



시각화 코드(파이썬) 리액트로 구현하는 법

👤 생성자	mejin
🕒 생성 일시	@2025년 10월 10일 오후 5:22
☰ 카테고리	웹 개발
👤 최종 편집자:	민서 곽
🕒 최종 업데이트 시간	@2025년 10월 10일 오후 7:27

요약

주피터 또는 코랩에서 JSON파일을 만들어서 저장하고 리액트에서는 /public에서 export 해서 사용한다.

흐름

1. 주피터에서 데이터 분석 + 시각화용 데이터 생성
 - a. 히스토그램 bin 결과, 상관계수 행렬
2. 그 결과를 json파일로 저장 (주피터에서)
 - a. hist.json, corr.json 같은 파일이 만들어짐
3. 리액트 웹앱의 /public 폴더에 복사
 - a. `fetch` 로 불러와 `Plotly` 등으로 시각화

상세

◆ 1. 히스토그램 bin 결과 생성하기

파이썬에서는 `numpy.histogram()`을 이용해, 히스토그램을 구성하는 **bin 구간** **경계값**과 **각 구간의 데이터 개수**를 얻을 수 있습니다.

```
import numpy as np
import json

# 예시 데이터 (pH 센서 데이터)
data = np.random.normal(7, 0.4, 1000) # 평균 7, 표준편차 0.4

# 히스토그램 계산 (30개의 구간)
counts, bin_edges = np.histogram(data, bins=30)

# 결과 확인
print("counts:", counts[:5])
print("bin_edges:", bin_edges[:6])
```

출력 예시:

```
counts: [ 1  2 10 23 50]
bin_edges: [5.8 5.9 6.0 6.1 6.2 6.3]
```

◆ 2. 상관관계수 행렬(correlation matrix) 생성하기

```
import pandas as pd

# 예시 데이터프레임
df = pd.DataFrame({
    "pH": np.random.normal(7, 0.4, 1000),
    "Temp": np.random.normal(40, 5, 1000),
    "Current": np.random.normal(10, 1.5, 1000)
})

corr = df.corr() # 상관관계수 행렬
print(corr)
```

출력 예시:

	pH	Temp	Current
pH	1.00	-0.05	0.02
Temp	-0.05	1.00	0.10
Current	0.02	0.10	1.00

◆ 3. JSON 파일로 저장하기 (Jupyter에서 실행)

이제 위 두 결과를 **JSON** 형태로 내보내면 됩니다.

리액트가 읽을 수 있게 dict → json.dump()로 변환합니다.

```
# 히스토그램용 JSON 저장
hist_payload = {
    "title": "pH Histogram",
    "counts": counts.tolist(),
    "bin_edges": bin_edges.tolist(),
    "xlabel": "pH",
    "ylabel": "Count"
}

with open("hist.json", "w", encoding="utf-8") as f:
    json.dump(hist_payload, f, ensure_ascii=False, indent=2)

# 상관계수용 JSON 저장
corr_payload = {
    "title": "Correlation Heatmap",
    "z": corr.values.tolist(),
    "x": corr.columns.tolist(),
    "y": corr.index.tolist()
}

with open("corr.json", "w", encoding="utf-8") as f:
    json.dump(corr_payload, f, ensure_ascii=False, indent=2)
```

✅ 이렇게 하면

hist.json

corr.json

두 파일이 주피터 노트북 실행 경로에 생성됩니다.

◆ 4. 생성된 파일 확인 및 이동

- 주피터 오른쪽 파일 브라우저(📁 아이콘)에서 hist.json, corr.json이 보일 거예요.
- 클릭 → "Download" 해서 리액트 프로젝트의 public/ 폴더에 넣으세요.

```
my-react-app/  
├── public/  
│   ├── hist.json  
│   ├── corr.json  
│   └── index.html  
├── src/  
│   ├── App.js  
│   └── Dashboard.jsx ← 시각화 코드
```

◆ 5. 웹에서 불러오기

리액트에서는 이렇게 하면 됩니다 📌

```
useEffect(() => {  
  fetch("/hist.json")  
    .then(res => res.json())  
    .then(data => setHist(data));  
}, []);
```

즉, 그래프를 그릴 데이터 수치를 json으로 내보내는 단계를 추가해야 한다.

기존 코드 예시로 수정 방향

```

# 숫자컬럼 히스토그램
num_cols = [c for c in ["ph","temp","current"] if c in dd.columns]
if num_cols:
    dd[num_cols].hist(bins=30, figsize=(10,6))
    plt.tight_layout(); plt.show()

# 상관관계
if len(num_cols) >= 2:
    corr = dd[num_cols].corr()
    plt.figure(figsize=(4,3))
    sns.heatmap(corr, annot=True, fmt=".2f")
    plt.title("Correlation"); plt.show()

```

```

import numpy as np, json, seaborn as sns, matplotlib.pyplot as plt

# 숫자형 컬럼 선택
num_cols = [c for c in ["ph","temp","current"] if c in dd.columns]

if num_cols:
    # (1) 히스토그램 시각화
    dd[num_cols].hist(bins=30, figsize=(10,6))
    plt.tight_layout(); plt.show()

    # (2) 히스토그램 데이터 → JSON 저장
    histograms = {}
    for col in num_cols:
        counts, bin_edges = np.histogram(dd[col].dropna(), bins=30)
        histograms[col] = {
            "title": f"{col} Histogram",
            "counts": counts.tolist(),
            "bin_edges": bin_edges.tolist(),
            "xlabel": col,
            "ylabel": "Count"
        }

    with open("hist.json", "w", encoding="utf-8") as f:

```

```

    json.dump(histograms, f, ensure_ascii=False, indent=2)
    print("✅ hist.json 저장 완료")

# (3) 상관계수 시각화
if len(num_cols) >= 2:
    corr = dd[num_cols].corr()

    plt.figure(figsize=(4,3))
    sns.heatmap(corr, annot=True, fmt=".2f")
    plt.title("Correlation"); plt.show()

# (4) 상관계수 행렬 → JSON 저장
corr_payload = {
    "title": "Correlation Heatmap",
    "z": corr.values.tolist(),    # 값 (2D 배열)
    "x": corr.columns.tolist(),  # 열 이름
    "y": corr.index.tolist()     # 행 이름
}

with open("corr.json", "w", encoding="utf-8") as f:
    json.dump(corr_payload, f, ensure_ascii=False, indent=2)
    print("✅ corr.json 저장 완료")

```

◆ 실행하면 결과

코랩 / 주피터에서 이걸 실행하면

작업 디렉토리에 다음 두 파일이 생깁니다.

```

hist.json
corr.json

```

- hist.json : 각 컬럼별 히스토그램 bin 데이터
(리액트에서 막대그래프로 그릴 때 사용)
- corr.json : 상관계수 행렬 (히트맵용 데이터)

◆ 이후 단계

1 코랩 왼쪽 파일탐색기에서

→ hist.json, corr.json 우클릭 → “다운로드”

2 리액트 프로젝트의 /public 폴더에 복사

→ /public/hist.json, /public/corr.json


3 리액트 코드에서 `fetch("/hist.json")`, `fetch("/corr.json")`

자동 다운로드

1) Google Colab: `files.download()` 로 자동 다운로드

Colab은 브라우저에서 바로 파일 저장 팝업을 띄울 수 있어요.

```
# (앞에서 hist.json, corr.json을 만든 뒤)

#  Colab 전용 자동 다운로드
try:
    from google.colab import files
    files.download("hist.json")
    files.download("corr.json")
except Exception as e:
    print("Colab이 아니거나 다운로드 중 문제가 발생했습니다:", e)
```

팁: 여러 파일을 한 번에 받으려면 압축해서 한 번만 내려받는 게 편해요.

```
import shutil

# hist.json, corr.json을 하나의 zip으로 묶기
shutil.make_archive("charts_payload", "zip", root_dir=".", base_dir=".")
# → charts_payload.zip 생성

# Colab에서 zip 한 번에 다운로드
```

```
from google.colab import files
files.download("charts_payload.zip")
```

2) Jupyter Notebook/Lab: 링크로 클릭-다운로드 (가장 깔끔)

Jupyter는 보안상 브라우저 자동 저장을 막는 경우가 많아서, **클릭 가능한 링크**를 노트북에 띄워서 받는 게 일반적이에요.

```
# (앞에서 hist.json, corr.json을 만든 뒤)


from IPython.display import FileLink, display, HTML

display(HTML("<h4>다운로드</h4>"))
display(FileLink("hist.json")) # 클릭하면 내려받기/열기
display(FileLink("corr.json"))
```

```
import shutil
from IPython.display import FileLink, display

# charts_payload.zip 생성
shutil.make_archive("charts_payload", "zip", root_dir=".", base_dir=".")
display(FileLink("charts_payload.zip"))
```

JupyterLab/서버 환경이라면:

- 노트북 좌측 파일 브라우저()에서 파일을 오른쪽 클릭 → **Download**
- 또는 위처럼 FileLink를 눌러 다운로드
- SSH 원격 서버일 경우, scp로 가져올 수도 있어요.