

04 PJT

Django에서 Data Science 활용하기

챕터의 포인트

- 목표
- 준비사항
- [실습] Django Template 그래프 출력
- [도전] Django 에서 데이터 사이언스 활용하기
- 제출

목표

| 목표

- Django 에서 데이터 사이언스 패키지 사용하기

django



matplotlib

NumPy

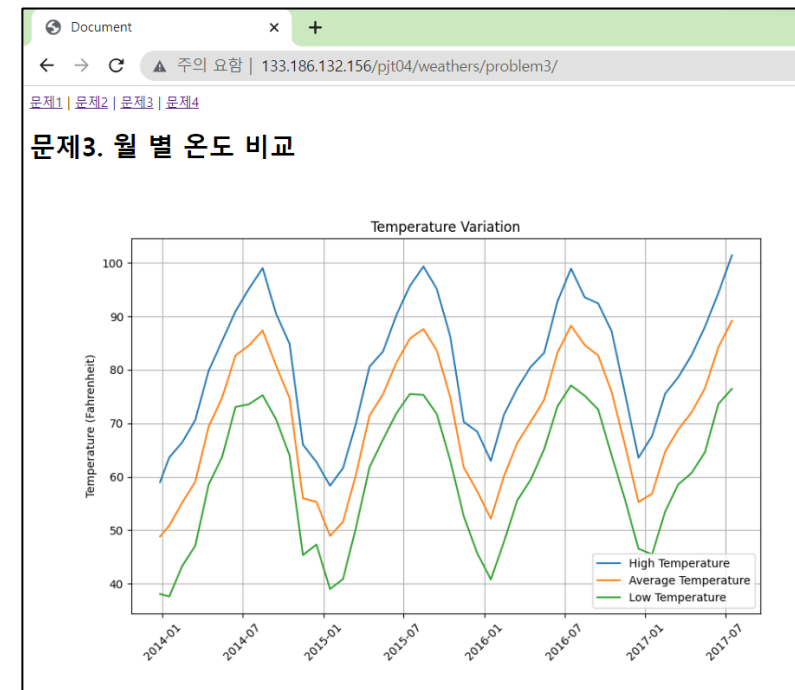


| 이전 프로젝트와의 차이점

- 이전 프로젝트
 - Matplotlib, Pandas, Numpy를 주피터 노트북에서 구동함
- 이번 프로젝트
 - Matplotlib, Pandas, Numpy를 Django 에서 구동함

Django 에서 데이터 사이언스 패키지를 사용하는 이유

- 결과를 웹 페이지에서 보여주기 위함



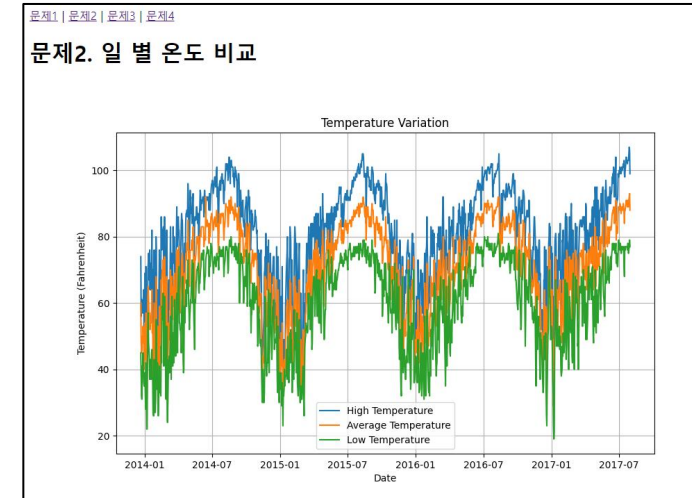
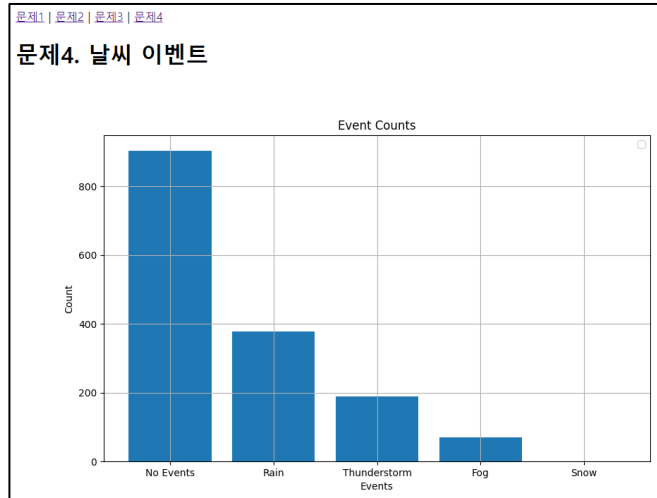
이번 프로젝트 결과 화면 예시

| Django 에서 데이터 사이언스 패키지를 사용하기 위해 알아야 할 내용

- 데이터 사이언스 3종 패키지 사용 방법
- Django 기본 사용 방법
 - 웹 페이지 구성(template)
 - 데이터 전달(View -> Template)
- 파이썬 BytesIO 패키지(곧, 학습 예정)

프로젝트 목표 세 가지

1. Numpy, Pandas, Matplotlib 복습
2. Django Web Framework 복습
3. Django로 데이터 사이언스 패키지 활용 방법 학습



| 학습 순서

1. 강사와 함께 Pandas 와 Matplotlib 를 Django 에서 구현해본다.
2. [도전] 캐글에서 날씨 데이터를 다운로드 받아 Django 에서 데이터 처리 및 시각화

준비사항

| 개발도구

- Visual Studio Code
- Google Chrome
- Python 3.9 +

| 필수 라이브러리

- 가상환경을 설정하여 아래 라이브러리 설치 후 요구사항을 구현합니다.
 - Django 3.2 +
 - Pandas
 - Numpy
 - Matplotlib

Django Template 그래프 출력

| View 에서 Template 으로 이미지 전달하기

- View 에서 Template 으로 이미지 형식의 데이터를 직접 전달할 수 없습니다.
- **저장된 이미지의 경로를 전달**하여 Template 에서 출력해야 합니다.
- matplotlib 의 그래프를 버퍼에 이미지 형식으로 저장 후 저장된 경로를 전달합니다.
 - 버퍼(buffer): 임시로 데이터를 저장하는 공간
- Python “BytesIO” 클래스
 - 파이썬의 내장 모듈인 “io” 모듈에 포함된 클래스
 - 메모리 내에 데이터를 저장 및 조작할 수 있는 기능 제공

View 에서 Template 으로 이미지 전달하기

- views.py

```
import base64
from io import BytesIO

def test(request):
    x = [1, 2, 3, 4, 5]
    y = [1, 2, 3, 4, 5]

    # 그래프 초기화
    plt.clf()

    # plot 생성
    plt.plot(x, y)
    plt.title('Test Graph')
    plt.xlabel('x label')
    plt.ylabel('y label')
    plt.grid(True)

    # 그래프 이미지를 임시로 저장할 버퍼 생성
    buffer = BytesIO()
    # 그래프를 버퍼에 저장. 이미지 형식은 png 로 설정
    plt.savefig(buffer, format='png')
    # 버퍼의 내용을 base64 로 인코딩
    image_base64 = base64.b64encode(buffer.getvalue()).decode('utf-8').replace('\n', '')
    buffer.close()
    context = {
        # 저장된 이미지의 경로를 전달
        'chart_image': f'data:image/png;base64,{image_base64}',
    }

    return render(request, 'test.html', context)
```

- test.html

```
{% extends 'base.html' %}

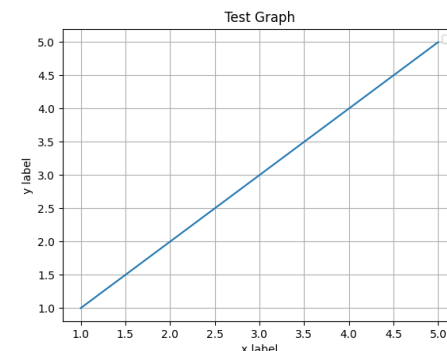
{% block content %}

<h1>Django Template 에서 그래프 출력하기</h1>


{% endblock content %}
```

- 출력 결과 예시

Django Template 에서 그래프 출력하기



Django 에서 데이터 사이언스 활용하기

| 공통 요구사항

- 캐글을 통해 데이터를 다운로드 받아 활용합니다.
 - 데이터셋: [Austin Weather\(텍사스 오스틴 날씨 데이터\)](#)
 - 데이터셋 요약: 2013-12-21 ~ 2017-07-31 까지의 일 별 날씨 데이터 (크기: 20kB)
- Django 프로젝트의 이름은 **mypjt**, 앱 이름은 **weathers** 로 지정합니다.
- .gitignore 파일을 추가하여 불필요한 파일 및 폴더는 제출하지 않도록 합니다.
- 명시된 요구사항 이외에는 자유롭게 작성해도 무관합니다.

| URL

- weathers 앱은 다음 URL 요청에 맞는 역할을 가집니다.

URL 패턴	역할
/weathers/problem1/	다운로드 받은 데이터(.csv) 출력
/weathers/problem2/	일 별 온도 비교를 위한 라인 그래프 출력
/weathers/problem3/	월 별 온도 비교를 위한 라인 그래프 출력
/weathers/problem4/	기상 현상 발생 횟수 히스토그램 출력

| View

- weathers 앱은 다음 역할을 가지는 view 함수를 가집니다.

View Method	역할
problem1	다운로드 받은 데이터(.csv) 를 Pandas DataFrame 형식으로 저장 및 problem1.html 렌더링
problem2	일 별 온도 비교를 위한 라인 그래프 생성 및 problem2.html 렌더링
problem3	월 별 온도 비교를 위한 라인 그래프 생성 및 problem3.html 렌더링
problem4	기상 현상 발생 횟수 히스토그램 생성 및 problem4.html 렌더링

| Templates

- 사용 템플릿 목록
 - A. base.html
 - B. problem1.html
 - C. problem2.html
 - D. problem3.html
 - E. problem4.html

| A. base.html

- 공통 부모 템플릿
 - 모든 템플릿 파일은 base.html 을 상속받아 사용합니다.
 - 다른 파일 템플릿 경로로 이동할 수 있는 링크들을 출력합니다.
- 출력결과 예시

[문제1](#) | [문제2](#) | [문제3](#) | [문제4](#)

B. problem1.html

- Pandas 를 사용하여 캐글에서 다운로드 받은 파일을 DataFrame 으로 읽어옵니다.
- 제공된 스타일을 활용하여 전체 데이터를 표 형태로 출력합니다.
- 출력 결과 예시

Document

127.0.0.1:8000/headers/problem1/

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

→

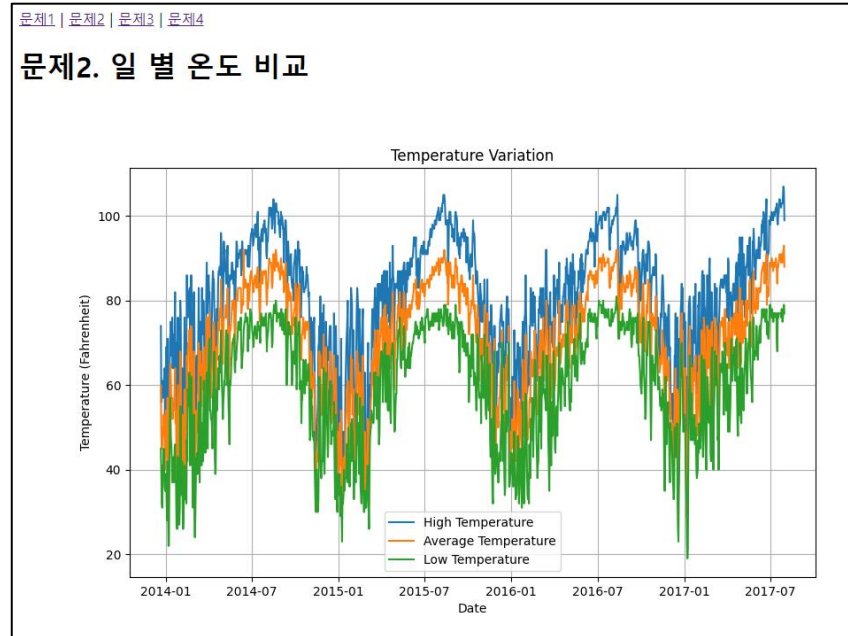
→

→

→

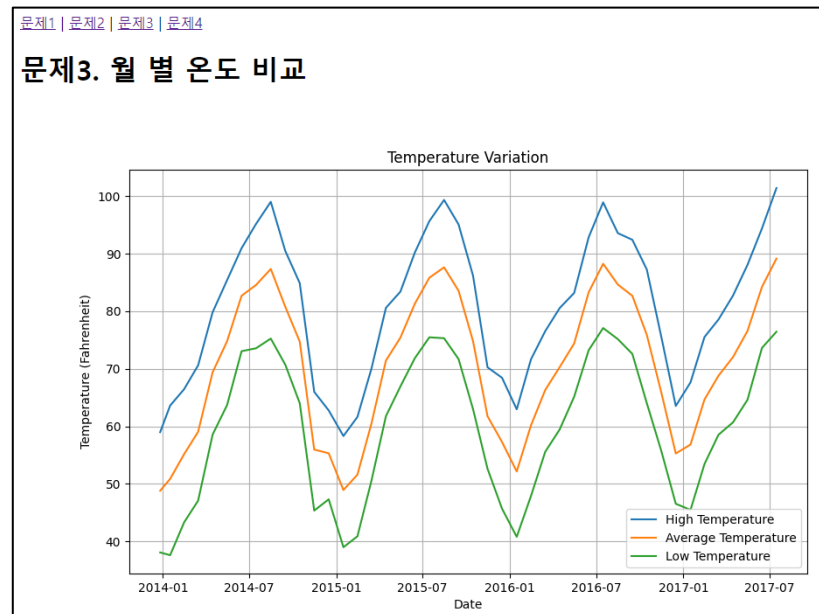
| C. problem2.html

- 일 별 최고, 평균, 최저 온도를 선 그래프로 출력합니다.
 - 날짜 필드: 날짜 형식으로 변환하여 사용합니다.
- 출력 결과 예시



| D. problem3.html

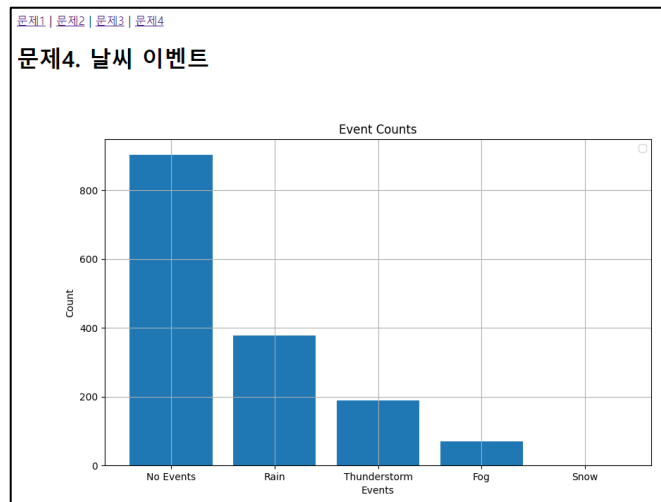
- 월 별 최고, 평균, 최저 온도의 평균을 선 그래프로 시각화 합니다
- 날짜 필드: 날짜 형식으로 변환하여 사용합니다.
- 온도 필드: 평균값 계산을 위해 숫자 형식으로 변환하여 사용합니다.
- 출력 결과 예시



E. problem4.html

- 기상 현상의 발생 횟수(“Events” 컬럼)를 히스토그램으로 출력합니다.
- 사용되는 Events 컬럼은 “결측치” 가 포함되어 있으며, “다중 값을 허용” 합니다.
- 다중 값은 각 기상 현상의 발생 횟수에 추가하여 계산합니다.
 - ex) “Rain , Thunderstorm” = “Rain + 1, Thunderstorm + 1” 로 계산합니다.

- 출력 결과 예시



제출

| 제출 시 주의사항

- 제출기한은 금일 18시까지입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 README.md 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점 등을 상세히 기록하여 제출합니다.
 - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- <https://lab.ssafy.com/> 에 프로젝트를 생성하고 제출합니다.
 - 프로젝트 이름은 '프로젝트 번호 + pjt' 로 지정합니다. (ex. 04_pjt)
- 반드시 각 반 담당 교수님을 Maintainer 로 설정해야 합니다.