



CHAPTER 25

미세먼지 데이터 시각화



핵심 학습 요소

핵심 학습 요소

- matplotlib, bar(), subplot(), DataFrame()



문제 분석



문제 분석

CT

✧ 입력

- 미세먼지 데이터를 초기 값으로 제공

✧ 출력

- 초미세먼지 막대그래프
- 미세먼지 막대그래프



문제 분석

❖ 문제 분해

- 미세먼지와 초미세먼지 데이터 나타내기
- 미세먼지 막대그래프로 나타내기
- 초미세먼지 막대그래프로 나타내기

❖ 사용자 생각하기

- 어떤 색깔을 사용할 것인지 고려하기
- 가로축, 세로축을 어떻게 표시할 지 고려하기

❖ 데이터

- 초미세먼지를 나타내기 위한 정수 배열
- 미세먼지를 나타내기 위한 정수 배열
- 1시부터 10시까지 시간을 나타내기 위한 문자열 리스트

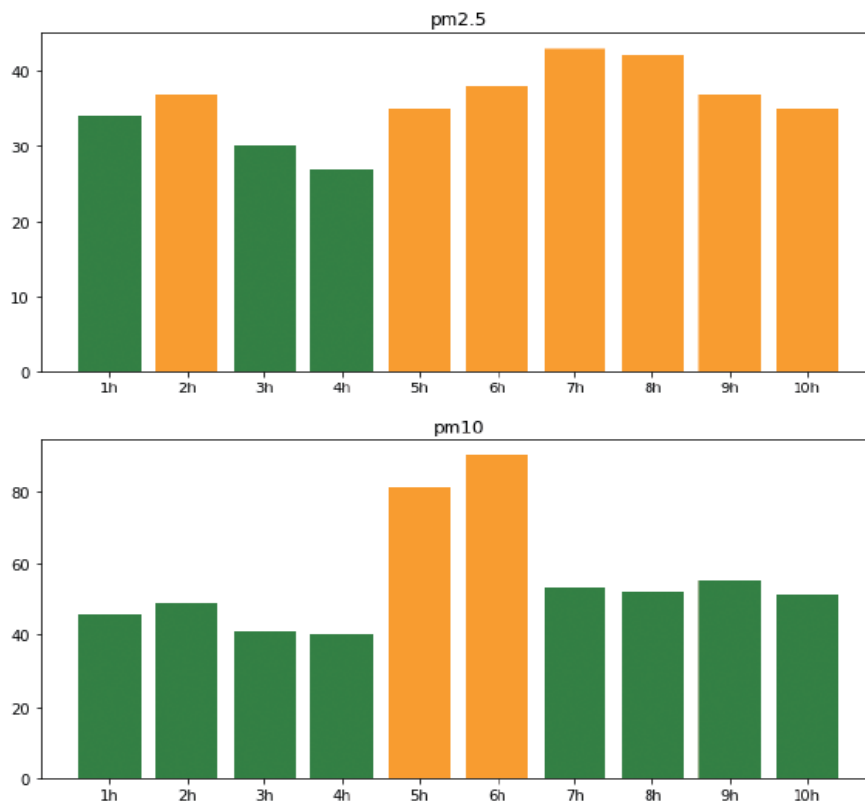
테스트와 디버깅

테스트와 디버깅

CT

출력

확인 및 유의사항



- 예상된 테스트 결과인지 확인한다.
- 원하는 색깔로 막대그래프가 나타나지 않으면 if 문이 제대로 실행되고 있는지 확인한다.

알고리즘

❖ 전체 알고리즘 뼈대





프로그래밍

❖ 변수

- hour: 1시부터 10시까지 문자열 리스트
- pm25: 시간에 따른 초미세먼지 정수형 배열
- pm10: 시간에 따른 미세먼지 정수형 배열



프로그래밍

- numpy 배열과 막대그래프를 그리는 데 필요한 패키지를 import
- 시간, 미세먼지, 초미세먼지 데이터를 저장하는 리스트와 배열 초기화

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
```

```
hour = ['1h', '2h', '3h', '4h', '5h', '6h', '7h', '8h', '9h', '10h']
pm25 = np.array([34, 37, 30, 27, 35, 38, 43, 42, 37, 35])
pm10 = np.array([46, 49, 41, 40, 81, 90, 53, 52, 55, 51])
```




프로그래밍

- 아래 프로그램은 막대그래프 크기와 막대그래프 배치를 정함
- 미세먼지 값에 따라 막대그래프 색깔을 지정
- 범례를 표시하고 막대그래프 제목을 정함

```
plt.figure(figsize=(10,10))
plt.subplot(2,1,1)

for i in range(10):

    if pm25[i] < 15:
        plt.bar(hour[i], pm25[i], color='blue')
    elif 15 <= pm25[i] < 35:
        plt.bar(hour[i], pm25[i], color='green')
    elif 35 <= pm25[i] < 75:
        plt.bar(hour[i], pm25[i], color='orange')
    elif pm25[i] >= 75:
        plt.bar(hour[i], pm25[i], color='red')

plt.title('pm2.5')
```



프로그래밍

- 다음 프로그램은 앞에서 그린 막대그래프 아래에 막대그래프 배치를 정함
- 미세먼지 값에 따라 막대그래프 색깔을 지정
- 막대그래프 제목을 정하고 화면에 표시

```
plt.subplot(2,1,2)

for i in range(10):
    if pm10[i] < 30:
        plt.bar(hour[i], pm10[i], color='blue')
    elif 30 <= pm10[i] < 80:
        plt.bar(hour[i], pm10[i], color='green')
    elif 80 <= pm10[i] < 150:
        plt.bar(hour[i], pm10[i], color='orange')
    elif pm10[i] >= 150:
        plt.bar(hour[i], pm10[i], color='red')

plt.title('pm10')
plt.show()
```



프로그래밍 – 전체 프로그램

- 1~3번째 줄 - 배열을 처리하기 위한 numpy 라이브러리와 그래프 표현을 위한 matplotlib를 import
- 그림의 가로 세로 크기를 설정한 후, 차례대로 초미세먼지 pm25와 미세먼지 pm10을 그래프로 보여줌

```
01 import numpy as np
02 import matplotlib.pyplot as plt
03 from matplotlib import style
04
05
06 hour = ['1h', '2h', '3h', '4h', '5h', '6h', '7h', '8h', '9h', '10h']
07 pm25 = np.array([34, 37, 30, 27, 35, 38, 43, 42, 37, 35])
08 pm10 = np.array([46, 49, 41, 40, 81, 90, 53, 52, 55, 51])
09
10 plt.figure(figsize=(10,10))    # 그림 가로, 세로 크기 설정
11 plt.subplot(2,1,1)             # 2행중 첫번째 행
12
13 for i in range(10):
14
15     if pm25[i] < 15:
16         plt.bar(hour[i], pm25[i], color='blue')
17     elif 15 <= pm25[i] < 35:
18         plt.bar(hour[i], pm25[i], color='green')
19     elif 35 <= pm25[i] < 75:
20         plt.bar(hour[i], pm25[i], color='orange')
21     elif pm25[i] >= 75:
22         plt.bar(hour[i], pm25[i], color='red')
23
```



프로그래밍 – 전체 프로그램

```
24 plt.title('pm2.5')
25
26 plt.subplot(2,1,2)          # 2행중 두번째 행
27
28 for i in range(10):
29     if pm10[i] < 30:
30         plt.bar(hour[i], pm10[i], color='blue')
31     elif 30 <= pm10[i] < 80:
32         plt.bar(hour[i], pm10[i], color='green')
33     elif 80 <= pm10[i] < 150:
34         plt.bar(hour[i], pm10[i], color='orange')
35     elif pm10[i] >= 150:
36         plt.bar(hour[i], pm10[i], color='red')
37
38 plt.title('pm10')
39 plt.show()
```