

# 4. 에너지 데이터 분석 실습



# 넘파이(Numpy)

NumPy(Numerical Python)는 데이터 분석을 포함해 수학과 과학연산을 위한 파이썬 기본 패키지로 고성능의 다차원 배열 객체와 다양한 객체에 대해 고속 연산을 가능하게 합니다.

```
data = np.array([1,2,3])
```

data

1
2
3

data

1
2
3

.max() = 3

data

1
2
3

.min() = 1

data

1
2
3

.sum() = 6

```
np.array([ [[1,2],[3,4]],  
          [[5,6],[7,8]] ])
```



	5	6
1	2	8
3	4	

```
np.random.random((4,3,2))
```

	0.3	0.6	0.8
0.2	0.5	0.3	0.8
0.7	0.6	0.1	0.5
0.4	0.5	0.5	0.3
0.1	0.1	0.4	

$$MeanSquareError = \frac{1}{n} \sum_{i=1}^n (Y_{prediction_i} - Y_i)^2 \rightarrow \text{error} = (1/n) * \text{np.sum}(\text{np.square}(\text{predictions} - \text{labels}))$$

# 넘파이(Numpy)

```
import numpy as np
lst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
arr = np.array(lst)
arr
```

```
[Out] array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

```
arr.mean()
```

```
[Out] 5.5
```

```
print('Mean:', arr.mean())
```

```
print('Median:', np.median(arr))
```

```
print('Range (Max - min):', np.ptp(arr))
```

```
print('Standard deviation:', arr.std())
```

```
print('80th percentile:', np.percentile(arr, 80))
```

```
print('0.2-quantile:', np.quantile(arr, 0.2))
```

```
[Out]
```

```
Mean: 5.5
```

```
Median: 5.5
```

```
Range (Max - min): 9
```

```
Standard deviation: 2.87228132323
```

```
80th percentile: 8.2
```

```
0.2-quantile: 2.8000000000000003
```

# 판다스(Pandas)

Pandas는 데이터 분석을 위해 널리 사용되는 파이썬 라이브러리 패키지입니다.  
행과 열로 이루어진 데이터프레임 자료구조를 사용하여, 데이터 분석에 있어 높은 수준의 성능을 발휘합니다.

## ■ 데이터프레임(DataFrame)

컬럼명(Column Names)

	power_usage	gas_usage	water_usage
2022-05-01	86	98	84
2022-05-02	70	31	88
2022-05-03	0	29	51
2022-05-04	32	77	75
2022-05-05	57	48	17

인덱스  
(Index)

데이터

# 판다스(Pandas)

## ■ Pandas 라이브러리 импорт

```
import pandas as pd
```

## ■ 데이터프레임 생성

```
dates = pd.date_range('20220501', periods=31)
data = np.random.randint(0,100,(31,3))
df = pd.DataFrame(data=data, index=dates,
                  columns=['power_usage', 'gas_usage', 'water_usage'])
```

## ■ 데이터 확인

df.head()

	power_usage	gas_usage	water_usage
2022-05-01	86	98	84
2022-05-02	70	31	88
2022-05-03	0	29	51
2022-05-04	32	77	75
2022-05-05	57	48	17

df.tail()

	power_usage	gas_usage	water_usage
2022-05-27	90	84	63
2022-05-28	39	33	29
2022-05-29	5	77	72
2022-05-30	42	73	8
2022-05-31	54	27	3

# 판다스(Pandas)

## ■ 자료구조 파악

df.info()

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 31 entries, 2020-01-01 to 2020-01-31
Freq: D
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  -
0   power_usage     31 non-null    int64
1   gas_usage       31 non-null    int64
2   water_usage     31 non-null    int64
dtypes: int64(3)
memory usage: 992.0 bytes
```

## ■ 데이터 타입 확인

df.dtypes

```
power_usage    int64
gas_usage      int64
water_usage    int64
dtype: object
```

## ■ Null 데이터 확인

df.isnull().sum()

```
power_usage    0
gas_usage      0
water_usage    0
dtype: int64
```

# 판다스(Pandas)

## 통계 정보

`df.describe()`

	power_usage	gas_usage	water_usage
count	31.000000	31.000000	31.000000
mean	45.161290	52.193548	52.161290
std	26.669704	31.679036	30.625803
min	3.000000	4.000000	0.000000
25%	29.000000	22.500000	18.500000
50%	41.000000	54.000000	57.000000
75%	61.500000	84.000000	76.000000
max	97.000000	99.000000	99.000000

## 데이터 상관관계 분석

`df.corr()`

	power_usage	gas_usage	water_usage
power_usage	1.000000	-0.187838	0.073957
gas_usage	-0.187838	1.000000	0.007285
water_usage	0.073957	0.007285	1.000000

# 판다스(Pandas)

## ■ 새 컬럼 생성하기

```
df['energy_usage'] = df['power_usage'] + df['gas_usage']  
df['date'] = pd.to_datetime(df.index, format= '%Y-%m-%d')  
df['day_of_week'] = df['date'].dt.day_name()  
print(df)
```

	power_usage	gas_usage	water_usage	energy_usage	date	day_of_week
2020-01-01	95	86	64	181	2020-01-01	Wednesday
2020-01-02	31	82	2	113	2020-01-02	Thursday
2020-01-03	77	16	55	93	2020-01-03	Friday
2020-01-04	43	4	0	47	2020-01-04	Saturday
2020-01-05	44	79	57	123	2020-01-05	Sunday
2020-01-06	27	63	52	90	2020-01-06	Monday
2020-01-07	45	33	16	78	2020-01-07	Tuesday



# 판다스(Pandas)

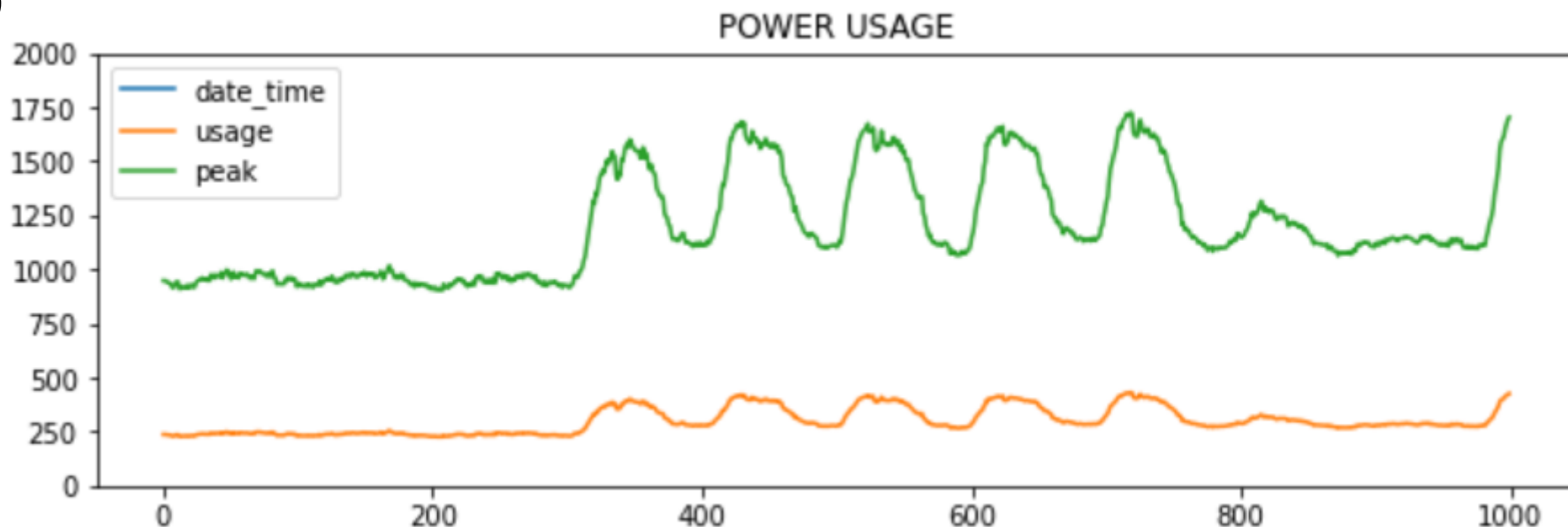
## ■ 파일에서 데이터를 로드하는 방법

```
# df_energy = pd.read_csv('data.csv')
```

```
df_energy = pd.read_csv(  
    'https://raw.githubusercontent.com/kgpark88/energy-bigdata-analysis/master/data.csv')
```

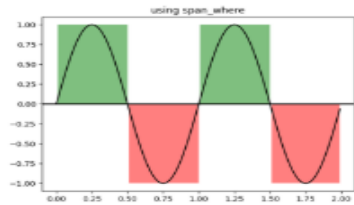
```
df_energy.plot(title='POWER USAGE', figsize=(10, 3), ylim=(0, 2000))
```

```
plt.show()
```

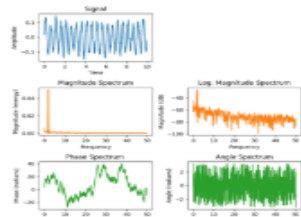


# 맷플롯립(Matplotlib)

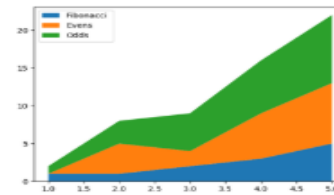
파이썬에서 데이터를 차트나 플롯(Plot)으로 그려주는 라이브러리 패키지로서  
가장 많이 사용되는 데이터 시각화(Data Visualization) 패키지입니다.



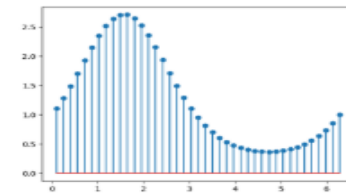
Using span\_where



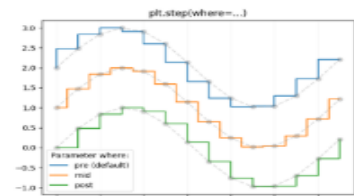
Spectrum Representations



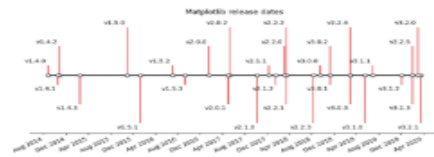
Stackplot Demo



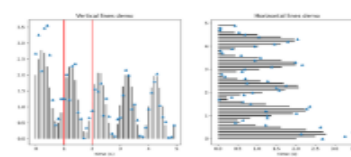
Stem Plot



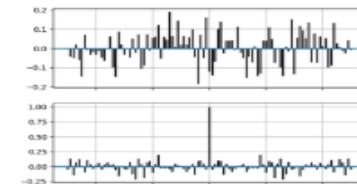
Step Demo



Creating a timeline with lines, dates, and text



hlines and vlines



Cross- and Auto-Correlation Demo

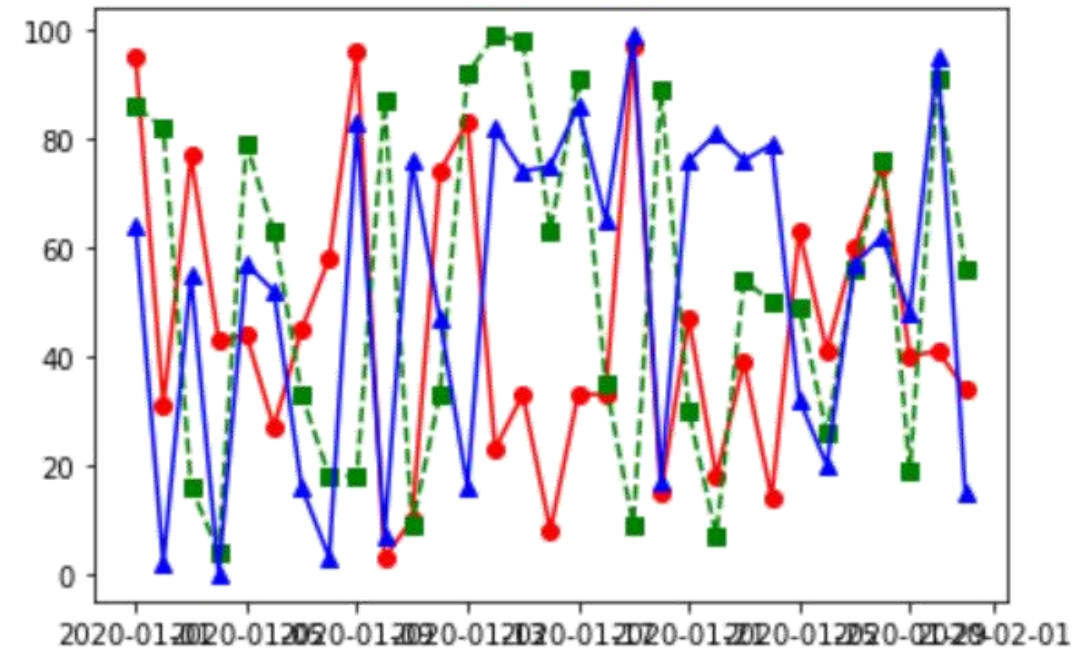
# 맷플롯립(Matplotlib)

## ■ 라이브러리 импорт

```
import matplotlib.pyplot as plt
```

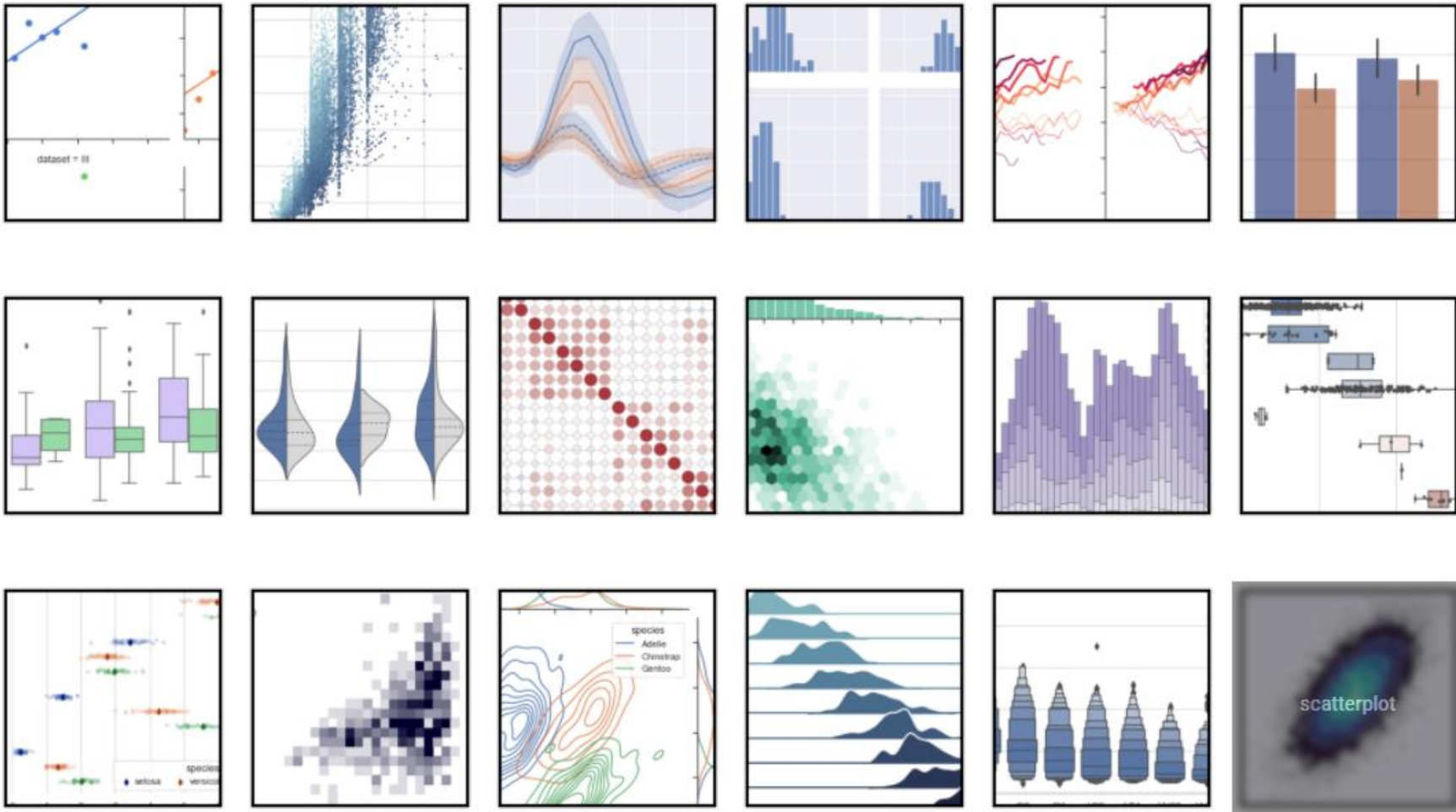
## ■ 선 그래프

```
plt.plot(df['power_usage'], 'ro-',  
         df['gas_usage'], 'gs--',  
         df['water_usage'], 'b^-',  
         plt.show())
```



# 씨본(Seaborn)

Matplotlib을 기반으로 다양한 색상 테마와 통계용 차트 등의 기능을 추가한 시각화 패키지입니다.



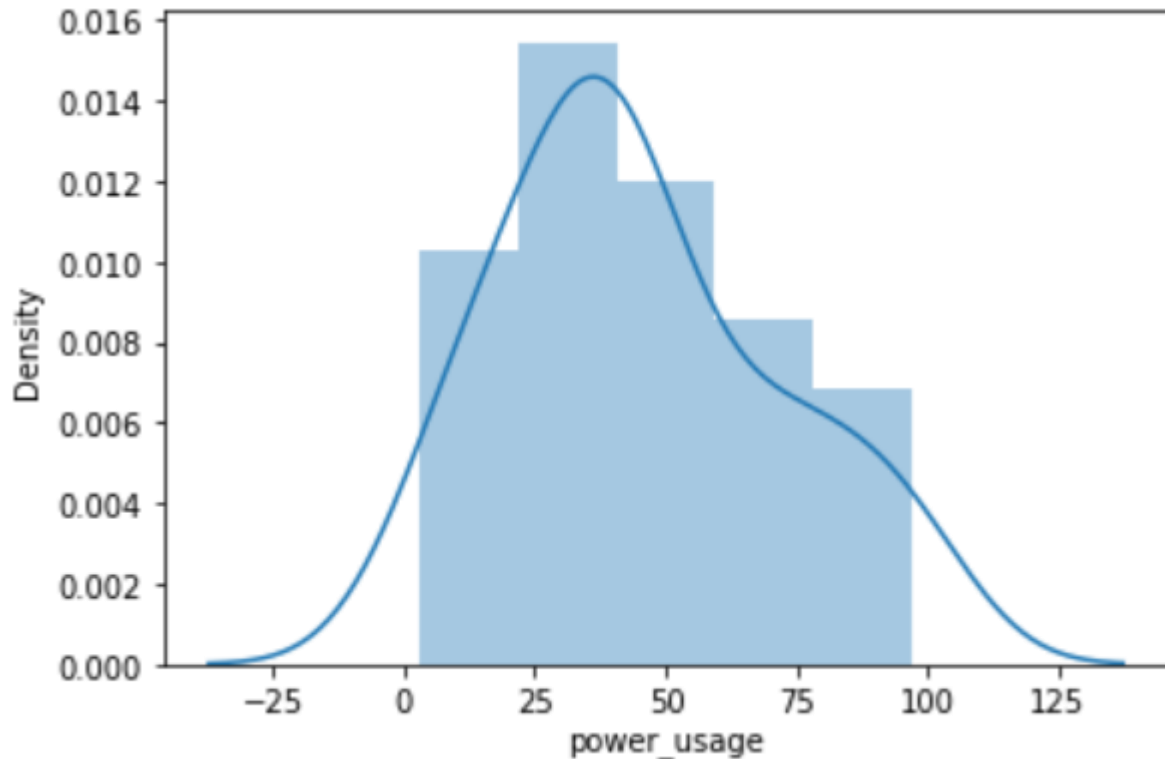
# 씨본(Seaborn)

## ■ 라이브러리 импорт

```
import seaborn as sns
```

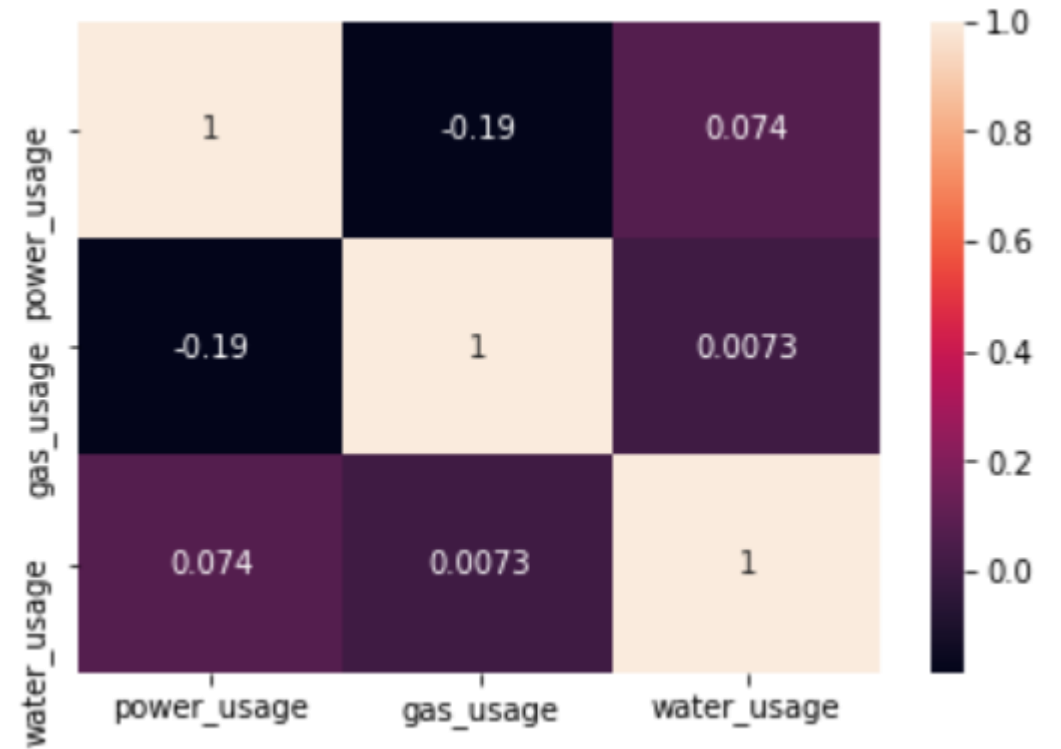
## ■ 분포 플롯

```
sns.distplot(df['power usage'])
```



## ■ 히트맵(상관관계)

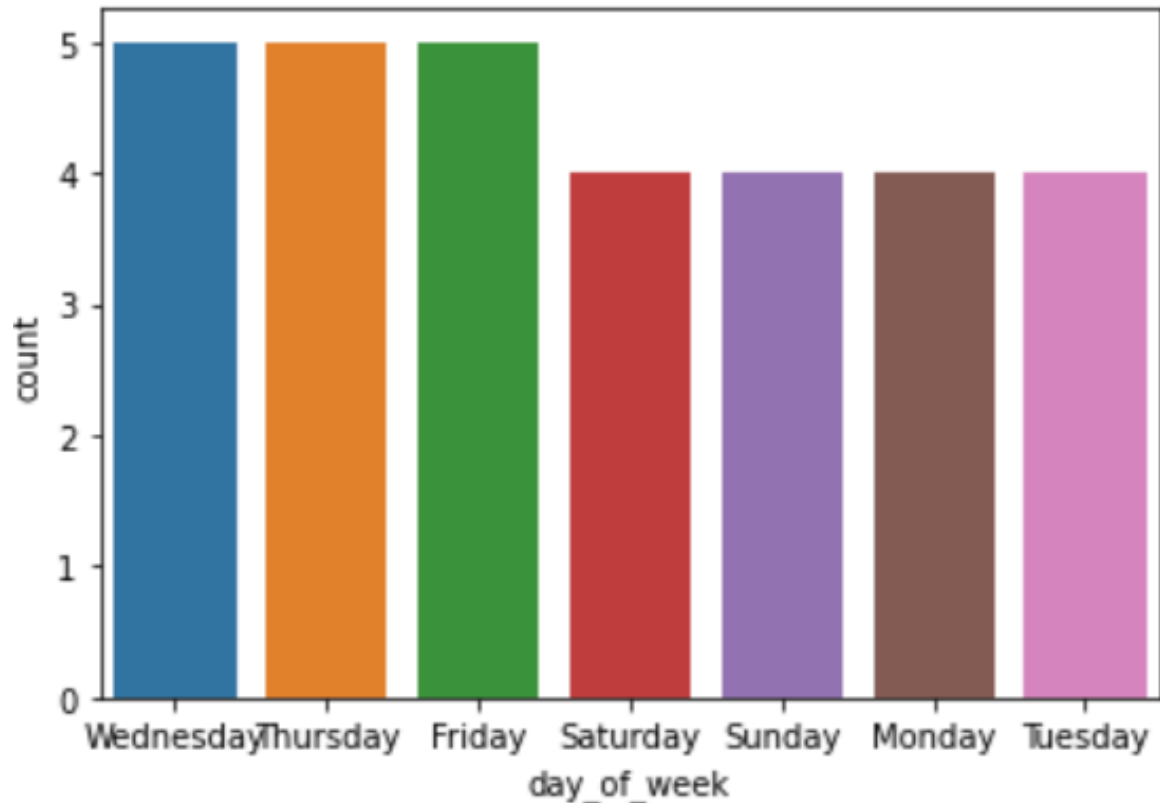
```
sns.heatmap(df.corr(), annot=True)
```



# 씨본(Seaborn)

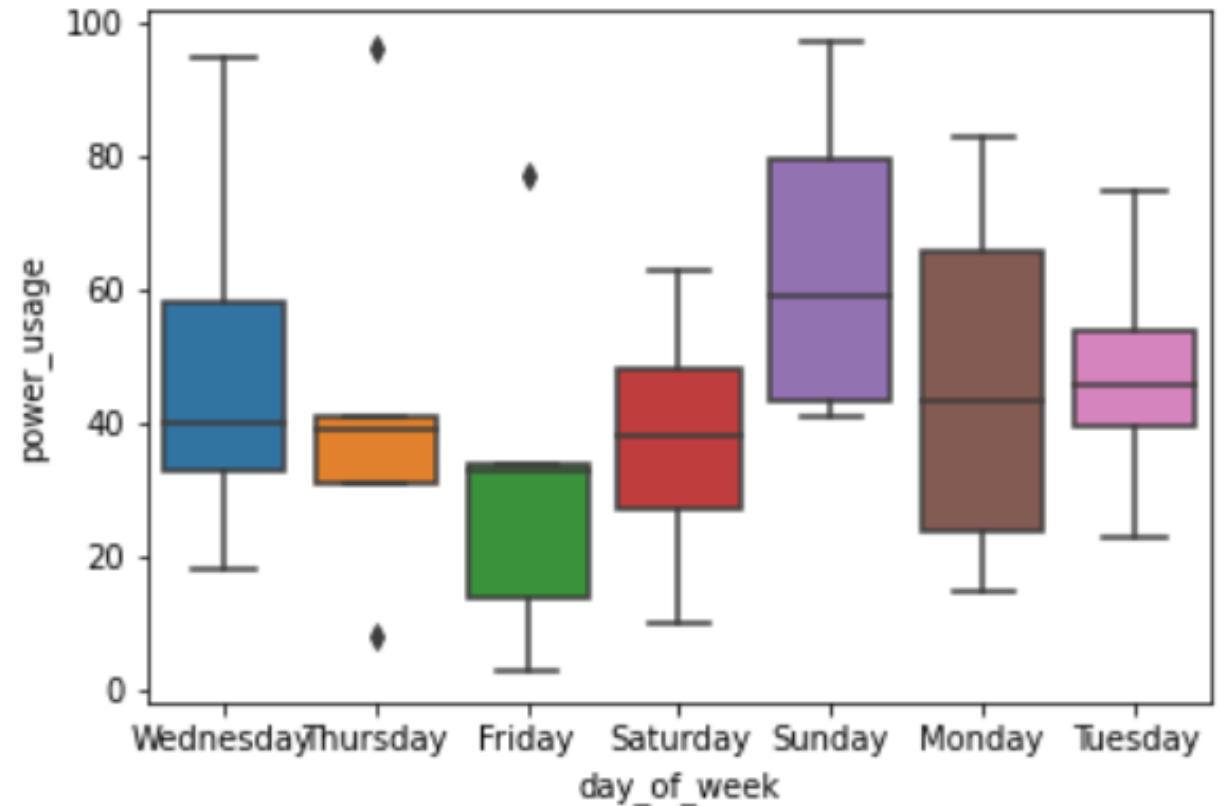
## ■ 카운트 플롯

```
sns.countplot(x='day_of_week',  
              data=df)
```



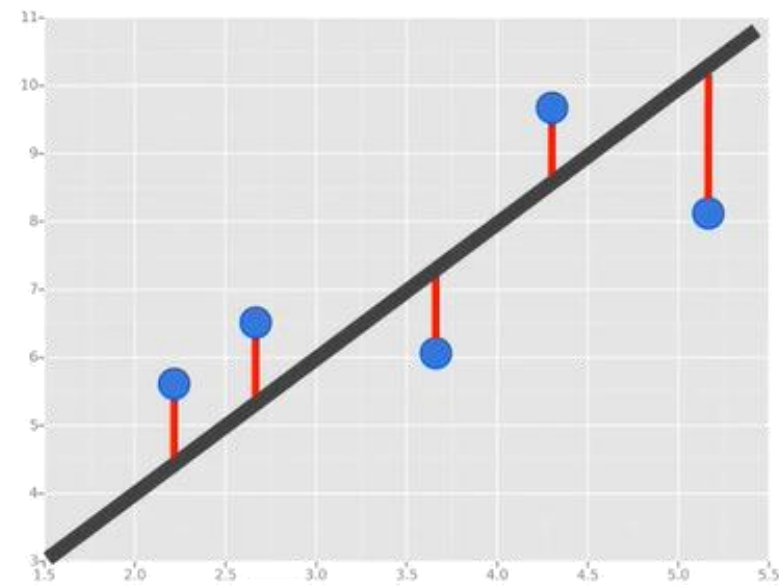
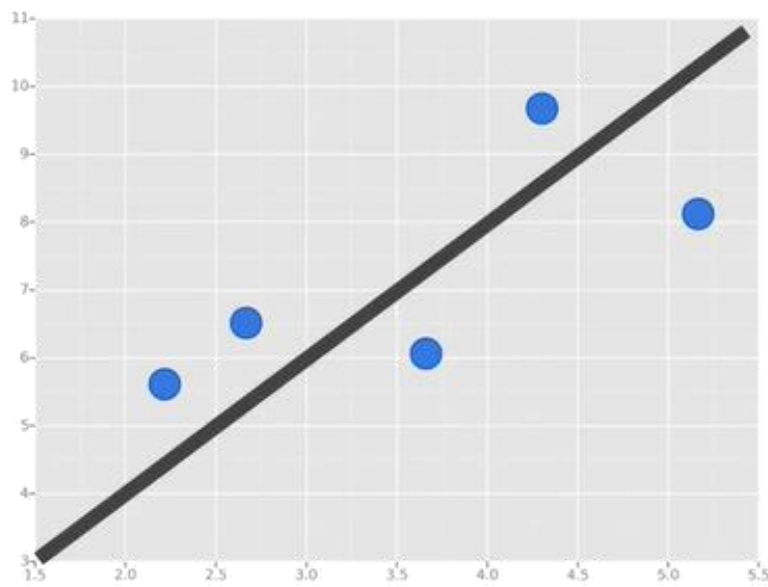
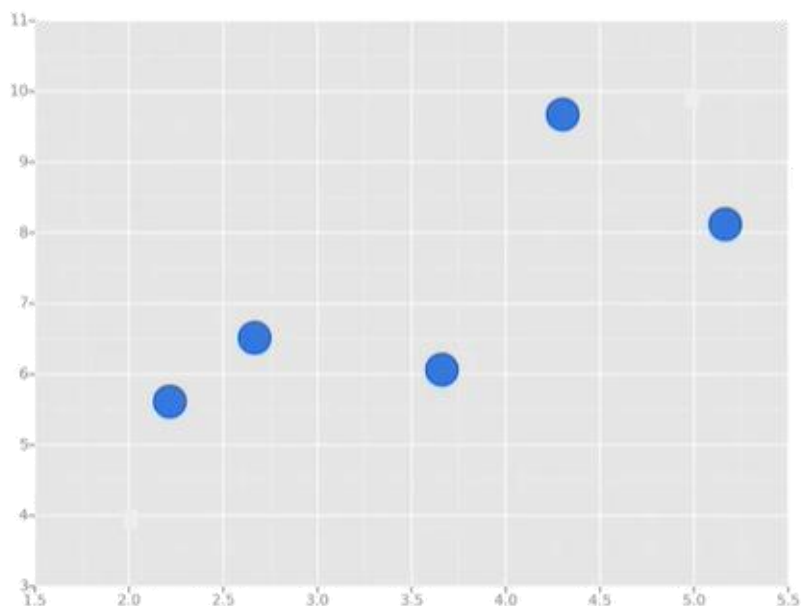
## ■ 박스 플롯

```
sns.boxplot(x='day_of_week',  
            y='power_usage', data=df)
```



# 선형 회귀

종속 변수  $y$ 와 한 개 이상의 독립 변수  $x$ 와의 선형 상관 관계를 모델링 하는 회귀분석 기법



# 선형 회귀

```
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt

# 생산량
output = [110, 125, 140, 145, 160, 166, 179, 190, 200, 215, 230, 250]

# 전력사용량
power_usage = [98, 115, 120, 136, 140, 156, 160, 177, 185, 195, 210, 225]

# p-value : 유의 확률, 일반적으로 0.05 미만일 때 유의미
slope, intercept, r_value, p_value, stderr = stats.linregress(output, power_usage)
```



# 선형 회귀

# 생산량 134개일 때 전기사용량 예측

```
product = 134
```

```
print("기울기(slope) : ", slope)
```

```
print("절편(intercept) : ", intercept)
```

```
print("상관계수(r_value) : ", r_value)
```

```
print("유의확률(p_value) : ", p_value)
```

```
print("{}개 => 예측량 {}kWh".format(  
    product, product*slope + intercept))
```

```
plt.scatter(output, power_usage)
```

```
x = np.arange(0, 300)
```

```
y = [(slope*num + intercept) for num in x]
```

```
plt.plot(x, y, 'b', lw=1)
```

```
plt.xlabel("Output(EA)")
```

```
plt.ylabel("Power Usage(kWh)")
```

```
plt.show()
```

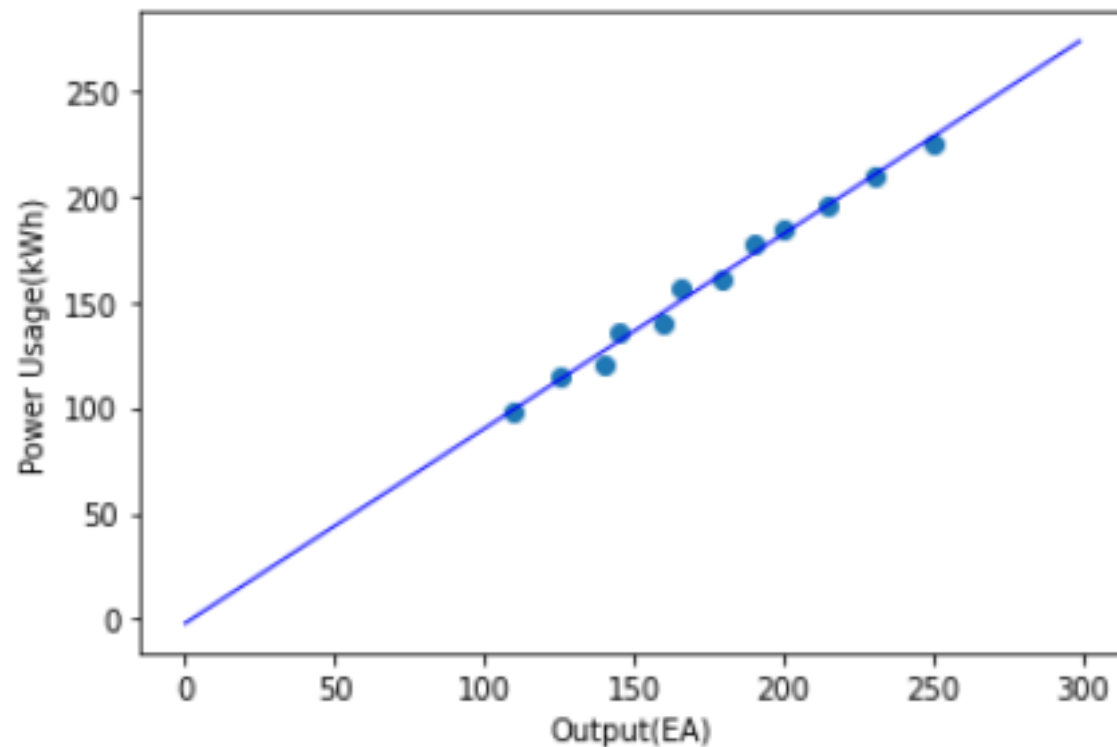
기울기(slope) : 0.9200457304535211

절편(intercept) : -2.024707604744151

상관계수(r\_value) : 0.9950415352828844

유의확률(p\_value) : 2.3409613797567155e-11

134개 => 예측량 121.26142027602768kWh



# 에너지 데이터 분석 실습



energy\_data\_analysis.ipynb

charts.ipynb

# 에너지 데이터 분석 실습

ChatGPT

chat.openai.com/?model=text-davinci-002-render-sha

+ New chat

Today

에너지 데이터 분석

Upgrade to Plus

Danny Park

파이썬 Pandas 사용법을 알려줘.

Save & Submit

Cancel

4. 데이터 불러오기:

- CSV 파일로부터 데이터 불러오기:

```
python
df = pd.read_csv('data.csv')
```

5. 데이터 조작:

- 열(column) 선택:

```
python
ages = df['나이']
```
- 행(row) 선택:

```
python
row = df.loc[0] # 인덱스를 사용한 선택
```
- 조건에 맞는 행 선택:

```
python
adults = df[df['나이'] >= 18]
```

6. 데이터 변경:

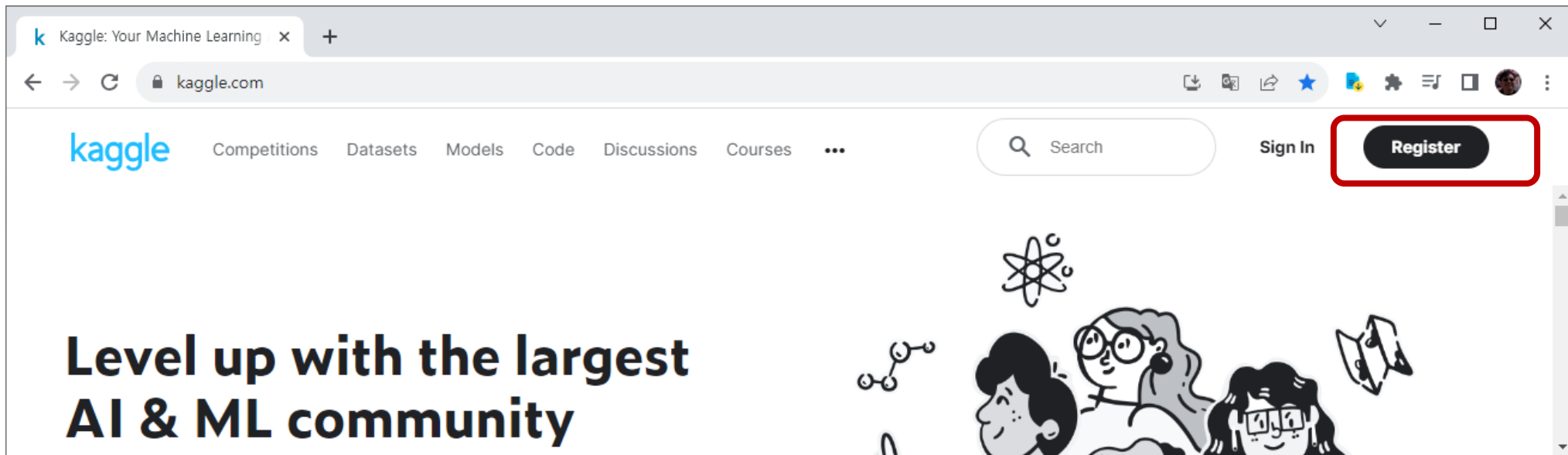
- 열 추가:

Regenerate

# Kaggle 사용

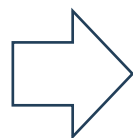
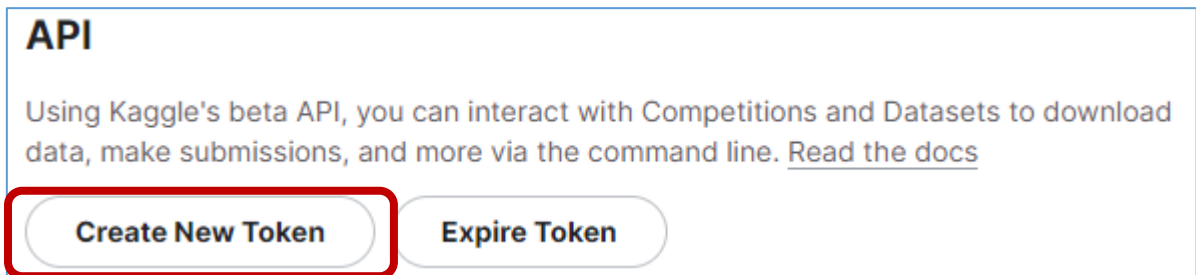
## ■ kaggle 가입

<https://www.kaggle.com/>



## ■ kaggle.json 다운로드

- <https://www.kaggle.com/settings> → API



Ensure kaggle.json is in the location ~/.kaggle/kaggle.json to use the API.

# Kaggle 사용

## ■ kaggle Phone verification

- kggle.json 다운로드 : <https://www.kaggle.com/settings>

→ Phone verification

## Settings

Control over your Kaggle account and all communications

Account Notifications

### Your email address

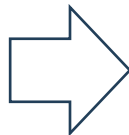
kgpark88@naver.com

Change email

### Phone verification

Your account is not verified. Verifying your account with a phone number allows you to do more on Kaggle, and helps prevent spam and other abuse.

Phone verify



### Just one thing—first verify your account

Enter your phone number and we'll send you a code

COUNTRY  
KR +82 KR

PHONE NUMBER  
1012341234

✓ 로봇이 아닙니다.

reCAPTCHA  
개인정보 보호 · 약관

Already have a code? [Enter it now.](#)

Cancel Send verification code

Verifying with a phone number helps us prevent spam and fraud on Kaggle.

- You can only have one Kaggle account. If you have another account, you'll need to first delete that account.
- The phone number needs to be yours, and not a public or shared number.
- Message and data rates apply.

Contact us for help >

Terms and Conditions · Privacy



### Please read the competition rules

By clicking on the "I Understand and Accept" button below, you agree to be bound by the competition rules for ASHRAE - Great Energy Predictor III.

#### One account per participant

You cannot sign up to Kaggle from multiple accounts and therefore you cannot submit from multiple accounts.

#### Use of Automated Machine Learning Tools (AMLT)

As further described in the Rules, this Competition permits the use of automated machine learning tool(s) ("AMLT") in the creation of Submissions. AMLT Teams (as defined in the Rules) are not eligible to win any prizes.

#### No private sharing outside teams

Privately sharing code or data outside of teams is not permitted. It's okay to share code if

I Understand and Accept

# 에너지 사용량 분석 실습

## ■ Competition 참여

- <https://www.kaggle.com/competitions/ashrae-energy-prediction/data>

The screenshot shows the 'Data Explorer' section of the Kaggle competition page. On the left, the file 'building\_metadata.csv' (45.53 kB) is listed. Below it, the 'Competition Rules' section features a laptop icon with a checkmark and a message: 'To see this data you need to agree to the competition rules. Join the competition to view the data.' A red box highlights the 'Join the competition' button. On the right, the 'Data Explorer' sidebar shows a list of files: 'building\_metadata.csv', 'sample\_submission.csv', 'test.csv', 'train.csv', 'weather\_test.csv', and 'weather\_train.csv'. Below this, a 'Summary' section indicates '6 files' and '34 columns'.

## ■ 데이터셋

- <https://www.kaggle.com/code/sudalairajkumar/simple-exploration-notebook-ashrae/data>

The screenshot shows the 'Data Sources' section of a Kaggle code notebook. On the left, the file 'building\_metadata.csv' (45.53 kB) is listed. A red box highlights the download icon (a downward arrow) and the right-pointing chevron icon. On the right, the 'Input (2.61 GB)' section shows a list of data sources. A red box highlights the 'Data Sources' section, which includes 'ASHRAE - Great Energy P' and a list of files: 'building\_metadata.csv', 'sample\_submission.csv', and 'test.csv'.

# 에너지 사용량 분석 실습



energy\_data\_exploration.ipynb

## ■ kaggle 패키지 설치

```
!pip install kaggle
```

## ■ kaggle API 키 업로드

```
from google.colab import files
```

```
files.upload()
```

```
!mkdir ~/.kaggle
```

```
!cp kaggle.json ~/.kaggle/
```

```
!chmod 600 ~/.kaggle/kaggle.json
```

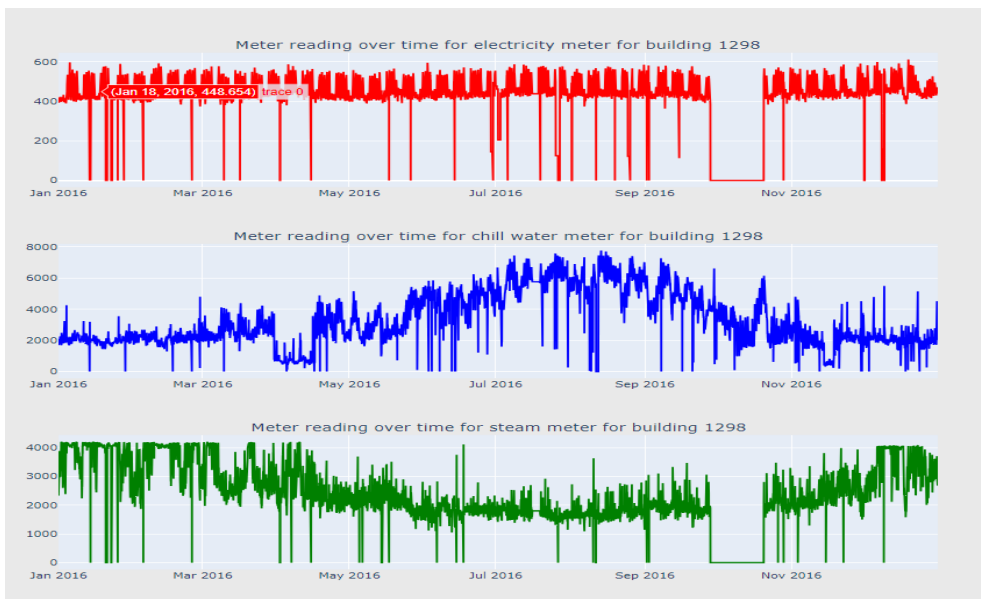
## ■ kaggle 패키지 설치

```
!kaggle competitions download -c ashrae-energy-prediction
```

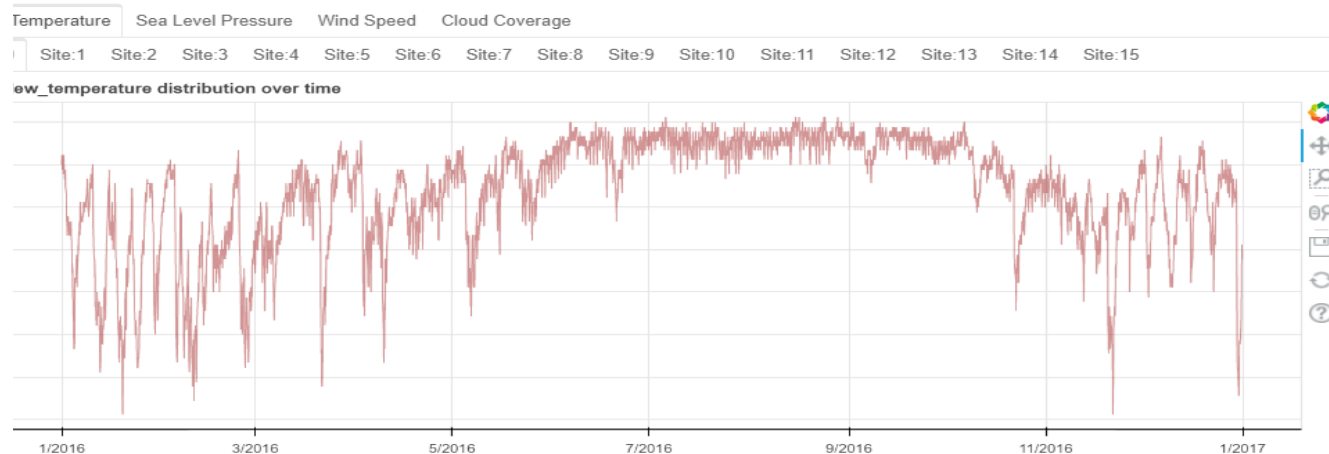
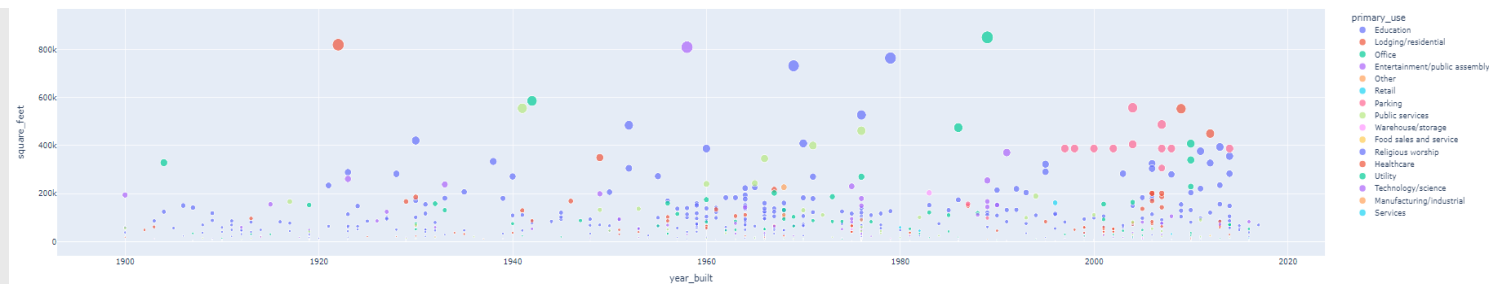
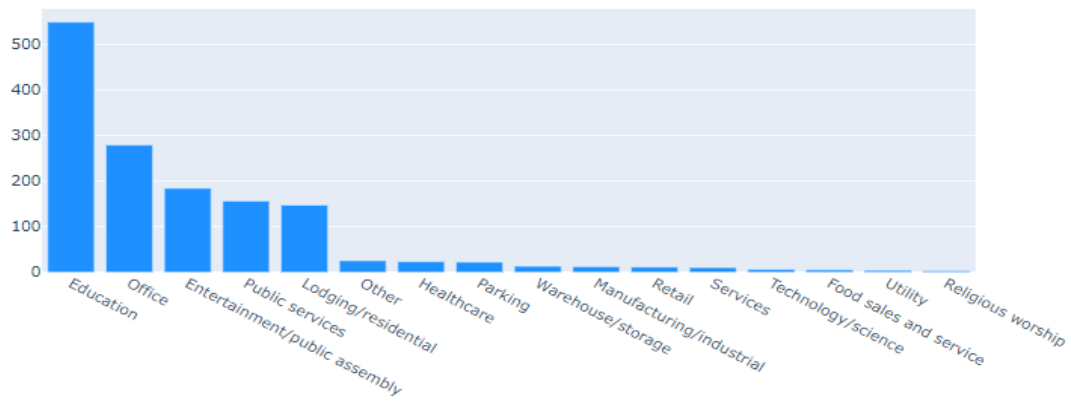
# 에너지 사용량 분석 실습



energy\_data\_exploration.ipynb

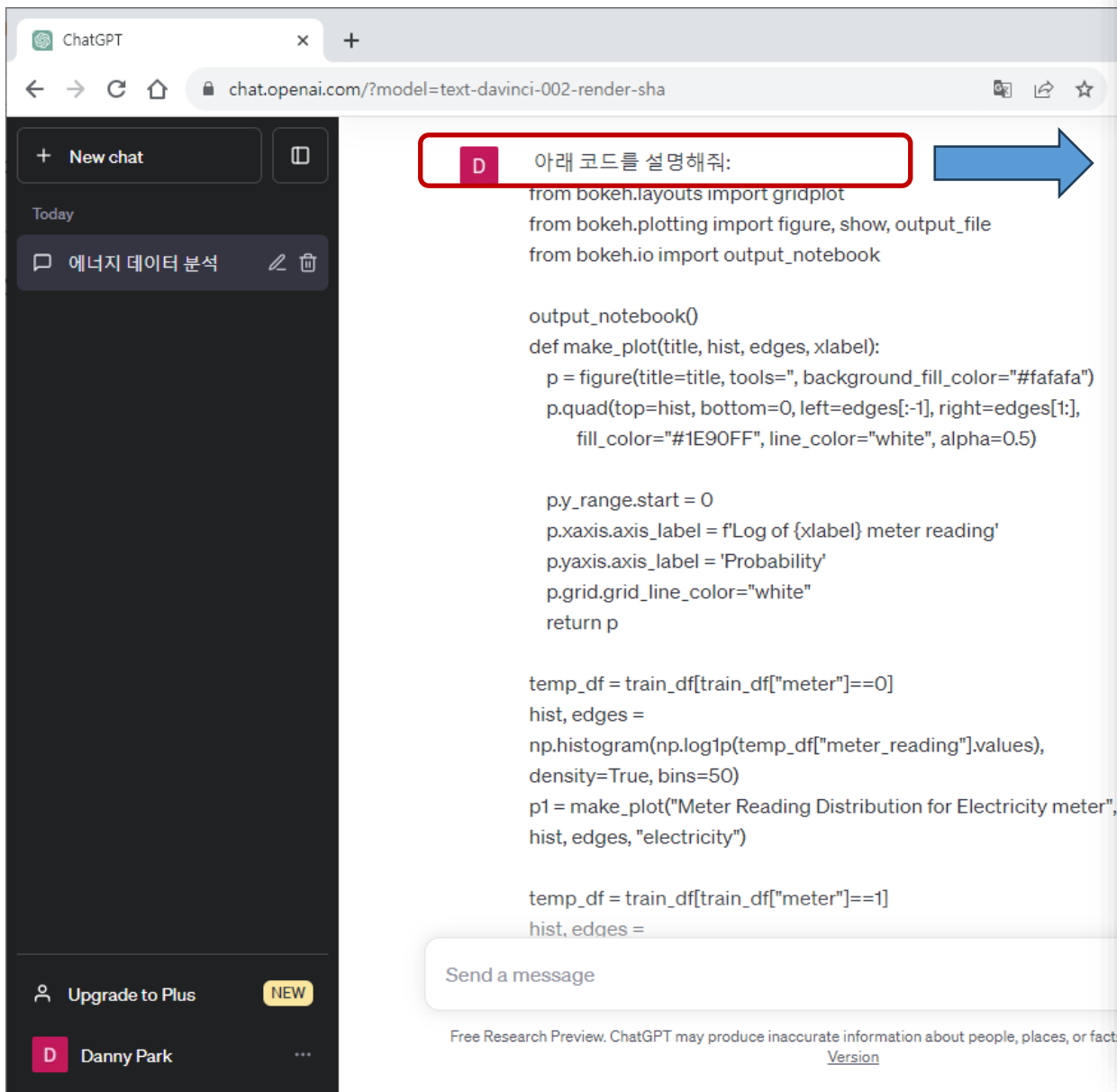


Primary use of Buildings - Count





# 에너지 사용량 분석 실습



ChatGPT

chat.openai.com/?model=text-davinci-002-render-sha

+ New chat

Today

에너지 데이터 분석

아래 코드를 설명해줘:

```
from bokeh.layouts import gridplot
from bokeh.plotting import figure, show, output_file
from bokeh.io import output_notebook

output_notebook()
def make_plot(title, hist, edges, xlabel):
    p = figure(title=title, tools="", background_fill_color="#fafafa")
    p.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],
           fill_color="#1E90FF", line_color="white", alpha=0.5)

    p.y_range.start = 0
    p.xaxis.axis_label = f'Log of {xlabel} meter reading'
    p.yaxis.axis_label = 'Probability'
    p.grid.grid_line_color="white"
    return p

temp_df = train_df[train_df["meter"]==0]
hist, edges =
np.histogram(np.log1p(temp_df["meter_reading"].values),
density=True, bins=50)
p1 = make_plot("Meter Reading Distribution for Electricity meter",
hist, edges, "electricity")

temp_df = train_df[train_df["meter"]==1]
hist, edges =
```

Send a message

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. [Version](#)



이 코드는 Bokeh 라이브러리를 사용하여 시각화를 생성하는데 관련된 것으로 보입니다. Bokeh는 대화형 시각화 라이브러리로, 웹 브라우저에서 그래프와 플롯을 생성하고 상호 작용할 수 있는 기능을 제공합니다. 코드의 내용을 하나씩 살펴보겠습니다.

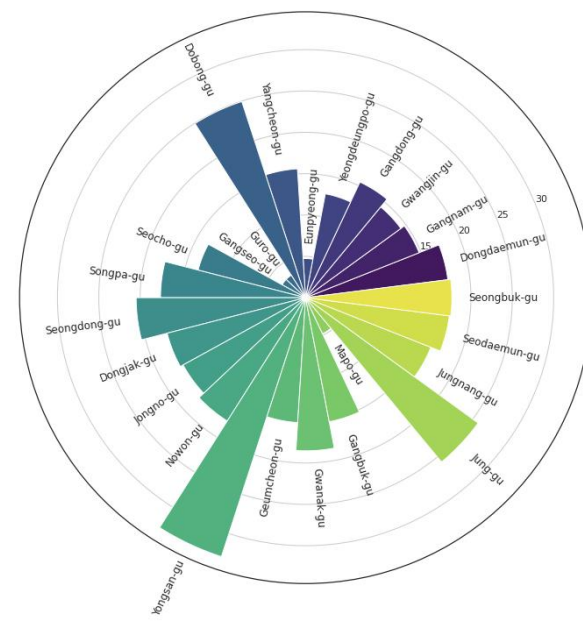
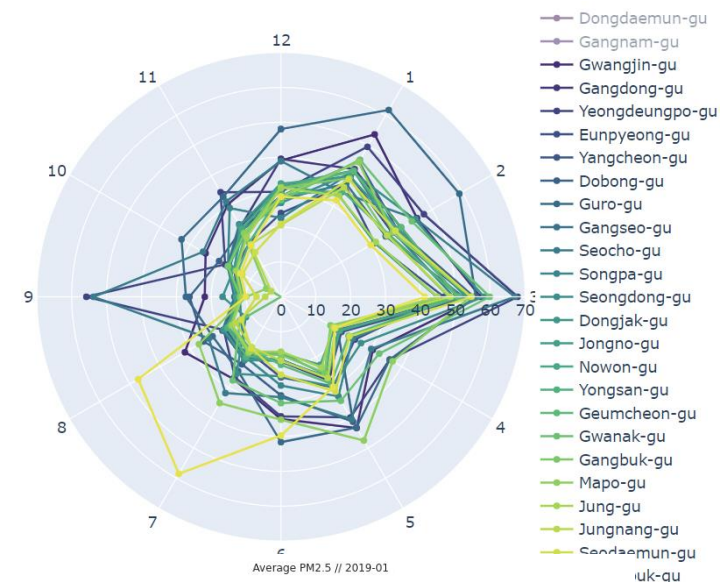
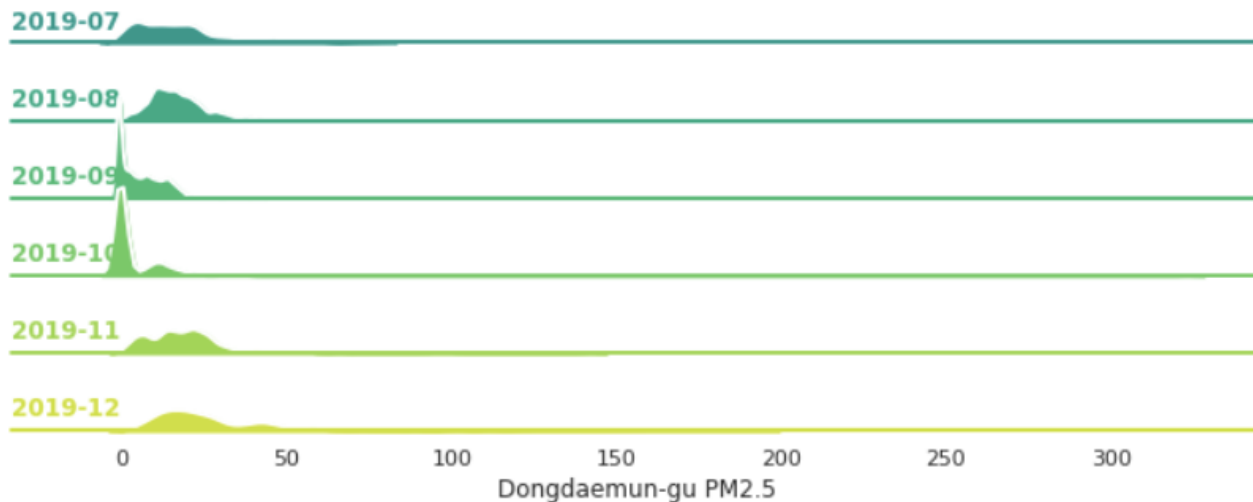
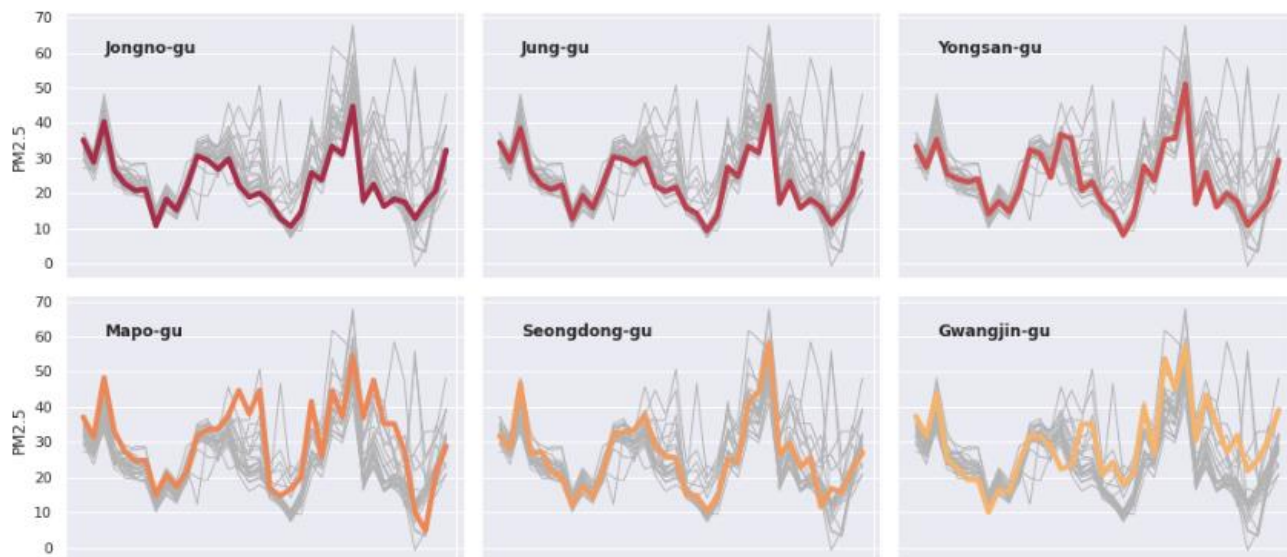
1. `from bokeh.layouts import gridplot`: Bokeh의 `gridplot` 레이아웃을 사용하기 위해 필요한 모듈을 가져옵니다.
2. `from bokeh.plotting import figure, show, output_file`: 그래프 생성을 위한 `figure`와 시각화를 보여주는 `show` 함수, 그리고 출력 파일을 생성하기 위한 `output_file` 함수를 가져옵니다.
3. `from bokeh.io import output_notebook`: Jupyter Notebook에서 Bokeh 시각화를 인라인으로 보여주기 위한 함수를 가져옵니다.
4. `output_notebook()`: Jupyter Notebook에서 Bokeh 플롯을 인라인으로 보여줍니다.
5. `make_plot(title, hist, edges, xlabel)`: 시각화를 생성하는 함수입니다. `title`은 그래프 제목, `hist`는 히스토그램 데이터, `edges`는 히스토그램의 구간 경계, `xlabel`은 x축 레이블을 의미합니다. 함수는 Bokeh의 `figure` 객체를 생성하고 히스토그램 데이터를 기반으로 그래프를 그립니다. 그래프의 스타일, 레이블 등을 설정한 후 `figure` 객체를 반환합니다.
6. 다음 코드 블록은 `train_df` 데이터프레임에서 각각 다른 `meter` 값에 따라 미터별 사용량 분포를 시각화합니다. 각각의 미터 유형에 대해 `hist` (히스토그램 데이터)와 `edges` (구간 경계)를 계산하고, 해당 데이터를 `make_plot` 함수를 사용하여 그래프로 그립니다.
7. 마지막으로 `gridplot`을 사용하여 각 그래프를 그리드 형태로 배열하고, `show` 함수를 호출하여 그리드를 보여줍니다. `gridplot`은 여러 개의 그래프를 원하는 형태로 배열하는데 사용됩니다. `ncols`는 열의 개수, `width`와 `height`는 각 그래프의 크기를 지정하는 옵션입니다.

이 코드는 Bokeh를 사용하여 서로 다른 미터 유형의 사용량 분포를 시각화하고 그것들을 그리드로 배열하여 한 번에 비교할 수 있도록 하는 것입니다.

# 다중 시계열 데이터 분석 시각화 실습



multiple\_time\_series\_data\_analysis.ipynb



# 노코드(No Code) 데이터 분석



<https://docs.kanaries.net/ko/pygwalker/use-pygwalker-with-streamlit>

## ■ 패키지 설치

```
pip install pandas
pip install pygwalker
pip install streamlit
```

## ■ pygwalker\_demo.py

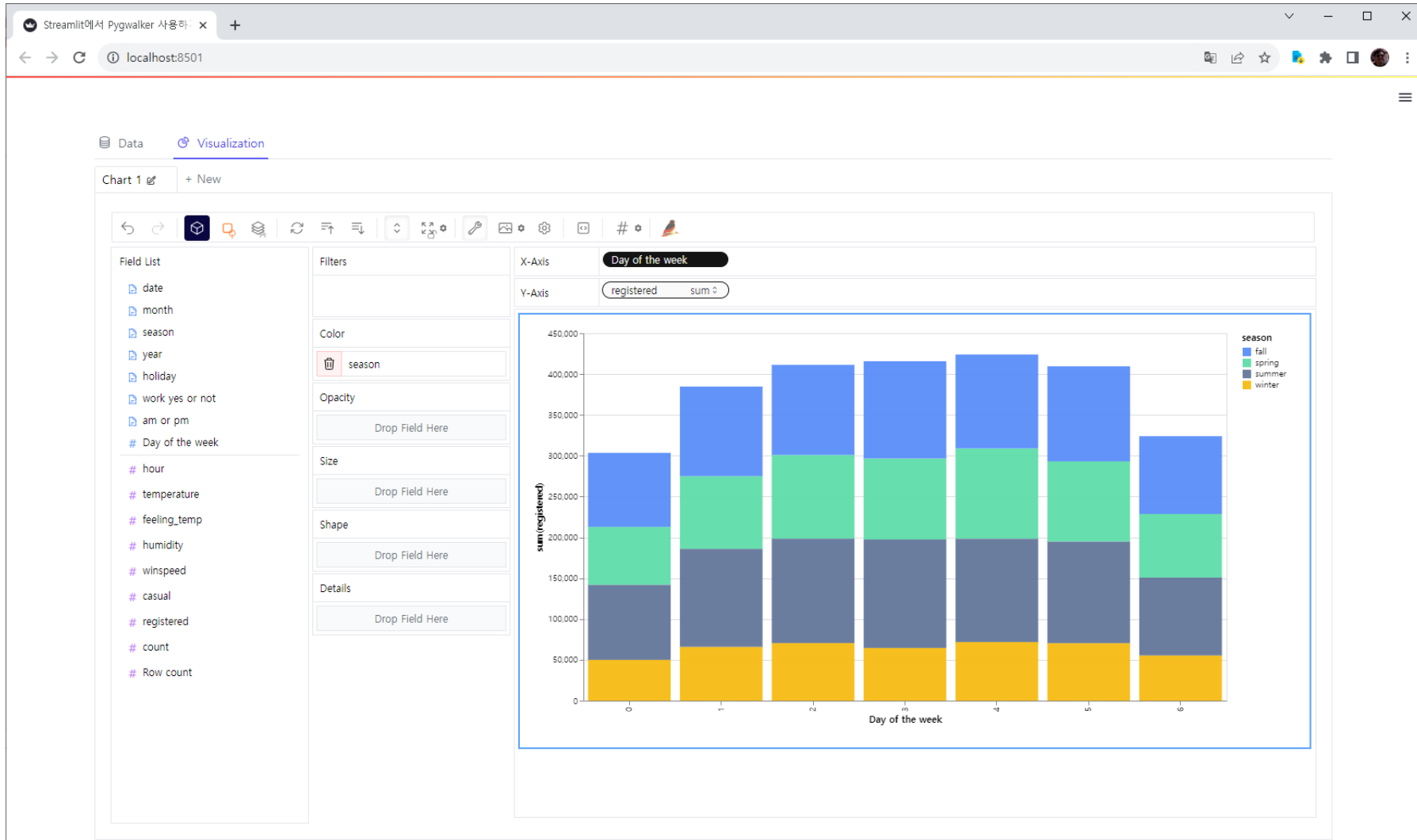
```
1  import pygwalker as pyg
2  import pandas as pd
3  import streamlit.components.v1 as components
4  import streamlit as st
5
6  # Streamlit 페이지의 너비 조정
7  st.set_page_config(
8      page_title="Streamlit에서 Pygwalker 사용하기",
9      layout="wide"
10 )
11
12 # 제목 추가
13 st.title("Streamlit에서 Pygwalker 사용하기")
14
15 # 데이터 가져오기
16 df = pd.read_csv("public-datasets/bike_sharing_dc.csv")
17
18 # Pygwalker를 사용하여 HTML 생성
19 pyg_html = pyg.walk(df, return_html=True)
20
21 # Streamlit 앱에 HTML 임베드
22 components.html(pyg_html, height=1000, scrolling=True)
```

## ■ 실행

```
streamlit run pygwalker_demo.py
```

# 노코드(No Code) 데이터 분석

■ 실행 `streamlit run pygwalker_demo.py`



THANK YOU

[kgpark88@gmail.com](mailto:kgpark88@gmail.com)