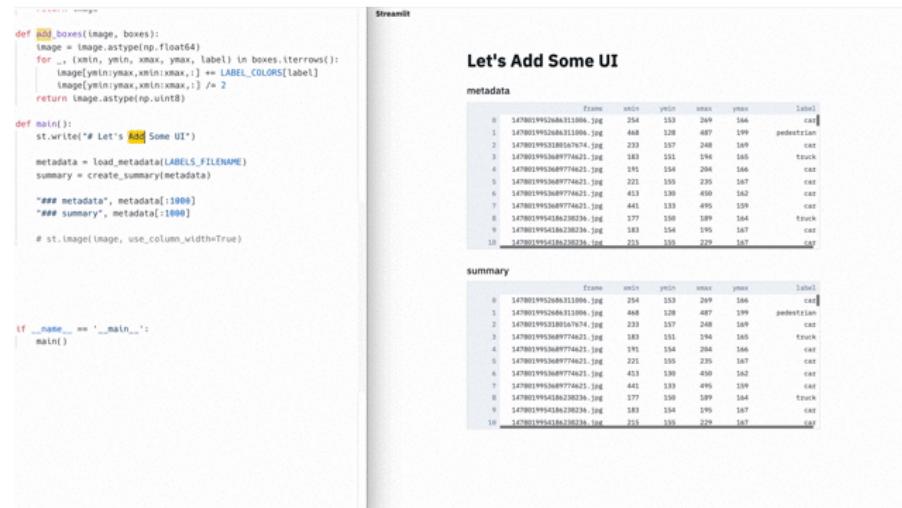
(2023년 11월 기준)

Adrien Treuille
Oct 1, 2019 · 7 min read · Member-only · Listen



Turn Python Scripts into Beautiful ML Tools

Introducing Streamlit, an app framework built for ML engineers



Coding a semantic search engine with real-time neural-net inference in 300 lines of Python.

```
def add_boxes(image, boxes):
    image = image.astype(np.float64)
    for _, (xmin, ymin, xmax, ymax, label) in boxes.iterrows():
        image[ymin:ymax, xmin:xmax, :] += LABEL_COLORS[label]
    image[ymin:ymax, xmin:xmax, :] /= 2
    return image.astype(np.uint8)

def main():
    st.write("# Let's Add Some UI")

    metadata = load_metadata(LABELS_FILENAME)
    summary = create_summary(metadata)

    """ metadata", metadata[:1000]
    """ summary", metadata[1000]

    # st.image(image, use_column_width=True)

if __name__ == '__main__':
    main()
```

The Streamlit UI shows a table titled "Let's Add Some UI" with columns: frame, xmin, ymin, xmax, ymax, and label. The table contains 10 rows of data, mostly labeled "car".

frame	xmin	ymin	xmax	ymax	label
0	254	133	269	166	car
1	468	128	487	199	pedestrian
2	233	137	248	169	car
3	183	131	196	165	truck
4	191	130	206	166	car
5	225	139	235	167	car
6	413	130	495	162	car
7	441	133	495	159	car
8	177	150	189	164	truck
9	183	154	195	167	car
10	215	135	229	167	car

In my experience, every nontrivial machine learning project is eventually stitched together with bug-ridden and unmaintainable internal tools. These tools — often a patchwork of Jupyter Notebooks and Flask apps — are difficult to deploy, require reasoning about client-server architecture, and don't integrate well with machine learning constructs like Tensorflow GPU sessions.



Adrien Treuille

2.1K Followers

Adrien is co-founder of Streamlit, the ML tooling framework. Adrien was a computer science prof at Carnegie Mellon, lead a Google X project, and was VP at Zoox.

Follow



Release

• 2019년 10월 1일

More from Medium

Dennis Nig... in Python in Plain English...



Creating an Awesome Web App With Python and Streamlit

Frank Andra... in Towards Data Scie...



Predicting The FIFA World Cup 2022 With a Simple Model using Python

Moez Ali



Top AutoML Python libraries in 2022

Yang Zhou in TechToFreedom



9 Fabulous Python Tricks That Make Your Code More Elegant

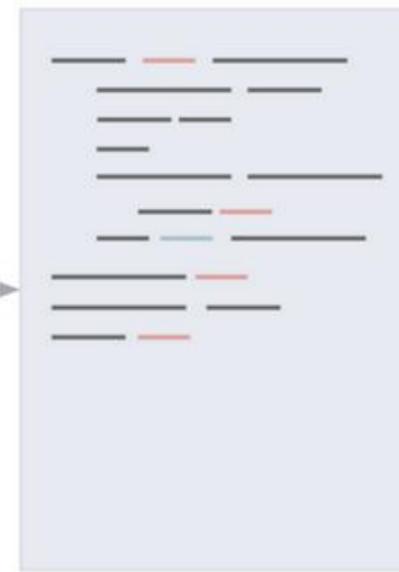
03. Streamlit 탄생 스토리

(2023년 11월 기준)

◆ 머신러닝 엔지니어의 가장 큰 문제점 (2019년 이전)



Step 1:
Explore in a Jupyter
notebook.



Step 2:
Copy-paste into a
Python script.

The unmaintainability trap



Step 3:
Write Flask app. Think
about HTTP requests,
HTML, callbacks, JS...



Step 4:
Uh oh. Need more
features. 😞

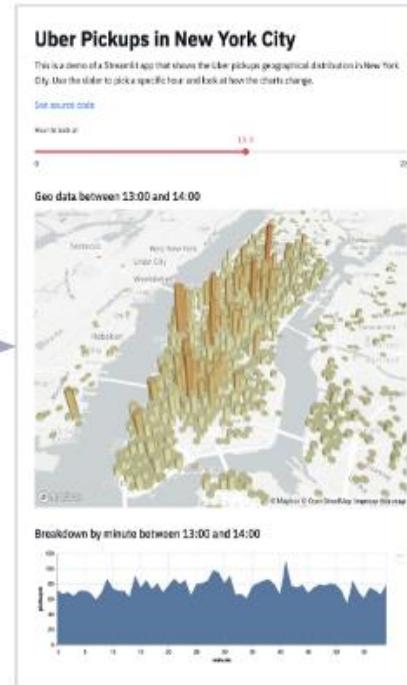
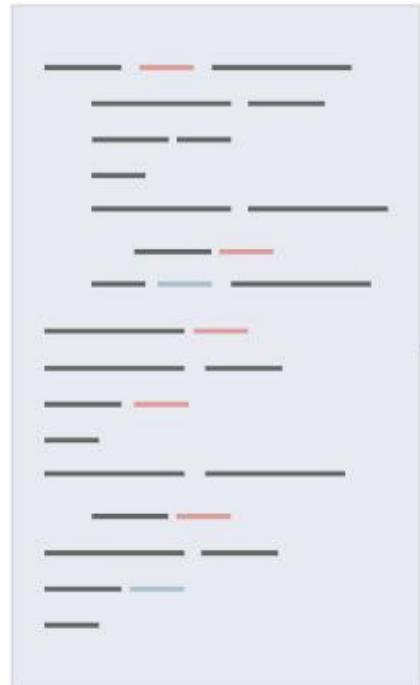
< The machine learning engineers' ad-hoc app building flow >

03. Streamlit 탄생 스토리

(2023년 11월 기준)

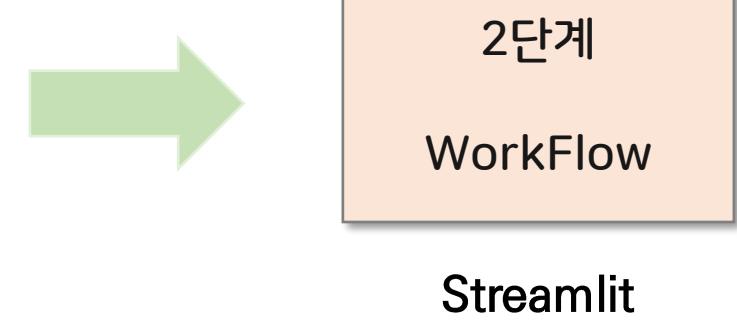
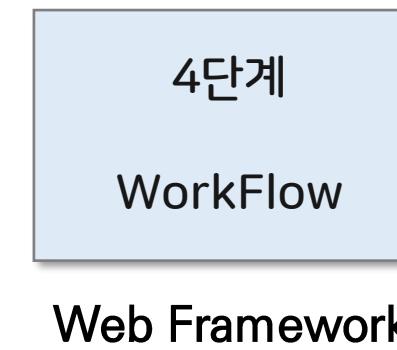
◆ Streamlit WorkFlow

The Streamlit workflow



Step 1:
Sprinkle in a few API calls into your existing Python script.

Step 2:
Show off your beautiful, performant tool! 🎉



◆ Scripting Workflow

- ✓ Coding just like on Google Colab or Jupyter Notebook For Data Analyst

기본 원칙	코드 예시
Embrace Python Scripting	<pre>>>> import streamlit as st >>> st.write('Hello, World!')</pre>
Treat widgets as variables	<pre>>>> import streamlit as st >>> x = st.slider('x') >>> st.write(x, 'squared is', x * x)</pre>
Reuse Data and Computation Key : Cache (Persistent, Immutable by default)	<pre>>>> import streamlit as st >>> import pandas as pd >>> data = pd.read_csv('iris.csv') >>> st.dataframe(data)</pre>

◆ Advanced Features Cache

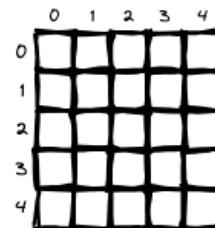
- ✓ Just call long-running functions once and save it into session

st.cache_data

anything you CAN store in a database



Python
primitives



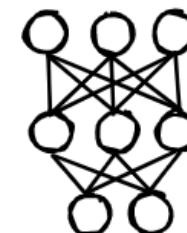
dataframes



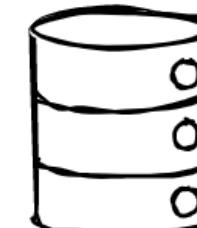
API calls

st.cache_resource

anything you CAN'T store in a database



ML models



database
connections

◆ Advanced Features Cache

- ✓ 동일한 연산을 여러 번 수행하지 않아도 사용자 상호 작용이 빨라지고 웹 성능이 향상됨
- ✓ `st.cache_data` vs `st.cache_resource`

<code>@st.cache_data</code>	<code>@st.cache_resource</code>
<ul style="list-style-type: none">✓ CSV 파일 불러오기✓ API 호출✓ NumPy 배열 변환✓ str, DataFrame, List 등이 함수 반환값	<ul style="list-style-type: none">✓ ML Models or Database Connections

a serializable data object

unserializable objects

05. Cache Data

(2023년 11월 기준)

◆ Persisting across Running App

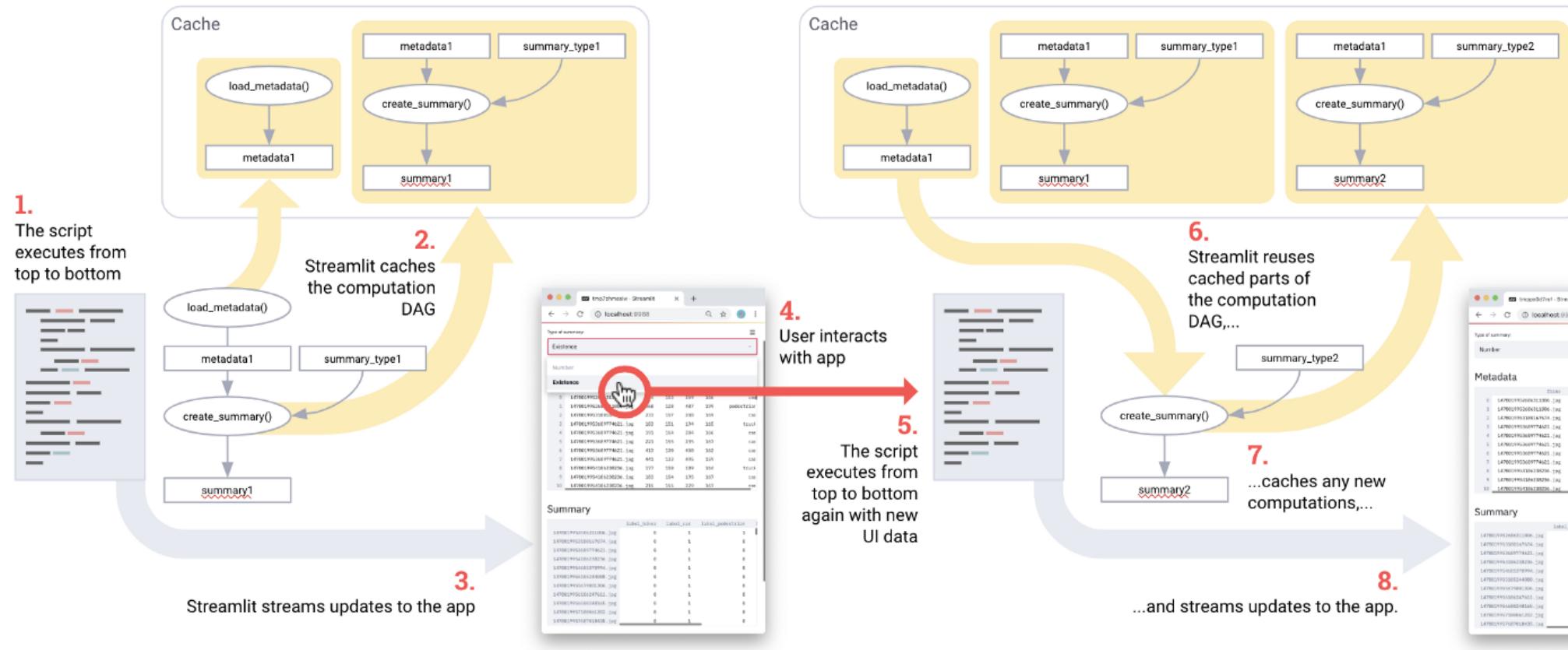


05. Cache Data

(2023년 11월 기준)

◆ Persisting across Running App

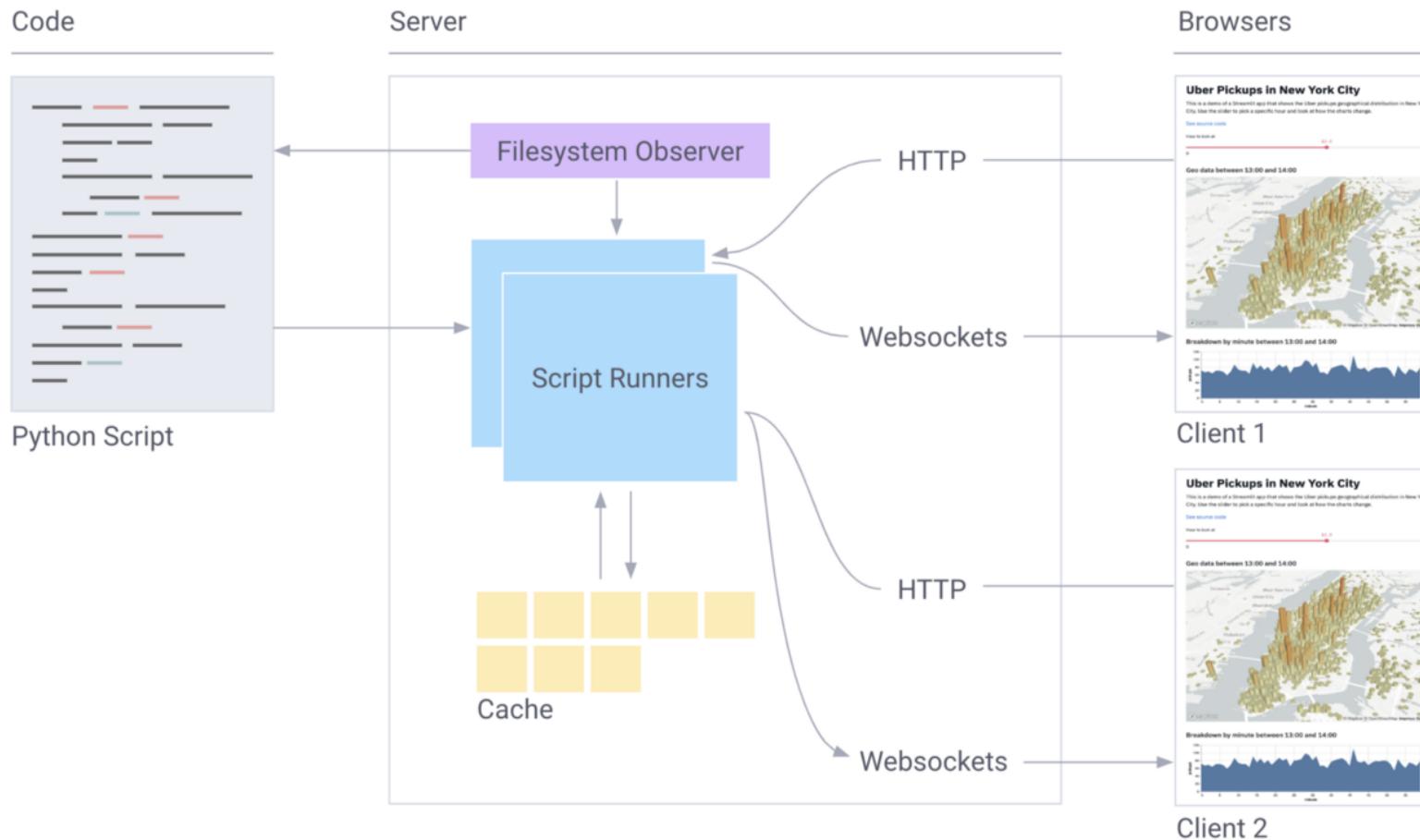
✓ UI가 변경될 때만, recomputing이 발생함



06. Block Diagram of Streamlit's Components

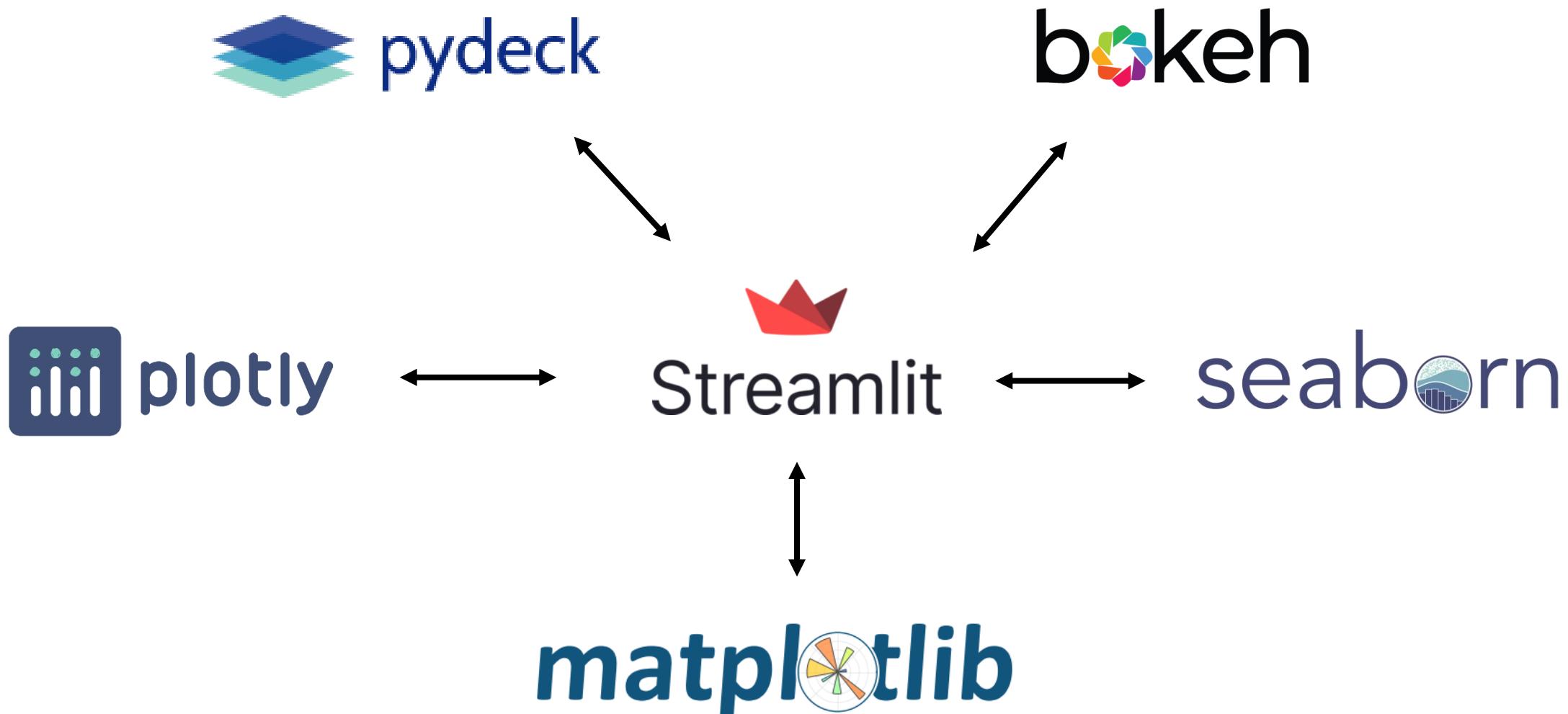
(2023년 11월 기준)

- ◆ Code → Server → Browsers
- ✓ Python Script → Cache → Client



07. Connect to Visualizations

(2023년 11월 기준)



◆ 설정 시스템을 사용하면 Streamlit Web 동작을 사용자 지정할 수 있음

- ✓ Streamlit Configuration File은 '~/.strealmit/config.toml' 경로에 지정(Global Settings)
- ✓ Environment Variables
 - `STREAMLIT_`으로 지정한다.
- ✓ 주요 sections and options

sections	설명
[server]	서버의 포트를 구성하고, Web 폴더의 기본 경로, CookieSecret 설정, CORS(Cross-Origin Resource Sharing) 설정
[theme]	웹의 기본 색상, 배경색 등의 옵션을 사용하여 Web의 형태 사용자 지정 가능
[browser]	실행할 기본 브라우저와 실행 동작 설정, IP address와 DNS name 설정 가능
[logger]	Logging 수준과, 로그 메시지의 대상을 정의
[runner]	경고 비활성화 또는 최대 메시지 크기 설정 등, Streamlit 스크립트가 실행하는 방식 조정

09. Interactive Widgets

(2023년 11월 기준)

◆ 기본 튜토리얼 확인 : <https://docs.streamlit.io/library/cheatsheet>

Home / Streamlit library / Cheat sheet

Cheat Sheet

This is a summary of the docs, as of [Streamlit v1.28.0](#).

Install & Import

```
streamlit run first_app.py
```

```
# Import convention
```

```
>>> import streamlit as st
```

Command line

```
streamlit --help
```

```
streamlit run your_script.py
```

```
streamlit hello
```

```
streamlit config show
```

```
streamlit cache clear
```

```
streamlit docs
```

```
streamlit --version
```

Pre-release features

```
pip uninstall streamlit
```

```
pip install streamlit-nightly
```

[Learn more about experimental features](#)

Magic commands

```
# Magic commands implicitly
```

```
# call st.write().
```

```
'_This_ is some **Markdown***'
```

```
my_variable
```

```
'dataframe:', my_data_frame
```

Control flow

```
# Stop execution immediately:
```

```
st.stop()
```

```
# Rerun script immediately:
```

```
st.rerun()
```

Connect to data sources

```
st.connection("pets_db", type='
```

```
conn = st.connection("sql")
```

```
conn = st.connection("snowflake")
```

```
>>> class MyConnection(BaseConn
```

```
>>> def _connect(self, **kwarg
```

```
>>>     return myconn.connect(
```

Display text

```
st.text('Fixed width text')
```

```
st.markdown('_Markdown_') # set
```

```
>>>     username = st.text_input('U
```

```
>>>     password = st.text_input('P
```

```
>>>     return self._instance
```

```
>>>     st.form_submit_button('Lo
```



Streamlit

강의 실습 영상 참고

작업형 제3유형

with  python™

01. 가설 검정 주요 용어

(2023년 11월 기준)

◆ 가설 (Hypothesis)

- ✓ 모집단의 특성, 특히 모수에 대한 가정 혹은 잠정적인 결론이다.

종류	설명
귀무가설 (H_0 , Null Hypothesis)	<ul style="list-style-type: none">기존과 비교하여 변화 혹은 차이가 없음을 나타내는 가설검정 방법에 따라 귀무가설의 내용이 달라짐
대립가설 (H_1 , Alternative Hypothesis)	<ul style="list-style-type: none">표본을 통해 확실한 근거를 가지고 입증하고자 하는 가설연구가설 (Research Hypothesis) 이라고도 함

◆ 가설 검정 시, 관심 있는 가설은 대립가설이며, 대립가설이 참이라는 확실한 근거가 없다면 귀무가설 채택

- ✓ 표현문구는 귀무가설 채택 (X), **귀무가설을 기각하지 못**한다 (O)
- ✓ 대립가설이 참이라는 확실한 근거 발견 시, **귀무가설을 기각**한다(O) 표현

01. 가설 검정 주요 용어

(2023년 11월 기준)

◆ 가설 검정

- ✓ 단계 1. 모집단에 대해 어떤 가설 설정
- ✓ 단계 2. 모집단으로부터 추출된 표본 분석
- ✓ 단계 3. 연구 가설이 틀린지 맞는지 타당성 여부를 검정하는 통계적 기법

◆ 검정통계량

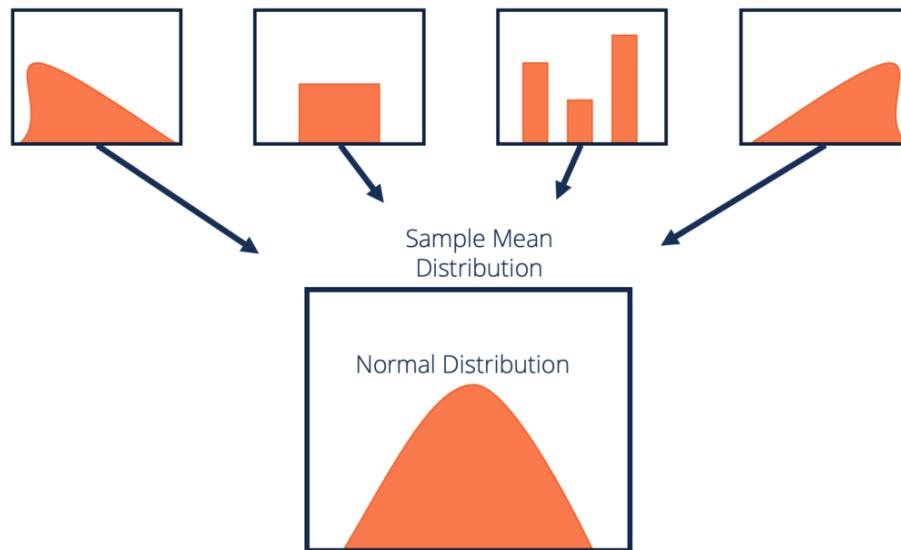
- ✓ 모수 추론하기 위해 필요한 표본 통계량
- ✓ 귀무가설이 참이라는 전제하에 추출된 확률표본의 정보를 이용하여 계산함
- ✓ 중심극한정리 이론에 근거하기 때문에, 적은 표본으로도 전체 모수를 추정할 수 있음

01. 가설 검정 주요 용어

(2023년 11월 기준)

◆ 중심 극한 정리(CLT : Central Limit Theorem)

- ✓ 정의 : 정규분포가 아닌 분포라도 선택된 표본들의 평균을 모아 다시 분포를 그리면 정규분포가 됨
- ✓ 단, 최소 표본의 크기가 작으면 안됨 ($n \geq 30$)
- ✓ CLT에 따르면, Sampling Distribution은 정규분포를 따름
- ✓ 모든 모집단 데이터의 평균 = 모든 샘플 평균들의 평균 = Sampling Distribution의 평균



출처 : <https://corporatefinanceinstitute.com/resources/data-science/central-limit-theorem/>

01. 가설 검정 주요 용어

(2023년 11월 기준)

◆ 가설 검정 오류

- ✓ 다음과 같은 통계적 오류가 발생할 가능성이 항상 존재함

	귀무가설(H_0)이 사실이라고 판정 (무죄)	귀무가설(H_0)이 거짓이라고 판정 (유죄)
귀무가설(H_0)이 사실	<p>옳은 결정 No Error ($1-\alpha$) 무죄를 무죄라 함</p>	<p>제1종 오류 무죄를 유죄라 함</p>
귀무가설(H_0)이 거짓	<p>제2종 오류 유죄를 무죄라 함</p>	<p>옳은 결정 No Error ($1-\beta$) 유죄를 유죄라 함</p>

01. 가설 검정 주요 용어

(2023년 11월 기준)

◆ 가설 검정 오류

- ✓ 다음과 같은 통계적 오류가 발생할 가능성이 항상 존재함

종류	설명
제 1종 오류 (Type I Error)	<ul style="list-style-type: none">귀무가설이 참이나 귀무가설을 기각하는 오류유의수준(Level of Significance)<ul style="list-style-type: none">- 제 1종 오류를 범할 최대 허용확률- α로 표기하며, 일반적으로 α값을 0.01, 0.05 또는 0.1 등의 값으로 설정신뢰수준(Level of Confidence)<ul style="list-style-type: none">- 귀무가설이 참일 때 이를 참이라고 판단하는 확률- $1 - \alpha$로 표기함
제 2종 오류 (Type II Error)	<ul style="list-style-type: none">귀무가설이 참이 아닌데, 귀무가설을 채택하는 경우베타수준(β)<ul style="list-style-type: none">- 제2종 오류를 범할 최대 허용확률검정력<ul style="list-style-type: none">- 귀무가설이 참이 아닌 경우 이를 기각할 수 있는 확률- $1 - \beta$로 표기함

01. 가설 검정 주요 용어

(2023년 11월 기준)

◆ 의사 결정의 원칙

- ✓ 귀무가설이 옳다는 가정하에 H_0 이 옳다는 증거를 제시해야 함 (Presumed Innocence)

◆ 의사 결정의 기준

- ✓ 가장 주의해야 하는 것은 1종 오류를 범하는 것
 - 나의 주장이 옳지 않은데 옳다고 결론 내리는 것
- ✓ 의사결정의 기준은 1종 오류를 범할 확률(유의 수준 α)

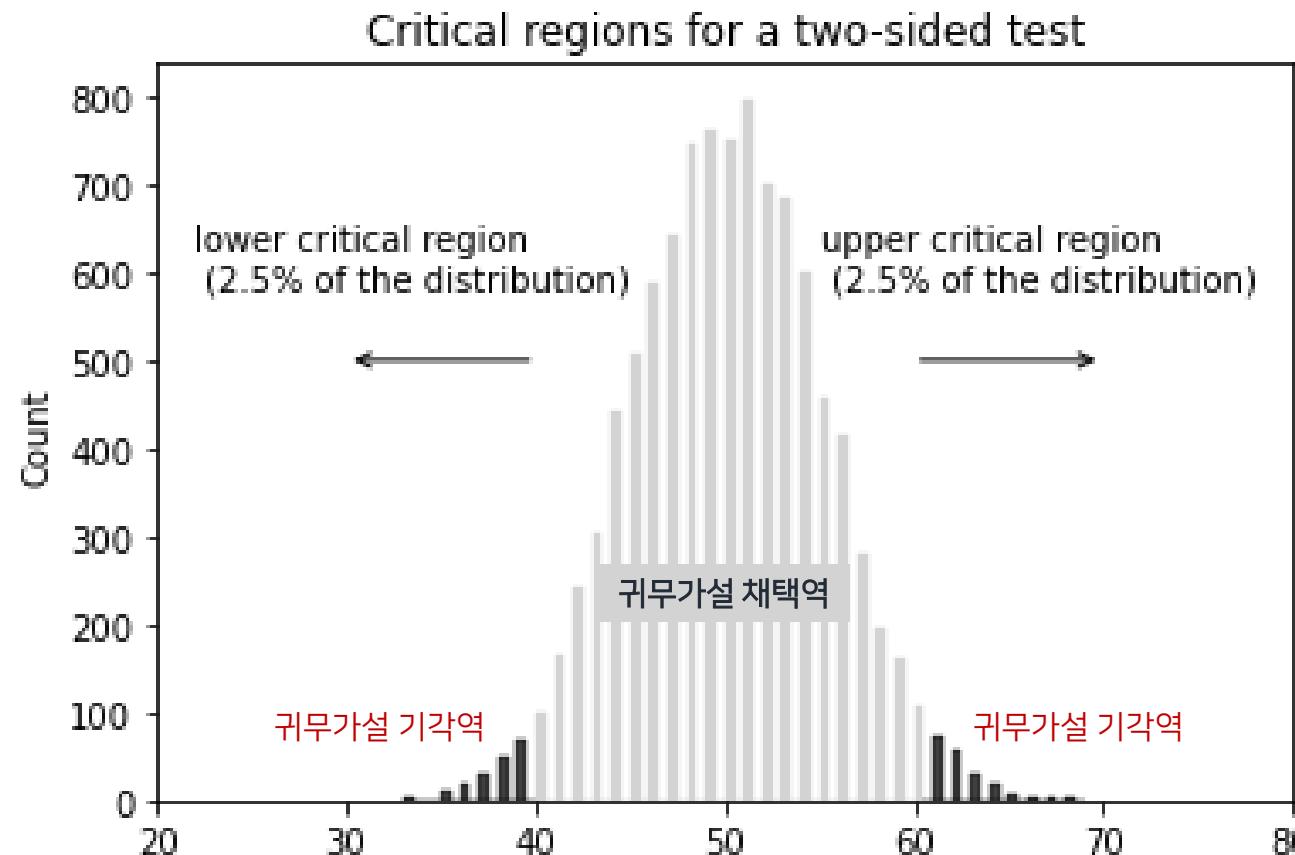
	의사결정	설명
$\alpha = 0$	H_0	무죄인 사람을 유죄라고 판단 할 오류 0%
$\alpha = 0.01 (1\%)$	$H_0 : 99\%$ $H_1 : 1\%$	매우 보수적이며 엄격한 판단 표본에서 구한 통계량 값이 나올 확률이 1%보다 작을 때 의미
$\alpha = 0.05 (5\%)$	$H_0 : 95\%$ $H_1 : 5\%$	오류가 5%까지 허용 상대적으로 덜 엄격한 의사결정

01. 가설 검정 주요 용어

(2023년 11월 기준)

◆ p-value와 임계값

- ✓ 귀무가설 H_0 가 참일 때 표본에서 얻어진 결과가 귀무가설을 기각하게 하는 확률
- ✓ 주어진 유의수준 α 하에서 귀무가설 H_0 의 채택 또는 기각 여부를 판정하여 주는 기준이 되는 값



01. 가설 검정 주요 용어

(2023년 11월 기준)

◆ p-value와 임계값

- ✓ 귀무가설 H_0 가 참일 때 표본에서 얻어진 결과가 귀무가설을 기각하게 하는 확률
- ✓ 주어진 유의수준 α 하에서 귀무가설 H_0 의 채택 또는 기각 여부를 판정하여 주는 기준이 되는 값

p-value	Sig. stars	해석	귀무가설
$p > 0.05$		통계적으로 유의하지 않음	채택(Retained)
$p < 0.05$	*	통계적으로 유의함($\alpha = 0.05$)	기각(Rejected)
$p < 0.01$	**	통계적으로 유의함($\alpha = 0.01$)	기각(Rejected)
$p < 0.001$	***	통계적으로 유의함($\alpha = 0.001$)	기각(Rejected)

02. 가설 검정 기법 - 단일표본 T-검정

(2023년 11월 기준)

◆ 가설 설정

- ✓ 모평균 μ 에 대한 검정 절차로 다음과 같이 귀무가설과 대립가설 설정

가설	표현식	설명
H_0	$\mu = \mu_0$	모평균과 표본평균은 같다
H_1	$\mu \neq \mu_0$	모평균과 표본평균은 같지 않다.

◆ 검정통계량

- ✓ 모분산을 알고 있을 때,

$$Z = \frac{\bar{X} - \mu}{\sigma / \sqrt{n}}$$

\bar{X} : 표본평균

σ : 모분산

n : 표본의 갯수

02. 가설 검정 기법 - 단일표본 T-검정

(2023년 11월 기준)

◆ 주요 메서드 - 실습

- ✓ 모평균의 값을 알고 있을 때

```
1 : from scipy import stats
2 : t_score, p_value = stats.ttest_1samp(df['Height'], popmean=75)
3 : print(round(t_score, 2), round(p_value, 2))
    0.87, 0.39
```

◆ 메서드 설명

- ✓ 독립 관측치 샘플 array의 예상값(평균)이 주어진 모집단 평균인 popmean과 같다는 귀무 가설에 대한 검정.

메서드	매개변수	설명
ttest_1samp(array, popmean)	array	Sample Observation
	popmean	Expected value in null hypothesis

03. 가설 검정 기법 - 독립표본 T-검정

(2023년 11월 기준)

◆ 가설 설정

- ✓ 두개의 독립 표본 X, Y 가 각각 $N(\mu_1, \sigma_1^2), N(\mu_2, \sigma_2^2)$ 를 따를 때, 두 모집단의 평균 차이 $\mu_1 - \mu_2$ 의 검정은 다음과 같음.

가설	표현식	설명
H_0	$\mu_1 = \mu_2$	그룹 1의 평균과 그룹 2의 평균은 같다.
H_1	$\mu_1 \neq \mu_2$	그룹 1의 평균과 그룹 2의 평균은 같지 않다.

◆ 가정(Assumptions) 확인

- ✓ 독립성
- ✓ Normality 정규성
- ✓ Homogeneity of Variance 분산의 동질성

03. 가설 검정 기법 - 독립표본 T-검정

(2023년 11월 기준)

◆ 정규성 검정

- ✓ 각 그룹의 표본수가 $N \leq 30$ 이하일 때, 검정해야 함
- ✓ 각 그룹의 표본수가 모두 $N \geq 30$ 이상일 때, 중심극한정리에 의해 정규성 가정을 만족했다고 봄

가설	설명
H_0	각 자료는 정규분포를 따른다.
H_1	각 자료는 정규분포를 따르지 않는다.

◆ 확인 방법

- ✓ Shapiro-Wilk tests

◆ 가정 위반 시, Mann-Whitney Test를 진행

03. 가설 검정 기법 - 독립표본 T-검정

(2023년 11월 기준)

◆ 등분산성 검정

- ✓ 각 그룹의 데이터수가 다르면 분산이 같은지 검정
- ✓ 만약에 각 그룹의 데이터수가 동일하면, 분산이 같다고 가정함

가설	설명
H_0	두 그룹의 분산은 차이가 없음
H_1	두 그룹의 분산은 차이가 있음

◆ 확인 방법

- ✓ bartlett, fligner, levene 검정 주로 사용
- ✓ 일반적으로는 levene 검정을 주로 사용

◆ 가정 위반 시

- ✓ Welch Test 사용

03. 가설 검정 기법 - 독립표본 T-검정

(2023년 11월 기준)

◆ 가설 설정

- ✓ 두개의 독립 표본 X, Y 가 각각 $N(\mu_1, \sigma_1^2), N(\mu_2, \sigma_2^2)$ 를 따를 때, 두 모집단의 평균 차이 $\mu_1 - \mu_2$ 의 검정은 다음과 같음.

가설	표현식	설명
H_0	$\mu_1 = \mu_2$	그룹 1의 평균과 그룹 2의 평균은 같다.
H_1	$\mu_1 \neq \mu_2$	그룹 1의 평균과 그룹 2의 평균은 같지 않다.

◆ 검정통계량

- ✓ 그룹 1의 표본평균 \bar{X} , 그룹 1의 표본평균 \bar{Y} 의 차이에 근거하여 구성
- ✓ 검정통계량 T 는 자유도 $n + m - 2$ 인 t -분포를 따른다.

$$T = \frac{\bar{X} - \bar{Y}}{S_p \sqrt{\frac{1}{n} + \frac{1}{m}}}, S_p^2 = \frac{(n-1)s_1^2 + (m-1)s_2^2}{n+m-2}$$

S_p^2 : 공통분산 σ^2 의 합동 표본분산
 $n - 1$: 그룹 1 표본의 자유도
 $m - 1$: 그룹 2 표본의 자유도

03. 가설 검정 기법 - 독립표본 T-검정

(2023년 11월 기준)

◆ 주요 메서드 - 등분산성 검정 실습

- ✓ 데이터의 각 변수들은 정규분포를 만족하며, 두 그룹은 등분산을 만족한가?

```
1 : from scipy import stats  
2 : stats.levene(df.loc[df['supp'] == "VC", 'len'], df.loc[df['supp'] == "OJ", 'len'])  
LeveneResult(statistic=1.2135720656945064, pvalue=0.2751764616144053)
```

◆ 메서드 설명

- ✓ Levene 검정은 모든 입력 샘플이 동일한 분산을 가진 모집단에서 나온 것이라는 귀무가설을 테스트 함

메서드	매개변수	설명
levene(sample1, sample2, ...)	sample1 : array_like	The sample data, possibly with different lengths. Only one-dimensional samples are accepted.

03. 가설 검정 기법 - 독립표본 T-검정

(2023년 11월 기준)

◆ 주요 메서드 - 독립표본 t-검정

- ✓ 데이터의 각 변수들은 정규분포를 만족하며, 두 그룹은 등분산을 만족한다!

```
1 : from scipy import stats  
2 : t_score, p_value = stats.ttest_ind(df.loc[df['supp'] == "VC", 'len'],  
                                         df.loc[df['supp'] == "OJ", 'len'], equal_var = True)  
3 : print(round(t_score, 4), round(p_value, 2))  
    -1.9153 0.06
```

◆ 메서드 설명

- ✓ 이 검정은 두 독립 샘플의 평균(예상) 값이 동일하다는 귀무가설에 대한 검정.
- ✓ 두 집단의 분산이 동일하다고 가정함.

메서드	매개변수	설명
	a, b	데이터의 크기가 동일해야 함.
ttest_ind(a, b, equal_var)	equal_var	If true, independent t-test / If false, Welch's t-test

04. 가설 검정 기법 – 대응표본 T-검정

(2023년 11월 기준)

◆ 가설 설정

- ✓ 실험단위를 동질적인 쌍으로 묶은 다음, 각 쌍에서 관측값의 차를 이용하여 두 모평균의 차이에 관한 추론
- ✓ 실험 이전의 집단과 실험 이후의 집단이 동일한 경우 사용하는 검정(쌍체비교)이라고 한다.
- ✓ $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \dots, (X_n, Y_n)$
- ✓ 각 쌍의 차이를 $D_i = X_i - Y_i (i = 1, 2, \dots, n)$ 로 정의할 때, D_i 는 $N(\mu_D, \sigma_D^2)$ 으로부터의 확률표본으로 가정한다.
 - ✓ $\mu_D = \mu_1 - \mu_2$

가설	표현식	설명
H_0	$\mu_D = 0$	실험전후 평균의 차이는 0이다
H_1	$\mu_D \neq 0$	실험전후 평균의 차이는 0이 아니다

◆ 가정(Assumptions) 확인

- ✓ 독립성
- ✓ 정규성

04. 가설 검정 기법 - 대응표본 T-검정

(2023년 11월 기준)

◆ 정규성 검정

- ✓ 실험전후 두 변수의 차이가 정규분포를 따르는지 확인, $N \leq 30$ 이하일 때, 검정해야 함
- ✓ 각 그룹의 변수가 정규분포를 따르는지는 검정할 필요가 없음
- ✓ 각 그룹의 표본수가 모두 $N \geq 30$ 이상일 때, 중심극한정리에 의해 정규성 가정을 만족했다고 봄

가설	설명
H_0	각 자료는 정규분포를 따른다.
H_1	각 자료는 정규분포를 따르지 않는다.

◆ 확인 방법

- ✓ Shapiro-Wilk tests

◆ 가정 위반 시, **Wilcoxon Signed-Ranks Test**를 진행

04. 가설 검정 기법 – 대응표본 T-검정

(2023년 11월 기준)

◆ 가설 설정

- ✓ 실험단위를 동질적인 쌍으로 묶은 다음, 각 쌍에서 관측값의 차를 이용하여 두 모평균의 차이에 관한 추론
- ✓ 실험 이전의 집단과 실험 이후의 집단이 동일한 경우 사용하는 검정(쌍체비교)이라고 한다.
- ✓ $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \dots, (X_n, Y_n)$
- ✓ 각 쌍의 차이를 $D_i = X_i - Y_i (i = 1, 2, \dots, n)$ 로 정의할 때, D_i 는 $N(\mu_D, \sigma_D^2)$ 으로부터의 확률표본으로 가정한다.
 - ✓ $\mu_D = \mu_1 - \mu_2$

가설	표현식	설명
H_0	$\mu_D = 0$	실험전후 평균의 차이는 0이다
H_1	$\mu_D \neq 0$	실험전후 평균의 차이는 0이 아니다

◆ 검정통계량

- ✓ 실험 전 표본평균 X , 실험 후 표본평균 Y 의 차이의 평균(D)에 근거하여 구성

$$t = \frac{D}{S_D / \sqrt{n}} \quad S_D: X \text{와 } Y \text{의 차이에 관한 표준편차}$$

04. 가설 검정 기법 - 대응표본 T-검정

(2023년 11월 기준)

◆ 주요 메서드 - 대응표본 t-검정

- ✓ 실험 전후의 차이를 나타내는 데이터는 정규분포를 이룬다.

```
1 : from scipy import stats  
2 : t_score, p_value = stats.ttest_rel(df['before_spr'], df['after_spr'])  
3 : print(round(t_score, 4), round(p_value, 2))  
14.8933 0.0
```

◆ 메서드 설명

- ✓ 이 검정은 두 독립 샘플의 평균(예상) 값이 동일하다는 귀무가설에 대한 검정.
- ✓ 두 집단의 분산이 동일하다고 가정함.

메서드	매개변수	설명
ttest_rel(a, b)	a, b : array_like alternative	데이터의 크기가 동일해야 함. option : two-sided(default), less, greater

05. 적합도 검정(goodness-of-fit) : χ^2 검정

(2023년 11월 기준)

◆ 가설 설정

- ✓ 어떤 실험에서 관측도수가 가정하는 이론상의 분포를 잘 따른다는 귀무가설을 검정하는 것
- ✓ 관측도수란, 실제 실험에서 단일 특성에 의해 분류된 각 범주의 관측값을 말함
- ✓ 관측도수가 얼마나 이론상의 분포 또는 주어진 형태를 잘 따르는지를 검정하는 가설 검정 기법을 적합도 검정
- ✓ 예) 우리나라 성인의 키가 평균 μ 와 분산 σ^2 을 갖는 정규분포 $N(\mu, \sigma^2)$ 을 따른다고 가정했을 때, 이 가정이 적합한지 검정

가설	설명
H_0	성인의 키는 정규분포 $N(\mu, \sigma^2)$ 를 따른다.
H_1	성인의 키는 정규분포 $N(\mu, \sigma^2)$ 를 따르지 않는다.

◆ 검정통계량

$$\chi^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

O_i : 관측값
 E_i : 기대값

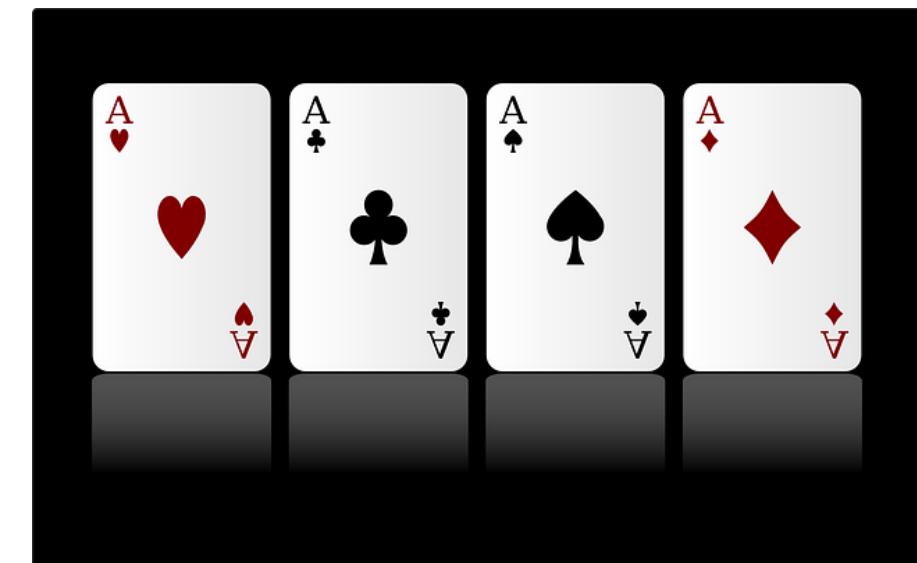
05. 적합도 검정(goodness-of-fit) : χ^2 검정

(2023년 11월 기준)

◆ 가설 설정

- ✓ 어떤 실험에서 관측도수가 가정하는 이론상의 분포를 잘 따른다는 귀무가설을 검정하는 것
- ✓ 관측도수란, 실제 실험에서 단일 특성에 의해 분류된 각 범주의 관측값을 말함
- ✓ 관측도수가 얼마나 이론상의 분포 또는 주어진 형태를 잘 따르는지를 검정하는 가설 검정 기법을 적합도 검정
- ✓ 예) 우리나라 성인의 키가 평균 μ 와 분산 σ^2 을 갖는 정규분포 $N(\mu, \sigma^2)$ 을 따른다고 가정했을 때, 이 가정이 적합한지 검정
- ✓ 예) 트럼프카드의 확률게임

Label	Index i	math.symbol	the value
hearts ♥	0	O_1	64
diamonds ◇	1	O_2	51
spades ♠	2	O_3	50
clubs ♣	3	O_4	35



<출처> <https://ethanweed.github.io/pythonbook/05.01-chisquare.html>

05. 적합도 검정(goodness-of-fit) : χ^2 검정

(2023년 11월 기준)

◆ 가설 설정

- ✓ 어떤 실험에서 관측도수가 가정하는 이론상의 분포를 잘 따른다는 귀무가설을 검정하는 것
- ✓ 관측도수란, 실제 실험에서 단일 특성에 의해 분류된 각 범주의 관측값을 말함
- ✓ 관측도수가 얼마나 이론상의 분포 또는 주어진 형태를 잘 따르는지를 검정하는 가설 검정 기법을 적합도 검정
- ✓ 예) 우리나라 성인의 키가 평균 μ 와 분산 σ^2 을 갖는 정규분포 $N(\mu, \sigma^2)$ 을 따른다고 가정했을 때, 이 가정이 적합한지 검정
- ✓ 예) 트럼프카드의 확률게임

가설	표현식	설명
H_0	$P = (.25, .25, .25, .25)$	모든 경우의 수는 동일하게 같은 확률값으로 선택하게 된다.
H_1	$P \neq (.25, .25, .25, .25)$	적어도 하나는 같은 확률값으로 선택하는 건 아니다

05. 적합도 검정(goodness-of-fit) : χ^2 검정

(2023년 11월 기준)

◆ 검정통계량

- ✓ 예) 트럼프카드의 확률게임

$$\chi^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

O_i : 관측값

E_i : 기대값

Label	Index i	math	관측값	기대 확률값	기대 빈도(E_i) (N=200)	차이값	제곱값	χ^2 통계량
hearts ♥	0	O_1	64	.25	50	14	196	$196/50 = 3.92$
diamonds ◇	1	O_2	51	.25	50	1	1	$1/50 = 0.02$
spades ♠	2	O_3	50	.25	50	0	0	$0/50 = 0.00$
clubs ♣	3	O_4	35	.25	50	-15	225	$225/50 = 4.50$

8.44

05. 적합도 검정(goodness-of-fit) : χ^2 검정

(2023년 11월 기준)

◆ 주요 메서드 - 카이제곱 검정

- ✓ 기준 분포와 동일한지 검정

```
1 : from scipy import stats  
2 : f_score, p_value = stats.chisquare(observed, f_exp = expected)  
3 : print(round(f_score, 4), round(p_value, 2))  
    16.73 0.0
```

◆ 메서드 설명

- ✓ 카이제곱 검정은 범주형 데이터에 주어진 빈도가 있다는 귀무가설을 테스트.

메서드	매개변수	설명
chisquare(f_obs, f_exp)	f_obs : array_like	각 category에서 관측된 빈도수
	f_exp : array_like	각 category의 비율에 따른 기대 빈도수

06. 독립성 검정 : χ^2 검정

(2023년 11월 기준)

◆ 가설 설정

- ✓ 두 범주형 변수 또는 특성이 존재할 때 두 특성이 서로 독립인지 여부에 대해 알아보는 검정
- ✓ 한 특성이 다른 특성에 영향을 미치는지 여부에 대하여 알아보는 검정
- ✓ 예) 영화 장르별 간식 구매의 상관성 검정

영화 장르	간식류 구매	간식류 비구매	행의 합계
작업	50	75	125
코미디	125	175	300
가족	90	30	120
공포	45	10	55
열의 합계	310	290	전체 합계 = 600

<출처 : https://wwwjmp.com/ko_kr/statistics-knowledge-portal/chi-square-test/chi-square-test-of-independence.html>

06. 독립성 검정 : χ^2 검정

(2023년 11월 기준)

◆ 가설 검정

가설	설명
H_0	영화 장르와 간식류 구입은 서로 독립적이다.
H_1	영화 장르와 간식류 구입은 서로 독립적인 것은 아니다 (= 상관성이 존재한다)

◆ 검정통계량

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(E_{ij} - O_{ij})^2}{E_{ij}}$$

$$df = (r - 1)(c - 1)$$

◆ 카이제곱 분포표

✓ $\alpha = 0.05, df = (r - 1)(c - 1)$

06. 독립성 검정 : χ^2 검정

(2023년 11월 기준)

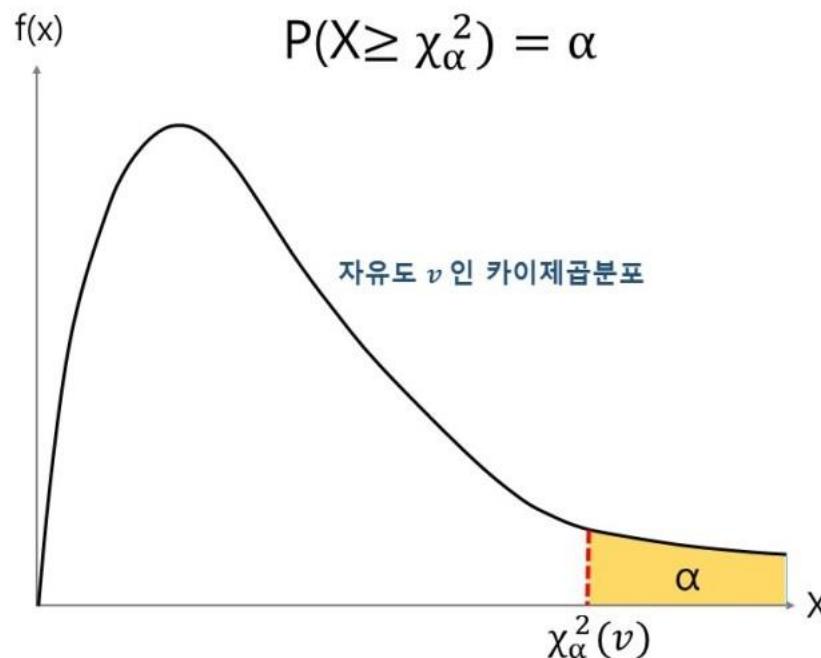
◆ 카이제곱 분포표

✓ $\alpha = 0.05$, $df = (r - 1)(c - 1)$

◆ χ^2 와 카이제곱 분포표와 비교

✓ $\chi^2 >$ 카이제곱 분포표 : H_0 기각

✓ $\chi^2 <$ 카이제곱 분포표 : H_0 채택



Degree of Freedom	Probability of Exceeding the Critical Value								
	0.99	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.01
1	0.000	0.004	0.016	0.102	0.455	1.32	2.71	3.84	6.63
2	0.020	0.103	0.211	0.575	1.386	2.77	4.61	5.99	9.21
3	0.115	0.352	0.584	1.212	2.366	4.11	6.25	7.81	11.34
4	0.297	0.711	1.064	1.923	3.357	5.39	7.78	9.49	13.28
5	0.554	1.145	1.610	2.675	4.351	6.63	9.24	11.07	15.09
6	0.872	1.635	2.204	3.455	5.348	7.84	10.64	12.59	16.81
7	1.239	2.167	2.833	4.255	6.346	9.04	12.02	14.07	18.48
8	1.647	2.733	3.490	5.071	7.344	10.22	13.36	15.51	20.09
9	2.088	3.325	4.168	5.899	8.343	11.39	14.68	16.92	21.67
10	2.558	3.940	4.865	6.737	9.342	12.55	15.99	18.31	23.21
11	3.053	4.575	5.578	7.584	10.341	13.70	17.28	19.68	24.72
12	3.571	5.226	6.304	8.438	11.340	14.85	18.55	21.03	26.22
13	4.107	5.892	7.042	9.299	12.340	15.98	19.81	22.36	27.69
14	4.660	6.571	7.790	10.165	13.339	17.12	21.06	23.68	29.14
15	5.229	7.261	8.547	11.037	14.339	18.25	22.31	25.00	30.58
16	5.812	7.962	9.312	11.912	15.338	19.37	23.54	26.30	32.00
17	6.408	8.672	10.085	12.792	16.338	20.49	24.77	27.59	33.41
18	7.015	9.390	10.865	13.675	17.338	21.60	25.99	28.87	34.80
19	7.633	10.117	11.651	14.562	18.338	22.72	27.20	30.14	36.19
20	8.260	10.851	12.443	15.452	19.337	23.83	28.41	31.41	37.57
22	9.542	12.338	14.041	17.240	21.337	26.04	30.81	33.92	40.29
24	10.856	13.848	15.659	19.037	23.337	28.24	33.20	36.42	42.98
26	12.198	15.379	17.292	20.843	25.336	30.43	35.56	38.89	45.64
28	13.565	16.928	18.939	22.657	27.336	32.62	37.92	41.34	48.28
30	14.953	18.493	20.599	24.478	29.336	34.80	40.26	43.77	50.89
40	22.164	26.509	29.051	33.660	39.335	45.62	51.80	55.76	63.69
50	27.707	34.764	37.689	42.942	49.335	56.33	63.17	67.50	76.15
60	37.485	43.188	46.459	52.294	59.335	66.98	74.40	79.08	88.38
									Not Significant
									Significant

06. 독립성 검정 : χ^2 검정

(2023년 11월 기준)

◆ 각 셀의 기대도수 계산

영화 장르	간식류 구매	간식류 비구매	행의 합계
작업	50 $125 \times 310/600 = 64.58$	75 $125 \times 290/600 = 60.42$	125
코미디	125 155	175 145	300
가족	90 62	30 58	120
공포	45 28.42	10 26.58	55
열의 합계	310	290	전체 합계 = 600

06. 독립성 검정 : χ^2 검정

(2023년 11월 기준)

◆ 각 셀의 검정통계량 계산

영화 장르	간식류 구매	간식류 비구매	행의 합계
작업	차이 $50 - 64.58 = -14.58$ 제곱값 : 212.67 기대값으로 나누기 : $212.67 / 64.58 = 3.29$	3.52	125
코미디	5.81	6.21	300
가족	12.65	13.52	120
공포	9.68	10.35	55
열의 합계	310	290	전체 합계 = 600
검정통계량	$3.29 + 3.52 + 5.81 + 6.21 + 12.65 + 13.52 + 9.68 + 10.35 = 65.03$		
자유도	$(r - 1) \times (c - 1) = (4 - 1) \times (2 - 1) = 3$		

06. 독립성 검정 : χ^2 검정

(2023년 11월 기준)

◆ 각 셀의 검정통계량 계산

영화 장르	간식류 구매	간식류 비구매	행의 합계
작업	3.29	3.52	125
코미디	5.81	6.21	300
가족	12.65	13.52	120
공포	9.68	10.35	55
열의 합계	310	290	전체 합계 = 600
검정통계량	$3.29 + 3.52 + 5.81 + 6.21 + 12.65 + 13.52 + 9.68 + 10.35 = 65.03$		
자유도	$(r - 1) \times (c - 1) = (4 - 1) \times (3 - 1) = 3$		
비교	검정통계량 (65.03) > 카이제곱 분포표 7.81 ($\alpha = 0.05$, $df = 3$)		
해석	H_0 기각 & H_1 채택, 영화장르와 간식류 구매사이에 상관성이 존재한다.		

06. 독립성 검정 : χ^2 검정

(2023년 11월 기준)

◆ 주요 메서드 - 카이제곱 검정

- ✓ 두 범주형 또는 명목형 변수가 서로 연관이 있는지 가능성 여부를 확인하는 통계 가설 검정

```
1 : from scipy import stats  
2 : result = stats.chi2_contingency([X1, X2, X3])  
3 : f_score, p_value = result[0], result[1]  
4 : print(round(f_score, 4), round(p_value, 4))  
    30.1275 0.0
```

◆ 메서드 설명

- ✓ 분할표(contingency table)에서 변수의 독립성에 대한 카이제곱 검정.

메서드	매개변수	설명
chi2_contingency(observed)	observed : array_like	분할표를 말하며, 표에는 각 범주에서 관찰된 빈도(즉, 발생 횟수)가 포함되어 있습니다. 2차원의 경우, 이 테이블은 종종 "R x C 테이블"로 설명됨

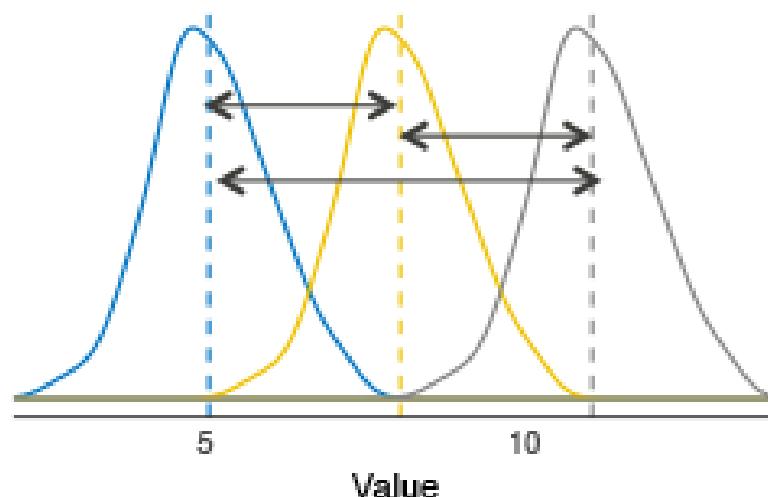
07. 분산분석

(2023년 11월 기준)

- ◆ 두개 이상의 집단에서 그룹 평균 간 차이를 그룹 내 변동에 비교하여 살펴보는 통계분석 방법
 - ✓ 집단 내에 분산보다 다른 집단과의 분산이 더 크면 유의하다고 할 수 있다는 개념을 가지고 있음

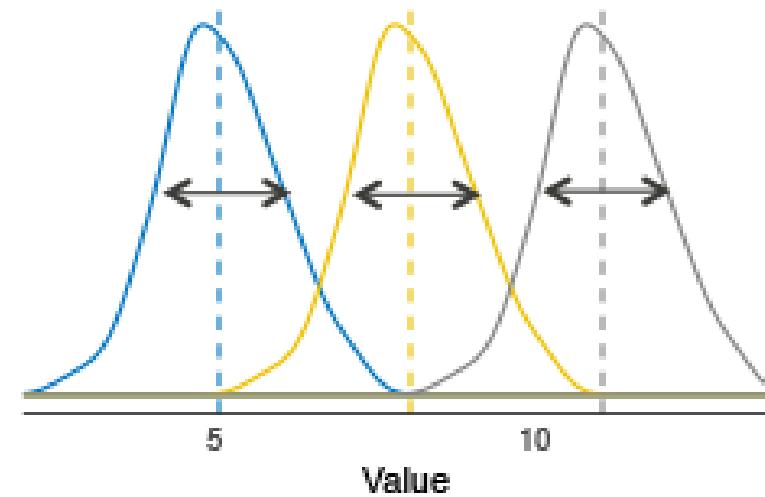
A

Between-group variation
(i.e. Differences among group means)



B

Within-group variation
(i.e. Variability within each group)



07. 분산분석

(2023년 11월 기준)

- ◆ 두개 이상의 집단에서 그룹 평균 간 차이를 그룹 내 변동에 비교하여 살펴보는 통계분석 방법
 - ✓ 집단 내에 분산보다 다른 집단과의 분산이 더 크면 유의하다고 할 수 있다는 개념을 가지고 있음

구분	명칭	독립변수 개수	종속변수 개수
단일변량 분산분석	일원배치 분산분석	1개	
	이원배치 분산분석	2개	1개
	다원배치 분산분석	3개 이상	
다면량 분산분석	MANOVA	1개 이상	2개 이상

◆ 일원배치분산분석 (One-Way ANOVA)

- ✓ 반응값에 대한 하나의 범주형 변수의 영향을 알아보기 위해 사용되는 검증 방법
- ✓ 모집단의 수에는 제한이 없으며, 각 표본의 수는 같지 않아도 된다.
- ✓ F 검정 통계량을 이용한다.

요인	제곱합	자유도	평균제곱	F-value	P
집단 간	SSB	집단수 - 1	$\frac{\text{집단 간 제곱합}}{\text{자유도}}$		-
집단 내	SSW	자료 수 - 집단 수	$\frac{\text{집단 내 제곱합}}{\text{자유도}}$	$\frac{\text{집단 간 평균제곱}}{\text{집단 내 평균제곱}}$	-
합계	SST	자료 수 - 1	-		-

◆ 용어

- ✓ SST(Sum of Squares) : 총 변동
- ✓ SSB(Sum of Squares Between) : 집단 간 변동
- ✓ SSW(Sum of Squares Within) : 집단 내 변동

◆ 일원배치분산분석 (One-Way ANOVA)

- ✓ 반응값에 대한 하나의 범주형 변수의 영향을 알아보기 위해 사용되는 검증 방법
- ✓ 모집단의 수에는 제한이 없으며, 각 표본의 수는 같지 않아도 된다.
- ✓ F 검정 통계량을 이용한다.

◆ 가정

- ✓ 집단의 측정치는 서로 독립적이어야 한다.
- ✓ 각 집단의 데이터를 정규분포를 따른다.
 - 집단별 크기가 25미만이면, 정규성을 체크함
 - 만약, 정규성 가정 위반시, Kruskal-Wallis Test를 사용함
- ✓ 표본 크기가 매우 다를 때, 분산의 동질성을 확인하며, Levene Test를 사용
 - 만약, 등분산 가정이 위반되면 Welch Test를 사용함

◆ 일원배치분산분석 (One-Way ANOVA)

- ✓ 반응값에 대한 하나의 범주형 변수의 영향을 알아보기 위해 사용되는 검증 방법
- ✓ 모집단의 수에는 제한이 없으며, 각 표본의 수는 같지 않아도 된다.
- ✓ F 검정 통계량을 이용한다.

◆ 가설

- ✓ 귀무가설 : K개의 집단 간 평균에는 차이가 없다.
- ✓ 대립가설 : K개의 집단 간 평균이 모두 같다고 할 수 없다.

◆ 주요 메서드 - 일원배치분산분석 (One Way ANOVA)

- ✓ 일원 분산 분석은 두 개 이상의 그룹이 동일한 모집단 평균을 가지고 있다는 귀무가설을 테스트.
- ✓ 테스트는 크기가 서로 다른 두 개 이상의 그룹의 샘플에 적용됩니다.

```
1 : from scipy import stats  
2 : f_score, p_value = stats.f_oneway(X1, X2, X3)  
3 : print(round(f_score, 4), round(p_value, 4))  
4 : 49.16 0.0
```

◆ 메서드 설명

- ✓ 일원배치분산분석을 수행하며, 각 sample은 Series 형태, array 형태로 오면 된다.

메서드	매개변수	설명
f_oneway(sample1, ...)	sample1 : array_like	The sample measurements for each group.

◆ 주요 메서드 - 일원배치분산분석 (One Way ANOVA)

- ✓ 일원 분산 분석은 두 개 이상의 그룹이 동일한 모집단 평균을 가지고 있다는 귀무가설을 테스트.
- ✓ 테스트는 크기가 서로 다른 두 개 이상의 그룹의 샘플에 적용됩니다.
- ✓ 귀무가설 기각, 대립가설 채택 시, 사후검정을 진행해야 함

◆ 사후검정 파이썬 코드 예시

```
1 : from statsmodels.stats.multicomp import pairwise_tukeyhsd  
2 : posthoc = pairwise_tukeyhsd(df['outcome_variables'], df['group'])  
3 : print(posthoc.summary())
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05  
=====
```

group1	group2	meandiff	p-adj	lower	upper	reject
setosa	versicolor	-0.658	0.0	-0.8189	-0.4971	True
setosa	virginica	-0.454	0.0	-0.6149	-0.2931	True
versicolor	virginica	0.204	0.0088	0.0431	0.3649	True

08. 주요 검정 방법 및 메서드 정리 (scipy 기준)

(2023년 11월 기준)

◆ 모수 통계 가설 검정

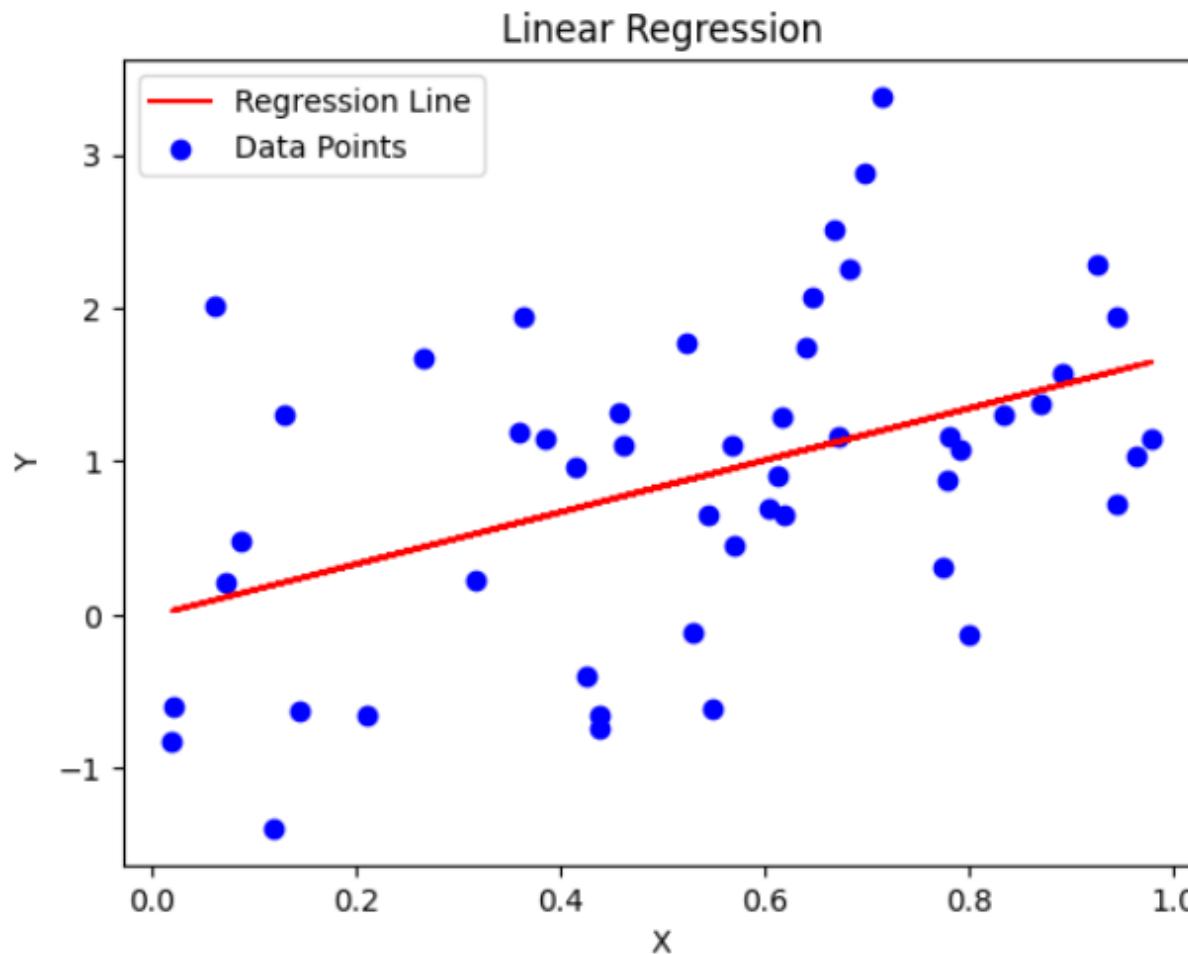
검정 방법	메서드	라이브러리 링크
단일표본 T-검정	.ttest_1samp()	https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_1samp.html
독립표본 T-검정	.ttest_ind()	https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html
대응표본 T-검정	.ttest_rel()	https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_rel.html
일원분산분석	.f_oneway()	https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.f_oneway.html
카이제곱검정(적합도 검정)	.chisquare()	https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chisquare.html
카이제곱검정(독립성 검정)	.chi2_contingency()	https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html

◆ 모수 통계 가설 검정

검정 방법	메서드	라이브러리 링크
정규성 검정	.shapiro()	https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html
등분산성 검정	.levene()	https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.levene.html

◆ 기본 개념

- ✓ 일반적으로 하나의 종속변수 Y 와 여러 독립(=예측)변수 간의 관계를 모델링하는데 사용됨



◆ 선형 회귀 모형 진단

- ✓ 회귀 모형이 적합하려면, 4가지 가정을 모두 만족시켜야 함
- ✓ 예측값과 실제 값의 차이인 잔차(Residual)를 이용하여 검증

가정	의미	진단 방법
선형성	종속변수는 독립변수의 선형 함수	잔차 산점도 : 선형성 확인
독립성	독립변수 사이에는 상관관계가 없어야 함	잔차 산점도 : 특정한 경향성이 없어야 함 더빈-왓슨 검정(Durbin-Watson Test)
등분산성	오차항의 분산은 등분산임	잔차 산점도 : 고르게 분포되어야 함
정규성	오차항의 평균은 0임	Shapiro Wilk 검정 Kolmogorov-Smirnov Goodness of Fit Test Q-Q plot

◆ 검정 설명

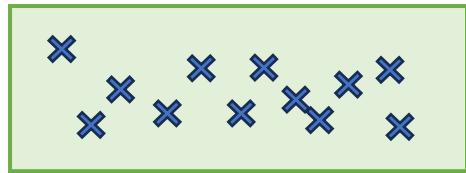
- ✓ Shapiro Wilk 검정 : 데이터의 정규성을 검증하는 방법
- ✓ Kolmogorov-Smirnov 검정 : 데이터가 예상되는 분포에 얼마나 잘 맞는지를 검정

10. 표준편차 vs. 표준오차

(2023년 11월 기준)

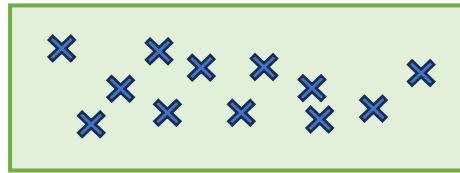
◆ 비교

표준편차(Standard Deviation)	표준오차(Standard Error)
$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$ <p style="text-align: center;">표본평균</p>	$S.E = \frac{\sigma}{\sqrt{n}}$ $S.E = \frac{s}{\sqrt{n}}$
<ul style="list-style-type: none">✓ 각 데이터가 평균과 얼마나 차이를 가지는지 알려주는 지표	<ul style="list-style-type: none">✓ 표본평균의 표준편차✓ 추정값인 표본평균들의 참값인 모평균과의 표준적인 차이✓ 수식에서 n이 커지면 표준오차는 줄어듬



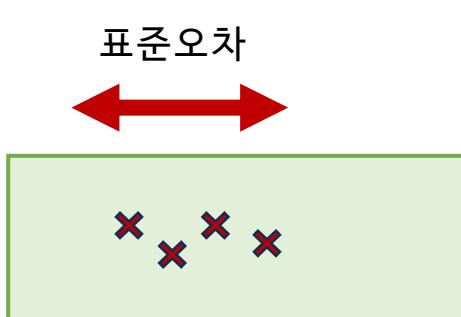
Sample 1 평균

표준편차



Sample 3 평균

표준편차



표준오차

Sample 4 평균

참고문헌

with  python™



빅데이터분석기사

실기 한권완성

파이썬[Python]

최예신, 박진원, 이경숙, 김주현 저

저자직강 동영상강의 교재 | 한글빅 | NAVER 카페

에듀 에듀
EDU EDU



←

Learning Statistics with Python

Search this book...

Learning Statistics with Python

PART I. BACKGROUND

- 1. Why do we learn statistics?
- 2. A brief introduction to research design

PART II. AN INTRODUCTION TO PYTHON

- 3. Getting Started with Python
- 4. More Python Concepts

PART III. WORKING WITH DATA

- 5. Descriptive statistics
- 6. Drawing Graphs
- 7. Data Wrangling
- 8. Basic Programming

PART IV. STATISTICAL THEORY

Learning Statistics with Python by Danielle Navarro and Ethan Weed is licensed under CC BY-SA 4.0 

Next

Learning Statistics with Python

(Python Adaptation by Ethan Weed)

I am a huge fan of [Danielle Navarro's book Learning Statistics with R](#). It is the most accessible statistics book I know of. My students love it. I love it. It's free, and it comes in not only R, but also JASP and JAMOVI flavors. The only problem is, I need to teach intro stats using Python, not R. What to do? Translate the book, obviously!

Since Danielle has been so kind as to open-source the book, I have gone to work translating the R bits to Python, and am learning a lot along the way. ~~To start with, I'm concentrating on translating the code, and putting off editing the textual references to R and R-specific functions for later.~~ Having started with just the code, I have now realized that a better approach is to go through the text line-by-line, and do the job properly the first time. It's a bit slower this way, but ultimately better, I think.

~~It's hard to say how far I'll get, but for now I'm having fun, and am excited that the students in my course won't have to forego this fantastic book, just because they need to do their stats in Python.~~

Update: having by now gotten as far as figuring out how to use Python to overlay the probability density for an F-distribution on top of a distribution created by taking the ratio of scaled random samples from two chi-square distributions, I think I'm committed to seeing this thing through.

Thanks very much to Danielle for the encouragement, and to [Emily Kothe](#) for the [bookdown adaptation](#) of LSR, which has been extremely helpful in creating this Python version.