

# **Advanced Python 01**

## **Jupyter notebook, github, basic numpy & pandas, data exploring**

Stella Sangyoon Bae

3<sup>rd</sup> June 2023

# 01 Introduction

# Who am I?



Sangyoon Bae, 배상윤, BS, Graduate student ([Interdisciplinary Program in Artificial Intelligence](#))

[stellasybae@snu.ac.kr](mailto:stellasybae@snu.ac.kr)

---

## Education

- BS in Biological Sciences (Student Designed Minor: Computational Neuroscience of Human Behavior and Cognition), Seoul National University

## Research Interests

- Multimodality, Graph deep learning, Data-driven computational neuroscience, Biological plausible models, Development of connectome and its contribution to cognitive trait, Simulation, Neural dynamics, Relationship between structure & function

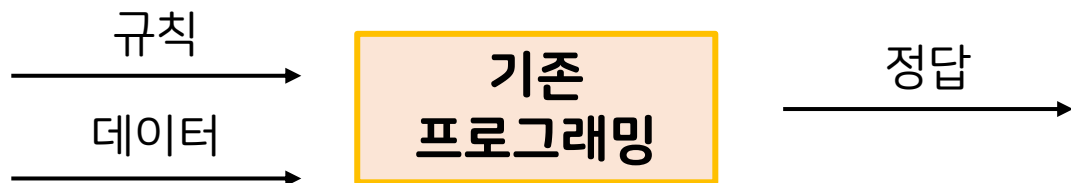
안녕하세요! 저는 서울대학교 공과대학 협동과정 인공지능전공 박사과정 배상윤입니다.

현재 기능적 자기공명 뇌 영상 (fMRI) 을 이용해 정신 질환 조기 진단 모델과 인간의 뇌를 닮은 인공지능을 개발하고 있으며, 서울대 파이썬 강의에서 실습 수업을 진행하는 조교로도 근무하고 있습니다. 만나서 반갑습니다 :)

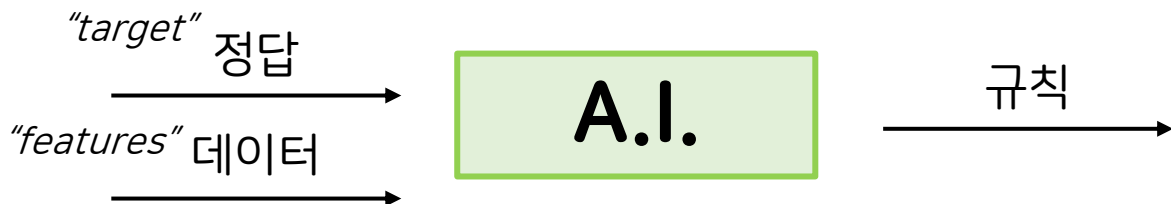
# 수업 진행 방식

- ppt는 간단한 개념을 담고 있어 초반부에 개념을 설명할 때만 사용됩니다.
- 주로 주피터 노트북을 키고 실습하는 것으로 이루어져 있습니다.

# 왜 연구에 A.I.를 사용할까요?



개발자의 역할: 규칙을 파악해서 프로그래밍 언어로 구현하기  
목적: 정답 얻기



개발자의 역할: 많은 예제를 얻어와서 AI에게 학습시키기  
목적: 규칙을 얻어서 결과값을 예측하기

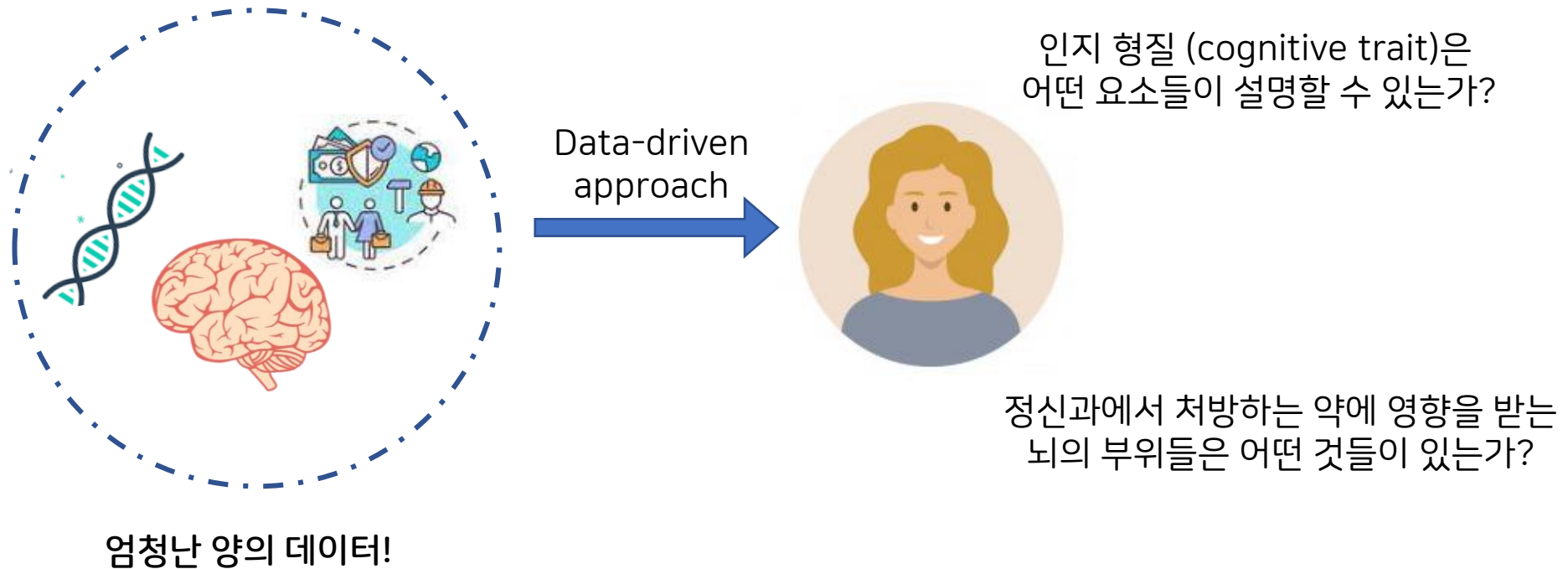
➔ 이런 '규칙'을 신경망 (패턴을 배울 수 있는 일련의 기능) 으로 만드는 것이 딥러닝

우리가 배울 '딥러닝' 은 인공지능의 구현 방법인 머신러닝의 방법론 중 하나이고, 딥러닝은 인공신경망의 한 종류입니다.

즉, 인공지능>머신러닝>인공신경망>딥러닝

# 'AI 기술은 방대한 데이터를 이용해 보이지 않던 것을 볼 수 있게 해준다!'

뇌이미징 데이터를 비롯하여 유전자, 사회경제적 요소 등등을 이용해 복잡한 인간 형질을 설명하려는 시도가 부상하고 있습니다. 정신질환 예측, 자살 예측, 지능 예측 등... AI 기술을 활용해 어떤 유전자가, 어떤 뇌 부위가 해당 형질에 관여하는 지 밝혀낼 수 있죠. 그렇다면 이런 연구에 사용되는 방대한 양의 데이터는 기존의 통계 모델로 처리할 수 있을까요? 통계 모델의 기본 구조 때문에 우리가 놓치는 게 있지는 않을까요?



# 행렬 (matrix)?

인공지능은 엄청나게 큰 데이터를 이용해 엄청나게 많은 계산을 수행합니다. 그렇기 때문에 일일이 한 줄 한 줄 입력을 넣어준다면 실험을 하는 우리도 힘들고, 컴퓨터가 막대한 계산량을 버티지 못할 거예요. 이 때 필요한 게 행렬입니다.

행렬은 괄호 안에 수 또는 문자를 행과 열에 맞게 나열한 것입니다.

$$A = \begin{pmatrix} 1 & 2 & -4 \\ -2 & 1 & 3 \end{pmatrix}, B = \begin{pmatrix} 0 & 1 & 4 \\ -1 & 3 & 1 \end{pmatrix}, C = \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}$$

문제를 행렬이 포함된 식으로 바꿔주면, 컴퓨터가 순식간에 계산을 할 수 있죠. 예를 들어 복잡한 3차 미분방정식을 풀 때, 이를 간단한 1차 미분방정식 3개로 표현해서 행렬로 나타내면 컴퓨터를 통해 쉽게 답을 구할 수 있습니다.

이렇게 만능처럼 보이는 행렬도 크기가 커지면 계산하기 힘들어져요. 이 때 필요한 게 행렬의 성질을 연구하는 학문인 선형대수학입니다. 나중에 대학에 와서 선형대수학을 배우면, 큰 행렬도 쉽게 계산할 수 있어요.

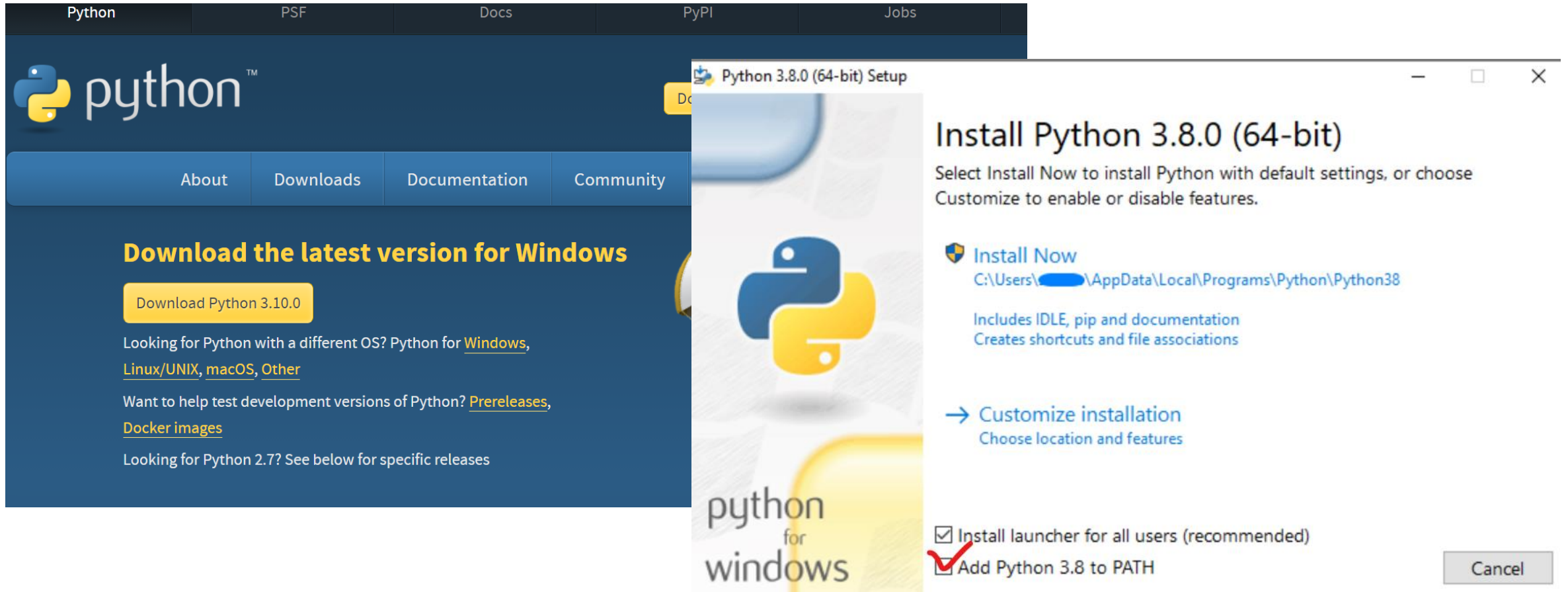
Python에서는 numpy라는 패키지를 이용합니다. 뒤에 가면 나와요!

## **02 Jupyter notebook**



# 01 Install python

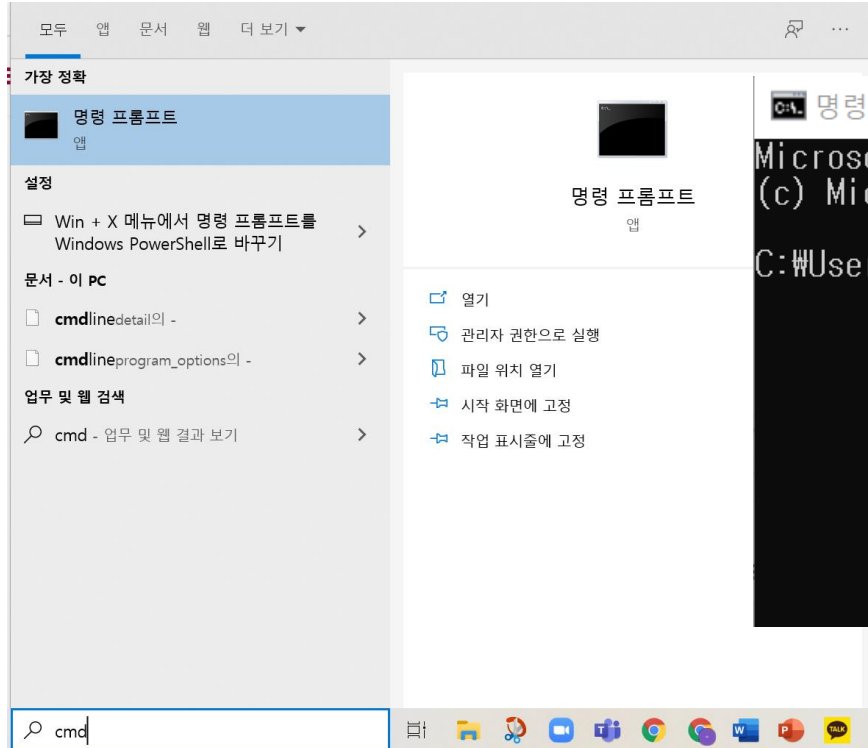
가장 흔하게 쓰이는 컴퓨터 언어로, 인공지능에 관련된 일을 하게 된다면 꼭 숙지해야 하는 언어입니다 영어와 아주 비슷해서 편하게 쓸 수 있어요. 구글에 python download를 치면 되고, 다운로드 할 때 꼭 'add Python to path' 를 선택하기!



# 02 Install Jupyter notebook

크롬과 같은 웹 브라우저 상에서 파이썬 인터프리터를 실행해주는 환경으로, 코드를 가독성 있게 정리하기 좋고, 계산 속도가 매우 빠릅니다.

STEP 1. 시작 창에서 cmd를 입력해 명령 프롬프트를 실행시킵니다.



이런 화면이 뜨면 성공!



# 02 Install Jupyter notebook

STEP 2. pip install jupyter를 입력해 jupyter notebook을 설치합니다.

명령 프롬프트 - pip install jupyter

```
Microsoft Windows [Version 10.0.19042.1052]  
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\stelli>pip install jupyter
```

```
Collecting jupyter
```

```
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
```

```
Collecting nbconvert
```

```
  Downloading nbconvert-6.1.0-py3-none-any.whl (551 kB)
```

```
| 551 kB 2.2 MB/s
```

```
Collecting ipywidgets
```

```
  Downloading ipywidgets-7.6.3-py2.py3-none-any.whl (121 kB)
```

```
| 121 kB 6.8 MB/s
```

```
Collecting jupyter-console
```

```
  Downloading jupyter_console-6.4.0-py3-none-any.whl (22 kB)
```

```
Collecting notebook
```

```
  Downloading notebook-6.4.0-py3-none-any.whl (9.5 MB)
```

```
| 9.5 MB 2.2 MB/s
```

```
Collecting ipykernel
```

```
  Downloading ipykernel-5.5.5-py3-none-any
```

```
| 67 kB 191 kB/s
```

```
Collecting qtconsole
```

```
  Downloading QtConsole-1.9.0-py2.py3-none-any.whl (54 kB)
```

```
| 54 kB 423 kB/s
```

```
Using legacy 'setup.py install' for pyrsistent, since package 'wheel' is not installed.
```

```
Using legacy 'setup.py install' for pandocfilters, since package 'wheel' is not installed.
```

```
Installing collected packages: ipython-genutils, traitlets, six, pywin32, pyrsistent, attrs, wcwidth, tornado, pyzmq, python-dateutil, pyparsing, parso, jupyter-core, jsonschema, webencodings, pygments, pycparser, prompt-toolkit, pickleshare, packaging, nest-asyncio, nbformat, matplotlib-inline, MarkupSafe, jupyter-client, jedi, decorator, colorama, backcall, async-generator, testpath, pywinpty, pandocfilters, nbclient, mistune, jupyterlab-pygments, jinja2, ipython, entrypoints, defusedxml, cffi, bleach, terminado, Send2Trash, prometheus-client, nbconvert, ipykernel, argon2-cffi, notebook, widgetsnbextension, qtpy, jupyterlab-widgets, qtconsole, jupyter-console, ipywidgets, jupyter
```

```
Running setup.py install for pyrsistent ... done
```

```
Running setup.py install for pandocfilters ... done
```

```
Successfully installed MarkupSafe-2.0.1 Send2Trash-1.7.1 argon2-cffi-20.1.0 async-generator-1.10 attrs-21.2.0 backcall-0.2.0 bleach-3.3.0 cffi-1.14.5 colorama-0.4.4 decorator-5.0.9 defusedxml-0.7.1 entrypoints-0.3 ipykernel-5.5.5 ipython-7.25.0 ipython-genutils-0.2.0 ipywidgets-7.6.3 jedi-0.18.0 jinja2-3.0.1 jsonschema-3.2.0 jupyter-1.0.0 jupyter-client-6.1.12 jupyter-console-6.4.0 jupyter-core-4.7.1 jupyterlab-pygments-0.1.2 jupyterlab-widgets-1.0.0 matplotlib-inline-0.1.2 mistune-0.8.4 nbclient-0.5.3 nbconvert-6.1.0 nbformat-5.1.3 nest-asyncio-1.5.1 notebook-6.4.0 packaging-20.9 pandocfilters-1.4.3 parso-0.8.2 pickleshare-0.7.5 prometheus-client-0.11.0 prompt-toolkit-3.0.19 pycparser-2.20 pygments-2.9.0 pyparsing-2.4.7 pyrsistent-0.17.3 python-dateutil-2.8.1 pywin32-301 pywinpty-1.1.3 pyzmq-22.1.0 qtconsole-5.1.0 qtpy-1.9.0 six-1.16.0 terminado-0.10.1 testpath-0.5.0 tornado-6.1 traitlets-5.0.5 wcwidth-0.2.5 webencodings-0.5.1 widgetsnbextension-3.5.1
```

화면에 글자가 계속 써지다가 멈추면 성공!

# 02 Install Jupyter notebook

STEP 3. jupyter notebook을 입력해 jupyter notebook을 실행시킵니다.

```
C:\Users\stell>jupyter notebook
[I 16:34:32.370 NotebookApp] Writing notebook server cookie
[I 16:34:32.901 NotebookApp] Serving notebooks from local di
[I 16:34:32.901 NotebookApp] Jupyter Notebook 6.4.0 is runni
[I 16:34:32.901 NotebookApp] http://localhost:8888/?token=e9
[I 16:34:32.901 NotebookApp] or http://127.0.0.1:8888/?toke
[I 16:34:32.901 NotebookApp] Use Control-C to stop this serv
[C 16:34:32.932 NotebookApp]
```

*이런 화면이 뜨다가 갑자기 인터넷 창이 켜지고 jupyter notebook 창이 나오면 성공!*

To access the notebook, open this file:  
file:///C:/Users/stell/AppData/Local/Programs/Jupyter Notebook/...  
Or copy and paste one of these URLs:  
http://localhost:8888/?token=e9...  
or http://127.0.0.1:8888/?token=e9...

	Name ↓	Last Modified	File size
<input type="checkbox"/>	0		
<input type="checkbox"/>	/		
<input type="checkbox"/>	3D Objects	한 달 전	
<input type="checkbox"/>	Contacts	한 달 전	
<input type="checkbox"/>	Desktop	15분 전	
<input type="checkbox"/>	Documents	9일 전	
<input type="checkbox"/>	Downloads	4분 전	
<input type="checkbox"/>	Favorites	한 달 전	
<input type="checkbox"/>	Links	한 달 전	
<input type="checkbox"/>	Music	한 달 전	
<input type="checkbox"/>	OneDrive	한 시간 전	

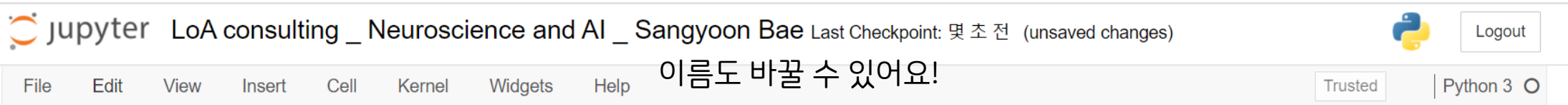
# 02 Install Jupyter notebook

STEP 4. python으로 동작하는 새로운 프로그램을 만들어봅시다.

노트북을 끝 때는 크롬 창을 끄고 나서  
아까 그 까만 창에 Ctrl+C 입력하기!



New → Python3 를  
클릭하면 새로운 notebook이 만들어집니다.



잘 작동하네요☺

Esc+a를 누르면 선택한 셀 앞에 새로운 셀이 생기고,  
Esc+b를 누르면 선택한 셀 뒤에 새로운 셀이 생깁니다.  
Esc+dd를 누르면 선택한 셀을 삭제합니다.

```
In [1]: print('hello world')  
hello world
```

## **03 Numpy and pandas**

# 01 Install numpy and pandas!

## numpy?

행렬 또는 다차원 배열을 위한 연산을 지원하는 Python 라이브러리

## pandas?

데이터 구조 파악, 분석 및 처리를 위한 Python 라이브러리

데이터를 이용해 행렬을 연산하는 AI를 비롯한 datascience 분야에 필수적인 파이썬 패키지들입니다.

오늘은 기초만 익히고 넘어갈 거고, 앞으로 수업시간에 새로운 기능을 하나둘씩 배워나갈 거예요.

일단 주피터 노트북 cell에 아래와 같은 command를 입력하고, 실행해봅시다.

```
!pip install numpy
```

```
!pip install pandas
```

## 02 Numpy (datatype - array)

배열(array): numpy가 데이터를 저장하고 다루는 기본 자료형으로, 행렬 그 자체라고 볼 수 있습니다.

```
▶ import numpy as np
```

```
▶ b = [1.0, 2.0, 3.0] # list  
  
b_np = np.array(b)  
print(b_np)
```

List를 numpy array로 변환한 모습입니다.

```
☞ [1. 2. 3.]
```

Jupyter notebook을 켜서 본격적으로 실습을 해볼까요?



## 03 Pandas (datatype - series)

Series: 데이터를 담을 수 있는 1차원 배열(array)로, 정수, 실수, 문자열, 딕셔너리 등의 다양한 데이터를 받아들여서 구성되어있습니다. pandas가 지원하는 여러 명령을 수행하기 쉽도록 구현된 것입니다.

```
import pandas as pd # 라이브러리 불러오기
```



```
# Series
```

```
s = pd.Series([5, 6, 7, 8, 9]) # 리스트를 데이터로 하는 Series를 생성
```

```
print(s) # 리스트의 각 숫자를 정보(필드값)로 갖는 5개의 레코드로 구성  
# Series이기 때문에 하나의 column만 가지고 있다.  
# 이때 첫번째 column에는 각 데이터 레코드를 나타내는 label 또는 index가 들어간다.  
# (현재는 자동 할당된 index)
```



```
0    5  
1    6  
2    7  
3    8  
4    9  
dtype: int64
```

## 03 Pandas (datatype – DataFrame)

DataFrame: 여러 데이터 필드를 담을 수 있는 2차원 배열입니다.

Series와 유사하지만, 여러 종류의 데이터를 각각의 column으로 하는 테이블 형태의 자료 구조입니다.

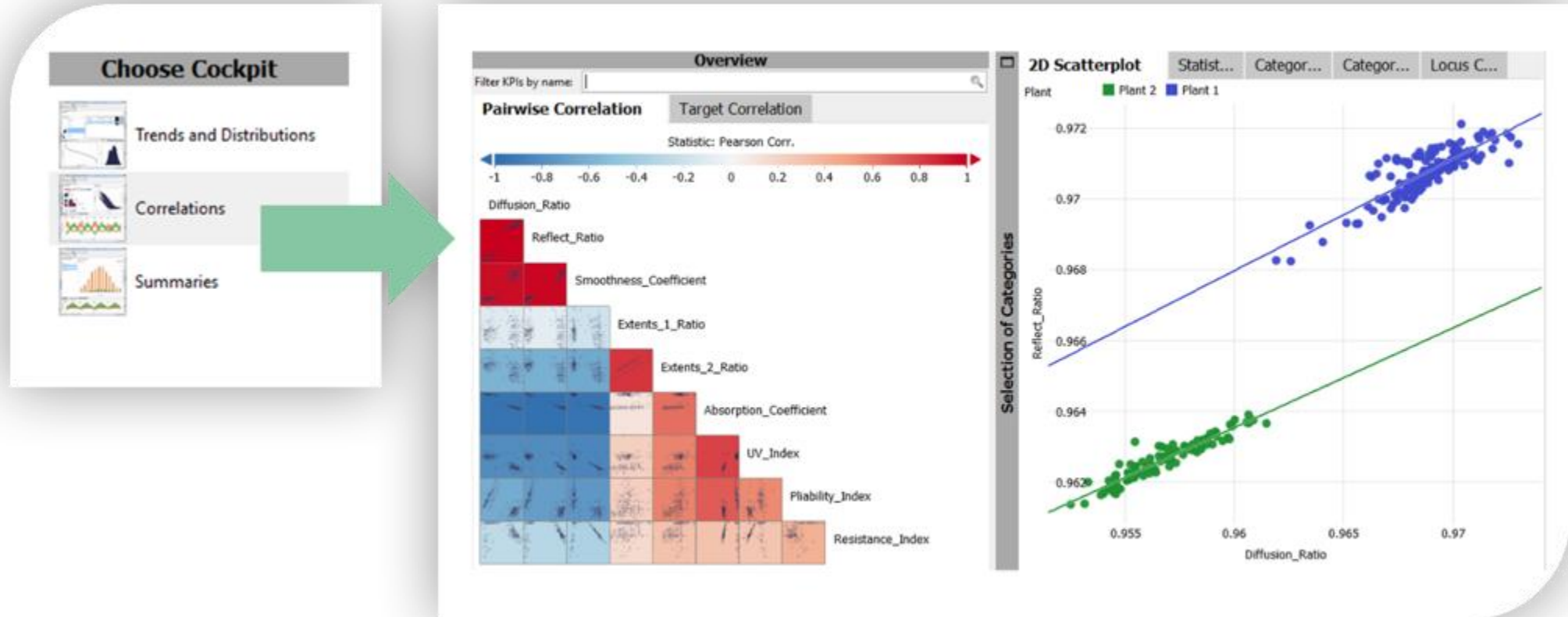
이때 각 column을 이루는 데이터는 서로 다른 type(자료형)일 수 있습니다.

	A	B	C	D	E	F
0	1.0	2021-12-01	1.0	3	test	foo
1	1.0	2021-12-01	1.0	3	train	foo
2	1.0	2021-12-01	1.0	3	test	foo
3	1.0	2021-12-01	1.0	3	train	foo
	float	string (Timestamp)	float	int (categorical)	string	string

Jupyter notebook을 켜서 본격적으로 실습을 해볼까요?

## 04 Data exploring

# 01 Why is data exploration important?

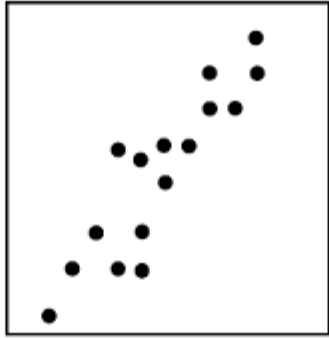


데이터를 분석하기 전 대략적인 overview를 그리기 위해서!

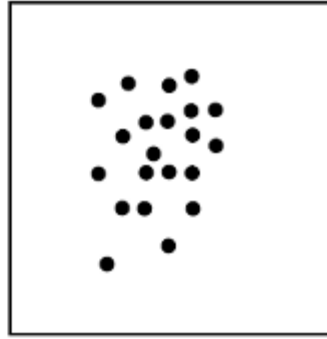
## 02 For example...



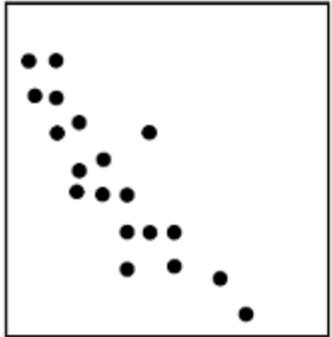
Strong positive correlation



Moderate positive correlation



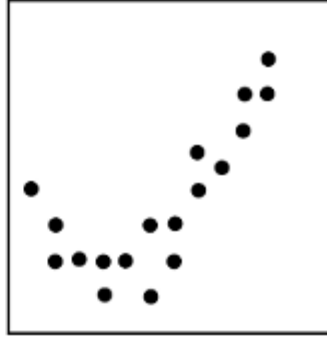
No correlation



Moderate negative correlation



Strong negative correlation



Curvilinear relationship

데이터가 어떤 형태로 분포하는 지 알아야  
어떤 모델에 fitting 할 지 알기 쉽겠죠!

Exploration은 statistical metric을 구해서 대략적인 감  
을 잡고, 데이터를 visualization 해봄으로써 할 수 있습  
니다.

## 03 Statistical metrics

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000



**Iris Versicolor**



**Iris Setosa**



**Iris Virginica**

Scikit-learn Iris는 iris의 세 가지 종류를 구분하는 문제를 푸는 데 쓰이는 데이터셋입니다.

우리는 이 통계치에서 무엇을 볼 수 있을까요?

## 03 Statistical metrics

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

이 표만 보서는 sepal length가 sepal width보다 크다, 등의 단편적인 정보만 알 수 있습니다.

우리가 궁금한 것은 'iris의 종류들이 각각 어떤 특징을 가지느냐?' 이죠.

어떻게 하면 알 수 있을까요?

## 03 Statistical metrics

각각의 iris 종류들만 출력해서 봅시다!



**Iris Setosa**

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	50.00000	50.000000	50.000000	50.000000	50.0
mean	5.00600	3.428000	1.462000	0.246000	0.0
std	0.35249	0.379064	0.173664	0.105386	0.0
min	4.30000	2.300000	1.000000	0.100000	0.0
25%	4.80000	3.200000	1.400000	0.200000	0.0
50%	5.00000	3.400000	1.500000	0.200000	0.0
75%	5.20000	3.675000	1.575000	0.300000	0.0
max	5.80000	4.400000	1.900000	0.600000	0.0



# 03 Statistical metrics

각각의 iris 종류들만 출력해서 봅시다!



Iris Versicolor

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	50.000000	50.000000	50.000000	50.000000	50.0
mean	5.936000	2.770000	4.260000	1.326000	1.0
std	0.516171	0.313798	0.469911	0.197753	0.0
min	4.900000	2.000000	3.000000	1.000000	1.0
25%	5.600000	2.525000	4.000000	1.200000	1.0
50%	5.900000	2.800000	4.350000	1.300000	1.0
75%	6.300000	3.000000	4.600000	1.500000	1.0
max	7.000000	3.400000	5.100000	1.800000	1.0

## 03 Statistical metrics

각각의 iris 종류들만 출력해서 봅시다!



**Iris Virginica**

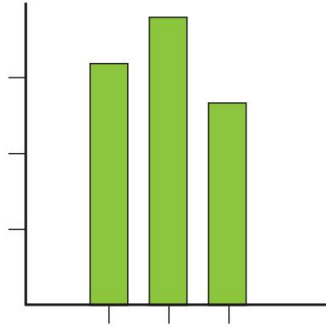
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	50.00000	50.000000	50.000000	50.00000	50.0
mean	6.58800	2.974000	5.552000	2.02600	2.0
std	0.63588	0.322497	0.551895	0.27465	0.0
min	4.90000	2.200000	4.500000	1.40000	2.0
25%	6.22500	2.800000	5.100000	1.80000	2.0
50%	6.50000	3.000000	5.550000	2.00000	2.0
75%	6.90000	3.175000	5.875000	2.30000	2.0
max	7.90000	3.800000	6.900000	2.50000	2.0

iris의 종류들이 각각 어떤 특징을 가지는 지 대략적으로 파악이 가능한가요?

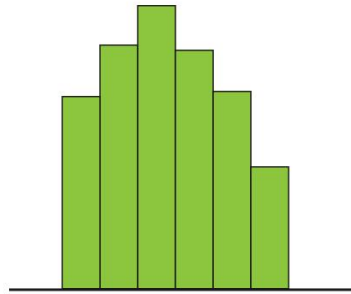
물론 ‘두 그룹이 다르냐 같냐’ 를 알기 위해서는 통계분석이 필요하지만, 짐작 가능하다는 것도 큰 의미를 가지죠!

# 04 Visualization

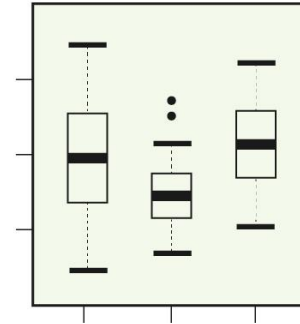
데이터를 시각화 하는 방법은 여러 가지가 있습니다. 데이터의 특성에 따라, 또 탐색 목적에 따라 유용한 방법이 달라집니다.



막대그래프



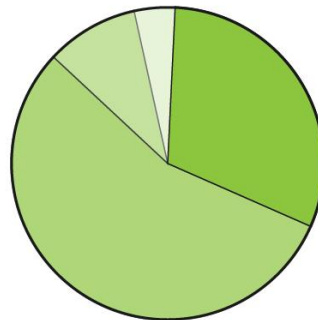
히스토그램



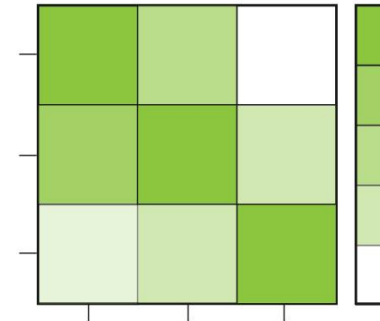
박스플롯



선그래프



파이차트



히트맵

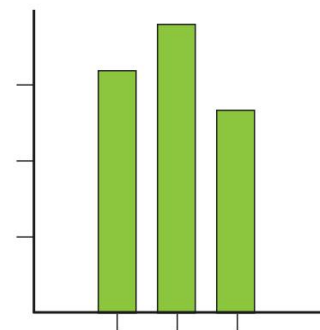
# 04 Visualization

## 막대그래프 (Bar Graph)

막대그래프는 직사각형 막대를 사용하여 범주형 (categorical) 데이터를 표현합니다.

여러 범주 값들의 빈도를 비교할 때 적합합니다.

가로축이 범주, 세로축이 각 범주의 빈도를 나타냅니다.

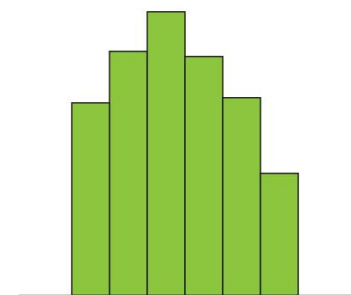


막대그래프

## 히스토그램 (Histogram)

히스토그램은 데이터의 분포를 표로 나타낸 그림입니다. 수치형 데이터에 사용합니다.

가로축에는 데이터의 구간이 표현되고 세로축에 구간별 데이터의 수치가 표현됩니다.



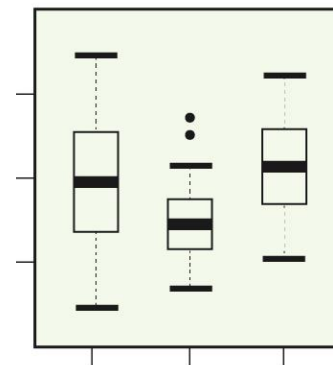
히스토그램

# 04 Visualization

## 박스플롯 (Boxplot)

박스그래프는 수치형 데이터의 분포를 보여주기 위해 사용되는 그래프입니다.

히스토그램과 비슷한 용도로 사용되지만 최소값, 1사분위값, 2사분위값, 3사분위값, 최대값을 쉽게 표현할 수 있다는 장점이 있습니다.



박스플롯

## 선그래프 (Line Graph)

선그래프는 수량을 점으로 표시하고 그 점들을 선으로 이어 그린 그래프입니다.

시간에 따라 지속적으로 변화하는 것을 표현할 때 유용합니다.



선그래프

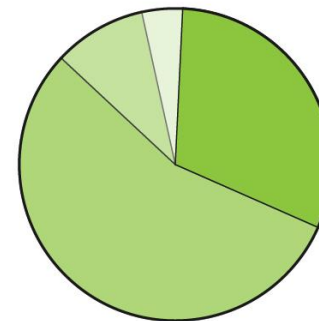
# 04 Visualization

## 파이차트 (Pie Chart)

파이차트는 원그래프라고도 합니다.

전체에 대한 각 부분의 비율을 백분율로 나타낸 그래프 입니다.

전체적인 비율을 쉽게 파악할 수 있습니다.

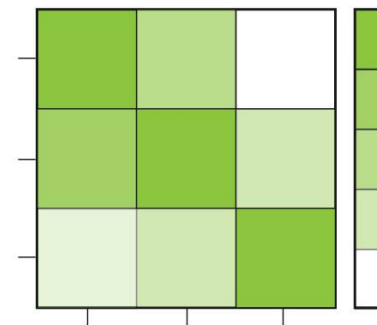


파이차트

## 히트맵 (Heatmap)

히트맵은 열을 뜻하는 히트와 지도를 뜻하는 맵을 결합한 용어입니다.

온도에 따라 색상이 바뀌는 것처럼 수치에 따라 다른 색으로 데이터를 표현해줍니다.



히트맵

# 04 Visualization



Iris Versicolor

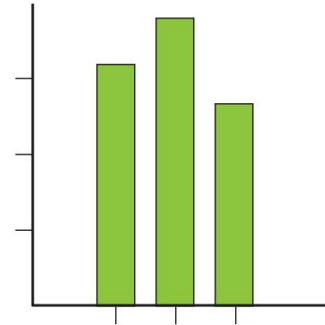


Iris Setosa

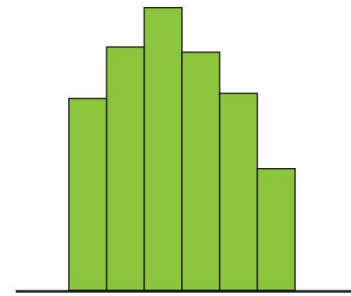


Iris Virginica

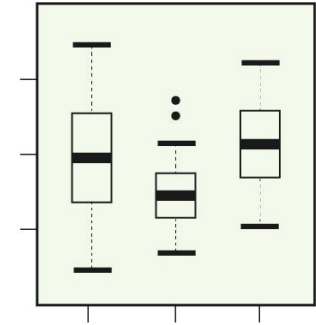
어떤 종류의 그래프가 '붓꽃을 분류하는 문제'를 풀기 전에 data exploration으로 적당할까요?



막대그래프



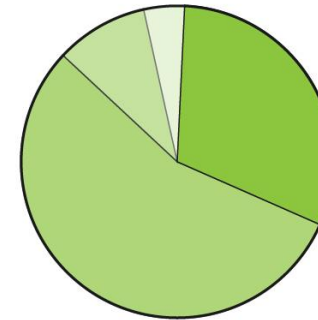
히스토그램



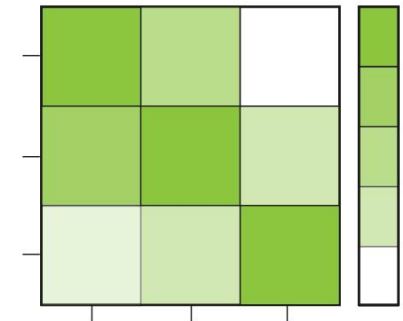
박스플롯



선그래프



파이차트



히트맵

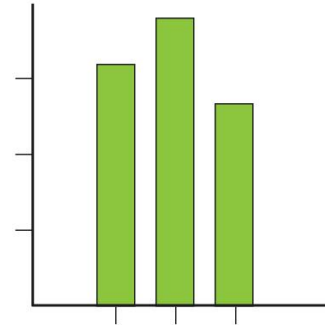
# 04 Visualization



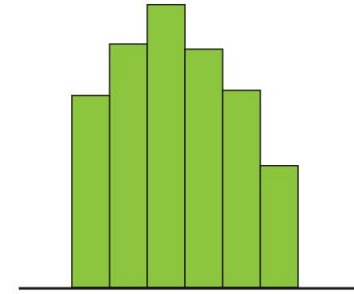
Iris Versicolor

Iris Setosa

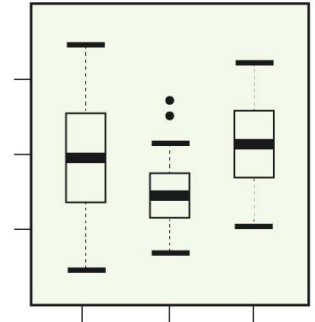
Iris Virginica



막대그래프

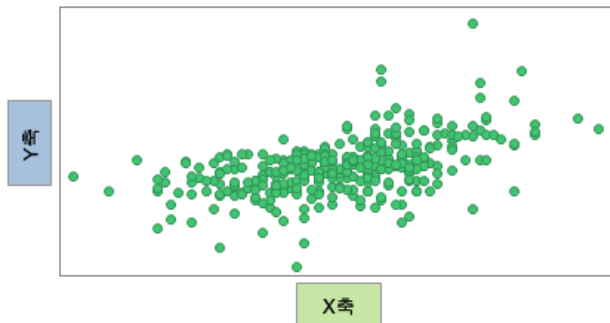


히스토그램

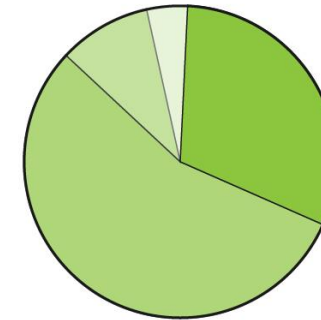


박스플롯

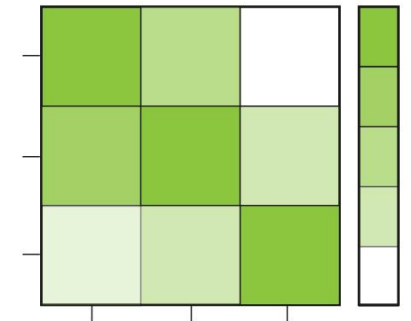
어떤 종류의 그래프가 '붓꽃을 분류하는 문제'를 풀기 전에 data exploration으로 적당할까요?



선그래프



파이차트



히트맵

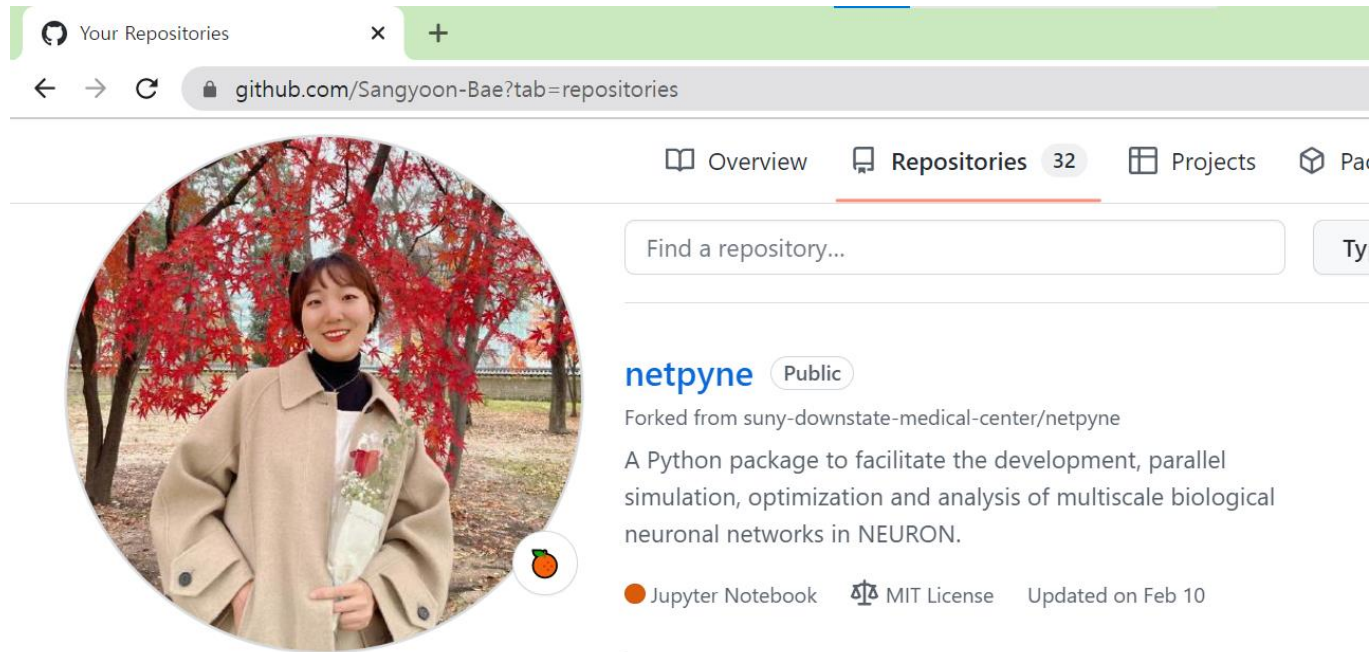
여기에 데이터의 분포를 보기 좋은 산점도 (scatterplot)도 해당합니다.

왜 적당한 지는 실습을 통해 알아보시다!



# 05 Github

# 01 Github



**Stella Sangyoon Bae**

Sangyoon-Bae

Graduate student, Interdisciplinary program in Artificial Intelligence, SNU. Loves neuroimage and bio-inspired AI.

일종의 원격 코드 저장소로, 많은 개발자들이 포트폴리오로 사용합니다.

우선 회원가입부터 해볼까요?

# 01 Github

깃헙에 원격으로 접속할 키를 만들어봅시다! 주피터 터미널을 열어서 아래와 같은 커맨드를 입력해봅시다.

```
ssh-keygen -t ed25519 -C "{name of key} {your email using in github}"
```


엔터를 쫓 치면 어디에 키가 저장되어 있는 지 알려줄거예요. 복사해서 그 파일로 가봅시다.

```
vi {path to the key}
```

# 01 Github

깃헙에 그 키를 등록해봅시다.

1.



Stella Sangyoon Bae

Your personal account [Switch to another account](#)

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Public p

Name

Stella Sangy

Your name may remove it at any

Public email

Select a ver

You have set yo uncheck "Keep

Bio

Graduate st Intelligence

## 2. SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



## 3. SSH keys / Add new

Title

Key type

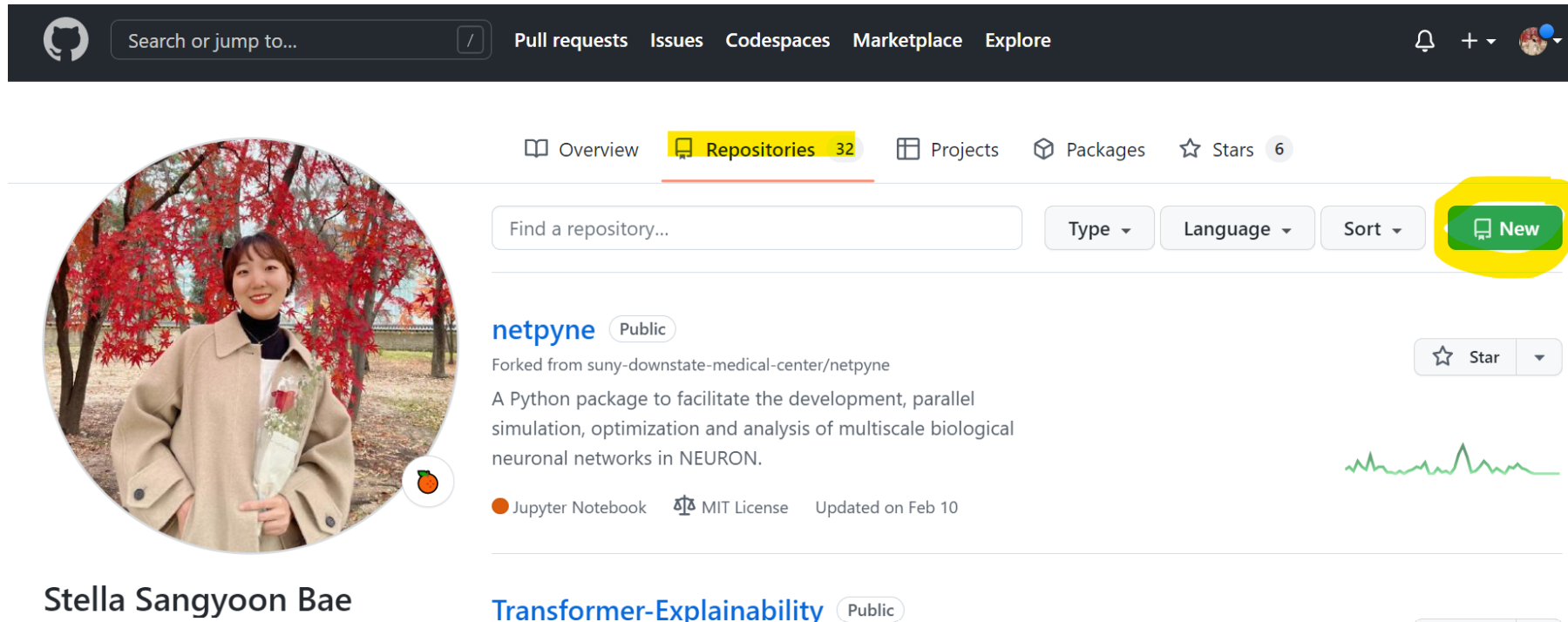
Authentication Key

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

# 01 Github

이제 깃헙을 사용할 준비가 되었어요! Repository를 만들어볼까요?



Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

Overview **Repositories 32** Projects Packages Stars 6

Find a repository... Type Language Sort **New**

**netpyne** Public  
Forked from suny-downstate-medical-center/netpyne  
A Python package to facilitate the development, parallel simulation, optimization and analysis of multiscale biological neuronal networks in NEURON.  
Jupyter Notebook MIT License Updated on Feb 10

**Transformer-Explainability** Public


# 01 Github

이제 깃헙을 사용할 준비가 되었어요! Repository를 만들어볼까요?

---

Owner \*

Repository name \*

 Sangyoon-Bae ▾


 / 

name ✓


Great repository names are short and memorable. Need inspiration? How about [probable-octo-couscous](#)?

Description (optional)

---

☒  Public

Anyone on the internet can see this repository. You choose who can commit.

☐  Private

You choose who can see and commit to this repository.

---

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☒ Add a README file


This is where you can write a long description for your project. [Learn more.](#)

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)


.gitignore template: None ▾

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▾

This will set  main as the default branch. Change the default name in your [settings](#).

---

 You are creating a public repository in your personal account.

---

Create repository

# 01 Github

깃헙 설정을 해볼까요? 먼저 초기화입니다.

```
git init
```

```
PS C:\Users\stell\GIRE basic python> git init  
Initialized empty Git repository in C:/Users/stell/GIRE basic python/.git/
```

이런 화면이 뜨면 성공한 거예요. 이제 사용자 등록을 해봅시다.

```
git config --global user.name {your user name}  
git config --global user.email {your email}
```

# 01 Github

원격 저장소와 우리 컴퓨터의 저장소를 연동해봅시다.

```
git clone {remote repository address}
```

The screenshot shows the GitHub interface for the repository 'Sangyoon-Bae / GIRE\_basic\_python'. The repository is public and has 1 branch (main) and 0 tags. The 'Code' dropdown menu is open, showing options for cloning the repository. The SSH URL is highlighted: `git@github.com:Sangyoon-Bae/GIRE_basic_python`. The repository description is 'Basic python course in GIRE'.

Repository: Sangyoon-Bae / GIRE\_basic\_python (Public)

Navigation: <> Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings

Branches: main (1 branch), 0 tags

Files: Iris\_sample.ipynb (sample code), README.md (Initial commit)

README.md

## GIRE\_basic\_python

Basic python course in GIRE

Clone options:

- Local
- Codespaces (New)
- Clone (HTTPS, SSH, GitHub CLI)
- Open with GitHub Desktop
- Open with Visual Studio

SSH URL: `git@github.com:Sangyoon-Bae/GIRE_basic_python`

Remote address는 여기서 얻을 수 있어요!



# 01 Github

이제 저 폴더 안에서 작업을 해봐요.

`cd {folder name}` #cd란? 현재 위치를 변경해주는 명령어

파일 상태를 확인해봅시다.

`git status`

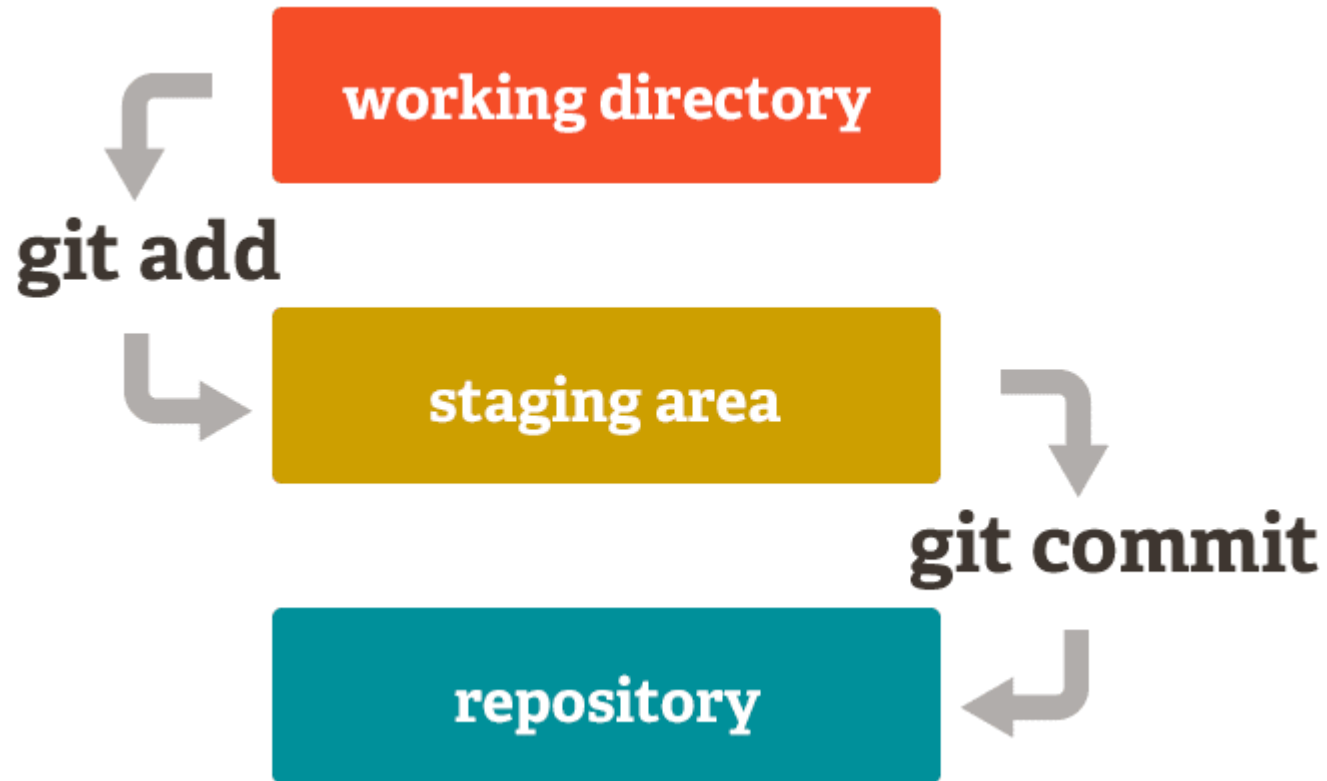
```
On branch main
Your branch is up to date with 'origin/main'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .ipynb_checkpoints/
    Iris_sample.ipynb

nothing added to commit but untracked files present (use "git add" to track)
```

아직 tracking이 안 되었네요.

# 01 Github

Working directory (현재 있는 디렉토리) 에서 git add로 staging area에 올리고, git commit으로 repository에 올려야 해요.



# 01 Github

일단 working directory -> staging area

```
git add {file you want to upload on the github repository}
```

꼭 지금 상태가 어떤 지 확인해봅시다.

```
git status
```

```
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Iris_sample.ipynb

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .ipynb_checkpoints/
```

잘 tracking 하네요.

# 01 Github

tracked file을 staging area에서 repository로 commit 해봅시다.

```
git commit -m "message you want to write"
```

Commit한 파일을 최종 업로드 해볼게요.

```
git push origin main
```

```
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 137.28 KiB | 3.71 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Sangyoon-Bae/GIRE_basic_python.git
b1676e1..f4d5475  main -> main
```



깃헙 레포지토리에서도 잘 올라가 있는 걸 확인할 수 있습니다.