

빅데이터 : 파이썬 데이터 분석

▼ 데이터 분석 프로세스

- 목표 설정 → 데이터 획득 → 데이터 준비 → 데이터 탐색 → 모델링 및 구축 → 발표 및 적용

개발자 입장

	수집	→ 처리	→ 분석	→ 적용
툴 (Python)	requests BeautifulSoup Scrapy	Database SQLAlchemy Pandas	Pandas Numpy Scipy	Matplotlib Plot.ly

▼ Jupyter Notebook

Q. Jupyter라는 이름은 3개의 프로그래밍 언어에서 따왔다. 어떤 언어일까?

A.

J : Julia
P : Python
R : R

- Jupyter = Ju(lia) + Pyt(hon) + R
- 웹 기반 + 통합 개발 환경 + 인터랙티브
- 노트북 = 문서(마크다운) + 코드 + 시각화 + 수식 표현
 - 실행되는 문서!
- IPython에서 시작 (2014년)
- 데이터 과학 분야의 표준 도구 (De Facto)
- 코드 작성과 실행, 출력 보기, 시각화 출력
- 할 수 있는 것들
 - 마크다운으로 문서화
 - Python 등 코드를 실행하고 결과 확인
 - REPL (Read Evaluate Print Loop), Interactive

▼ 도움

- 노트북 도움말 : h
- API 도움 받기 : help(), ?, ??
- 자동완성 : TAB
- Tool Tip : Shift + tab
- Magic 명령어 : % 또는 %, %magic, %load, %run, %history, %lsmagic ..

◦ 시각화 자료 통합

▼ 수식 표현

- \$, \$\$: LaTeX
- \begin, \end

$$\int_0^{\infty} \frac{x^3}{e^x - 1} dx = \frac{\pi^4}{15}$$

이게 뭐야

$$\begin{aligned} a_1 &= b_1 + c_1 \\ a_2 &= b_2 + c_2 - d_2 + e_2 \\ a_{11} &= b_{11} & a_{12} &= b_{12} \\ a_{21} &= b_{21} & a_{22} &= b_{22} + c_{22} \end{aligned}$$

◦ 셀 실행

- 셀 또는 터미널 실행창 (New → Terminal)

◦ 만들어진 노트북으로 슬라이드쇼 가능

- html로 저장

```
jupyter nbconvert some-notebook.ipynb --to slides
```

- 웹 서버로 실행

```
jupyter nbconvert some-notebook.ipynb --to slides --post serve
```

- RISE : rise.readthedocs.io

- 기본

- 셀(Cell) 단위로 작성하고 실행한다
- 선언한 변수 등은 노트북 안에서 컨텍스트를 갖는다
- 노트북의 내용 (코드, 마크다운, 출력 등)은 checkpoint로 저장(캐싱)된다
- 노트북 파일(ipynb)는 JSON 파일

- 확장

- extension : `jupyter_contrib_nbextensions`
- widgets : `jupyter widgets`
- theme : `jupyter theme`

- 어디에 쓸까?

- 데이터 분석과 개발 과정 전반에서 사용
- 개발 프로토타입을 만들 때
- 그냥 개발용으로 (지원 언어 40+)

- 누가 쓸까?

- 데이터를 다루는 누구나!
- 개발과 기록을 한 번에, 내보내기과 공유

- 시작하기

- 설치형
 - 그냥 설치 (Python, pip)
 - **anaconda** : 패키지 + 환경 관리
 - `docker`
- 서비스형

- Google Colab
- Kaggle
- Cloud : AWS, GCP, Azure

▼ 데이터 구하기

- Data Sources : RDB, DW, Data Lake, File(CSV, log, txt, Excel), Service, ...
- Open Dataset, Open API
- 스크래핑
- 한땀한땀 손맛
- 돈맛

스크래핑

- 파이썬으로 쉽고 편하게 스크래핑을 개발할 수 있다
- requests, urllib3
- scrapy
- selenium
- 파싱 : BeautifulSoup, lxml, JSON
- 단, 규칙 (robot.txt, 저작권 정보)과 예의(과도한 요청 자제)를 지키자

▼ 데이터 처리와 분석

- pandas + Scipy + Numpy + python
- 데이터를 불러오고
 - 닦고 채우고 지우고 자르고 붙이고 돌리고 묶고 분류하고 계산하고 변형한 뒤,,,
- 분석, 저장하거나 다음 단계로 전달

▼ 시각화

- 분석된 숫자가 있는데 왜 필요할까?

- 필요성 : 앤스컴 콰르텟
- 예뻐서.. → 사람들이 이해하기 쉽게 전달 하려고
- EDA의 중요한과정
- 데이터 탐색 : 데이터 이해, 트렌드와 패턴 파악, 특이값 찾아내기, 모델 선정
- 패턴을 인식하는 인간의 능력 활용하기
- 데이터 리터러시 (데이터 문해력)
- 시각화 방법 선택 → 다방면으로 살펴보기
- ▼ 데이터 시각화 형태 고르는 법

💡 데이터 시각화 형태 고르는 방법

<https://brunch.co.kr/@joecool/148>

How to Choose the Right Type of Chart for Your Message
(<https://education.microsoft.com/ko-kr/course/0a60eeb6/1>)

<https://www.python-graph-gallery.com/>

- 많이 사용하는 라이브러리들
 - matplotlib : 기본
 - seaborn : 쉬운 API, matplotlib 기반
 - plot.ly and plot.ly express : Javascript, Interactive
 - Bokeh
 - Altair and vega
 - 모두 Jupyter Notebook에서 사용 가능
- matplotlib
 - 이것저것
 - 기본
 - 타이틀 : set_title

- 레이블 : set_xlabel, set_ylabel 또는 plot에 label 속성
- 범례 : legend

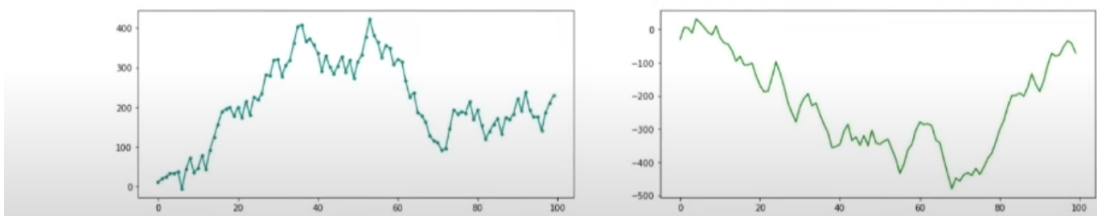
▼ 여러개 그리기 : subplots

```
In [7]: figure, axes = plt.subplots(nrows=1, ncols=2, figsize=(20, 4))

data_a = np.random.randint(-50, 50, 100).cumsum()
data_b = np.random.randint(-50, 50, 100).cumsum()

axes[0].plot(data_a, color='teal', marker='.') # 색상과 마커를 함께 지정
axes[1].plot(data_b, 'g')
```

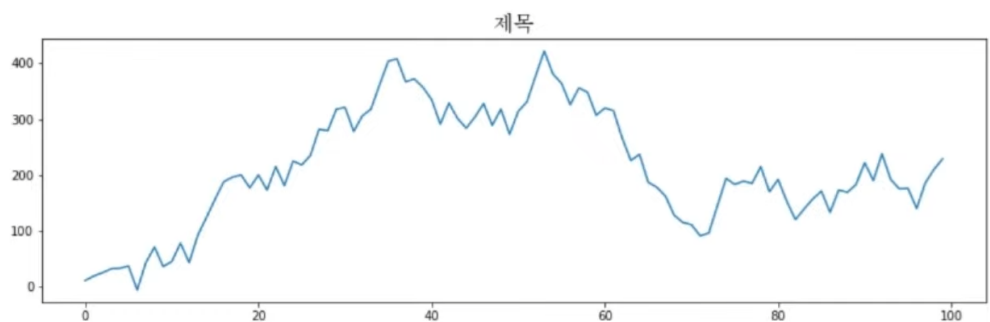
Out[7]: [<matplotlib.lines.Line2D at 0x21ed4a51288>]



```
In [8]: # 폰트 설정
import matplotlib.font_manager as fm
fp = fm.FontProperties("batang", size=18)

plt.title("제목", fontproperties=fp)
plt.plot(data_a)
```

Out[8]: [<matplotlib.lines.Line2D at 0x21ed4ab8688>]



▼ Scatter

- Prep iris data

```
In [10]: import requests

iris_url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
r = requests.get(iris_url)
open('iris.csv', 'wb').write(r.content)
```

Out[10]: 4551

```
In [14]: # https://data-science-history.com/109

import re

iris=pd.read_csv('iris.csv', header=None)
iris.columns = ["sepal length", "sepal width", "petal length", "petal width", "class"]
iris['class'] = iris['class'].map(lambda i: re.sub('Iris-', '', i))

# class별로 데이터를 분할하기.
setosa = iris[iris['class'] == 'setosa']
versicolor = iris[iris['class'] == 'versicolor']
virginica = iris[iris['class'] == 'virginica']

plt.figure(figsize=(8, 6))
plt.style.use('ggplot')

# class 별로 마킹을 다르게 하기.
setosa_sc = plt.scatter(setosa['sepal length'], setosa['sepal width'], marker='o', color='b')
versicolor_sc = plt.scatter(versicolor['sepal length'], versicolor['sepal width'], marker='x', color='r')
virginica_sc = plt.scatter(virginica['sepal length'], virginica['sepal width'], marker='v', color='k')

# legend 좌측 상단에 삽입
plt.legend((setosa_sc, versicolor_sc, virginica_sc), ('setosa', 'versicolor', 'virginica'), loc='upper left')

# 제목 달기.
plt.title("Iris dataset", fontsize=20)
```

▼ 다른 시각화 툴

▼ Seaborn

- matplotlib 기반
- 좀 더 쉬운 API와 스타일

In [12]:

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style="dark")

# Simulate data from a bivariate Gaussian
n = 10000
mean = [0, 0]
cov = [(2, .4), (.4, .2)]
rng = np.random.RandomState(0)
x, y = rng.multivariate_normal(mean, cov, n).T

# Draw a combo histogram and scatterplot with density contours
f, ax = plt.subplots(figsize=(6, 6))
sns.scatterplot(x=x, y=y, s=5, color=".15")
sns.histplot(x=x, y=y, bins=50, pthresh=.1, cmap="mako")
sns.kdeplot(x=x, y=y, levels=5, color="w", linewidths=1)
```

Out[12]: <AxesSubplot:>




```
In [13]: import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme()

# Load the example flights dataset and convert to long-form
flights_long = sns.load_dataset("flights")
flights = flights_long.pivot("month", "year", "passengers")

# Draw a heatmap with the numeric values in each cell
f, ax = plt.subplots(figsize=(9, 6))
sns.heatmap(flights, annot=True, fmt="d", linewidths=.5, ax=ax)
```

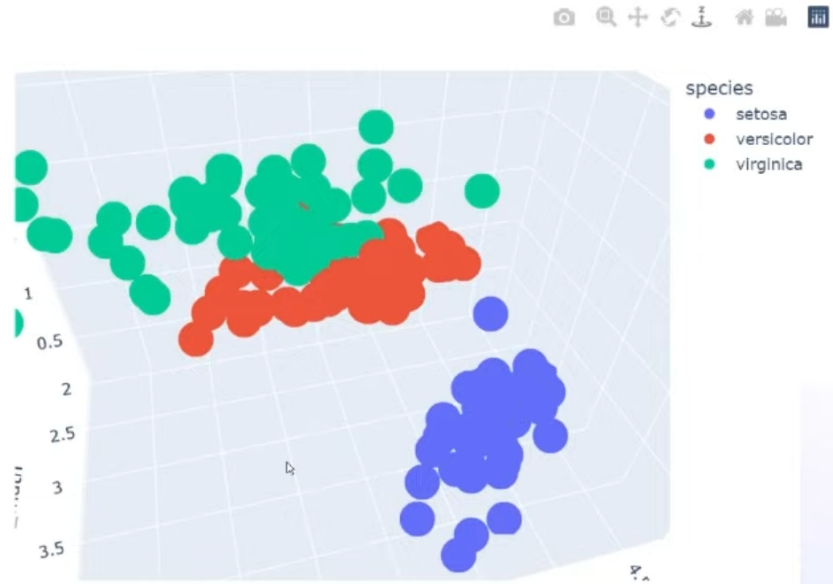
Out[13]: <AxesSubplot:xlabel='year', ylabel='month'>



▼ plotly and Express

- js 기반
- 인터랙티브

```
In [2]: import plotly.express as px
df = px.data.iris()
fig = px.scatter_3d(df, x='sepal_length', y='sepal_width', z='petal_width',
                    color='species')
fig.show()
```



- bokeh






▼ 데이터 분석 개선하기

- 협업
 - nbconvert, nbviewer, binder : Publish
- Jupyter Lab : 차세대 Notebook
- 성능
 - 코드 : 프로파일링
 - GPU 가속 : CUDA, cuDF
 - Clustering / parallelism : dask, vaex
- JIT : PyPy, numba

- Cloud
 - AWS SageMaker
 - GCP AI Platform
 - Azure ML Notebook
 - JetBrains DataLore
- Alt. Tool
 - VSCode도 노트북 지원
 - JetBrains DataSpell

▼ Closing

더보기

- 컨퍼런스 / 커뮤니티 : PyCon, PyData, JupyterCon
- 참조
 -  파이썬 라이브러리를 활용한 데이터 분석
 -  파이썬 데이터 사이언스 핸드북
 -  밑바닥부터 시작하는 데이터 과학
 -  데이터 과학 트레이닝 북
 -  jupyterbook, kaggle, dacon, github, etc