Final Assignment

1. 지하철 유무임 승차비율 데이터 분석 (50점)

[데이터 로드]

```
import csv  # csv 모듈 불러오기
import numpy as np  # numpy 모듈 불러오기
import matplotlib.pyplot as plt  # matplotlib의 pyplot 서브패키지 불러오기

data = []  # 데이터를 저장할 빈 리스트 생성
with open('Metro_fare.csv', 'r', encoding='cp949') as f:  # 'Metro_fare.csv'을 열고 파일 객체를 'f'로 지정
    source = csv.reader(f, delimiter=',')  # csv.reader 함수를 통해 csv파일 읽어오기
    print(source)  # source 출력
    header = next(source)  # 첫 번째 행을 header로 지정
    for row in source:  # csv 파일에서 한 행씩 읽어와
          data.append(row)  # data 리스트에 저장
```

1) 총 승차인원(유임승차+무임승차)과 총 하차인원(유임하차+무임하차)이 가장 많은 역과 승차인원을 출력하세요. (25점)

```
data_np = np.array(data, dtype = object)  # data를 numpy 배열로 변환/여러 데이터 유형을 저장하기 위해 dtype을 object로 설정
```

```
def change_int(x):  # 숫자로 변환하는 함수 정의
    x = int(x.replace(',',''))  # 쉼표를 제거하고 숫자로 변환
    return x
```

```
max_on = 0  # 최대 승차 인원 수를 저장할 변수 초기화
max_down = 0  # 최대 하차 인원 수를 저장할 변수 초기화

for i in range(len(data_np)):  # 데이터 배열의 각 행에 대해 반복

    for j in range(4, 8):  # 승하차 인원 정보가 있는 열에 대해 반복
        if isinstance(data_np[i][j], str):  # 데이터가 문자열인 경우에만 처리
            data_np[i][j] = change_int(data_np[i][j])  # change_int 함수를 사용해 문자열을 숫자로 변환


    on = data_np[i][4] + data_np[i][6]  # 유임 승차 인원 수와 무임 승차 인원 수를 더해 총 승차 인원 수 계산
    if on > max_on:  # 총 승차 인원 수가 max_on에 저장된 값보다 큰 경우
        max_on = on  # 최대 승차 인원 수 업데이트
        index_on = i  # 최대 승차 인원 수가 기록된 행의 인덱스 저장


    down = data_np[i][5] + data_np[i][7]  # 유임 하차 인원 수와 무임 하차 인원 수를 더해 총 하차 인원 수 계산
    if down > max_down:  # 총 하차 인원 수가 max_down에 저장된 값보다 큰 경우
        max_down = down  # 최대 하차 인원 수 업데이트
        index_down = i  # 최대 하차 인원 수가 기록된 행의 인덱스 저장

print(f"최대 승차: {data_np[index_on][3]}역 {max_on}")  # 최대 승차 인원 수와 그 역의 이름을 출력
print(f"최대 하차: {data_np[index_down][3]}역 {max_down}")  # 최대 하차 인원 수와 그 역의 이름을 출력
```

[출력 결과]

```
최대 승차: 강남역 2209994
최대 하차: 강남역 2175932
```

2) 유임승차자의 비율(유임승차/총 승차인원)이 가장 높은 상위 10개 역과 각각의 유임승차자 비율을 구하세요. (25점)

```
ratio = []  # 승차 비율을 저장할 리스트를 생성

for i in range(len(data_np)):  # 데이터 배열의 각 행에 대해 반복
    data_np[i][9] = change_int(data_np[i][9])  # 9번째 열의 데이터를 숫자로 변환

    a = data_np[i][4] / (data_np[i][4] + data_np[i][6])  # 유임 승차 인원 수를 총 승차 인원 수로 나눈 유임 승차자의 비율 계산
    ratio.append(a)  # 유임 승차자 비율을 ratio 리스트에 추가

data_upd = np.column_stack((data_np, ratio))  # data_np 배열과 ratio 리스트를 열 방향으로 합쳐 data_upd 배열 생성
sorted_array = data_upd[np.argsort(data_upd[:, -1])[::-1]]  # data_upd 배열에서 유임 승차자의 비율 열을 기준으로 내림차순 정렬

for i in range(10):  # 상위 10개의 행에 대해 반복
    print(f"{i+1}. {sorted_array[i][3]}역 {sorted_array[i][10]:.5f}")  # 숫자, 역 이름, 유임 승차자의 비율 출력
```

[출력 결과]

```
1. 한양대역 0.95570
2. 홍대입구역 0.94716
3. 홍대입구역 0.94658
4. 서울역역 0.94036
5. 마곡나루(서울식물원)역 0.94005
6. 신논현역 0.93840
7. 여의도역 0.93735
8. 한강진역 0.93722
9. 디지털미디어시티역 0.93384
10. 청라국제도시역 0.93327
```

2. 지하철 시간대별 승하차 데이터 분석 (50점)

[데이터 로드]

```
import csv

data = []  # 데이터를 저장할 빈 리스트 생성
with open('Metro_time.csv','r',encoding ='cp949') as f:  # 'Metro_time.csv'파일을 열고 파일 객체를 'f'로 저장
    source = csv.reader(f, delimiter=',')  # csv.reader 함수를 통해 csv파일 읽어오기
    header = next(source)  # 첫 번째 행을 header로 지정
    header2 = next(source)  # 두 번째 행을 header2로 지정

    for row in source:  # csv 파일에서 한 행씩 읽어와
        data.append(row)  # data 리스트에 저장
```

1) 출근 시간대 (7시~9시) 승차인원과 하차인원 각각에 대해 정렬된 barplot을 그리고, 인원이 가장 많은 역과 인원을 각각 찾으세요. (25점)

```
data_np = np.array(data, dtype = object) # 리스트 형태의 데이터를 배열로 변환

work_on = []  # 승차 인원을 저장할 리스트 생성
work_down = []  # 하차 인원을 저장할 리스트 생성
max_1 = 0  # 최대 승차 인원 수를 저장할 변수 초기화
max_2 = 0  # 최대 하차 인원 수를 저장할 변수 초기화

for i in range(len(data_np)):  # 데이터 배열의 각 행에 대해 반복
    for j in range(4,52):  # 승하차 인원 정보가 있는 열에 대해 반복
        if isinstance(data_np[i][j], str):  # 데이터가 문자열인 경우
              data_np[i][j] = change_int(data_np[i][j])  # change_int 함수를 사용해 문자열을 숫자로 변환

    sum_on = data_np[i][10] + data_np[i][12]  # 7~8시, 8~9시 승차 인원 수를 합하여 총 승차 인원 수 계산
    work_on.append(sum_on)  # 총 승차 인원 수를 work_on 리스트에 추가
```

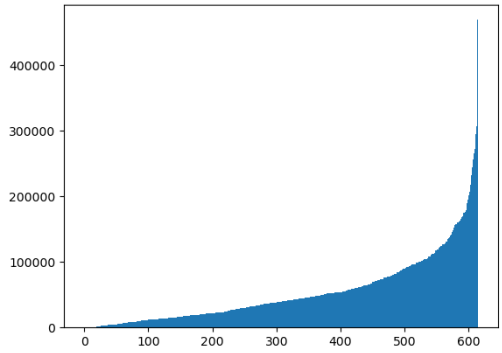
```
    if sum_on > max_1:  # 해당 역의 승차 인원 수가 최대 승차 인원 수보다 큰 경우
        max_1 = sum_on  # 최대 승차 인원 수를 업데이트
        index_1 = i  # 최대 승차 인원 수가 기록된 행의 인덱스를 저장


    sum_down = data_np[i][11] + data_np[i][13]  # 7~8시, 8~9시 하차 인원 수를 합하여 총 하차 인원 수 계산
    work_down.append(sum_down)  #총 하차 인원 수를 work_down 리스트에 추가

    if sum_down > max_2:  # 해당 역의 하차 인원 수가 최대 하차 인원 수보다 큰 경우
        max_2 = sum_down  # 최대 하차 인원 수를 업데이트
        index_2 = i  # 최대 하차 인원 수가 기록된 행의 인덱스 저장

# 승차인원 그래프 그리기
on_np = np.array(work_on)  # work_on 리스트를 numpy 배열로 변환
on_sort = on_np[np.argsort(on_np)]  # 승차 인원 수를 오름차순으로 정렬

plt.bar(range(len(on_sort)),on_sort, width=2)  # 승차 인원을 막대그래프 그리기
```



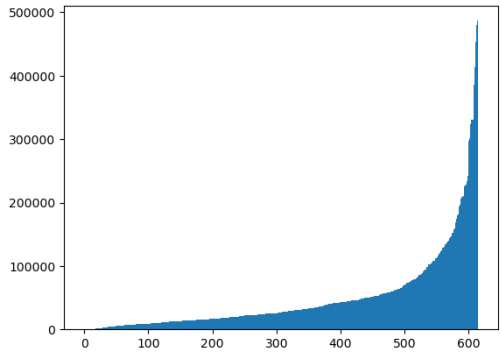

```
# 승차인원이 가장 많은 역 출력
print(f"승차인원 가장 많은 역: {data_np[index_1][3]}역 {max_1}")  # 최대 승차 인원 수에 해당하는 역의 이름과 승차 인원수 출력
```

[출력 결과]

```
승차인원 가장 많은 역: 신림역 469127
```

```
# 하차인원 그래프 그리기
down_np = np.array(work_down)  # work_down 리스트를 numpy 배열로 변환
down_sort = down_np[np.argsort(down_np)]  # 하차 인원 수을 오름차순으로 정렬

plt.bar(range(len(down_sort)),down_sort, width=2)  # 하차 인원을 막대그래프로 그리기
```



```
# 하차인원이 가장 많은 역 출력
print(f"하차인원 가장 많은 역: {data_np[index_2][3]}역 {max_2}")  # 최대 하차 인원 수에 해당하는 역의 이름과 하차 인원수 출력
```

[출력 결과]

```
하차인원 가장 많은 역: 가산디지털단지역 486420
```

2) 각 시간대별 모든 지하철역에서의 총 승차인원의 합과 총 하차인원의 합을 구하고, x축을 시간, y축을 인원으로 하는 그래프로 나타내시오. (25점)

```
for i in range(len(data_np)):  # 데이터 배열의 각 행에 대해 반복
    for j in range(4,52):  # 승하차 인원 정보가 있는 열에 대해 반복
        if isinstance(data_np[i][j], str):  # 데이터가 문자열일 경우
            data_np[i][j] = change_int(data_np[i][j])  # 문자열을 숫자로 변환

on_col = data_np[:,4:-1:2]  # 승차 인원 데이터 열을 추출
total_on = np.sum(on_col,axis=0)  # 각 시간대별 모든 지하철역의 승차 인원의 총합 계산


down_col = data_np[:,5:-1:2]  # 하차 인원 데이터 열을 추출
total_down = np.sum(down_col,axis = 0)  # 각 시간대별 모든 지하철역 하차 인원의 총합 계산

# 그래프 그리기
time = np.arange(4,28)  # x축에 사용할 시간대를 생성 (오전 4시 ~ 익일 오전 4시 전)

plt.plot(time,total_on,'blue',label='board')  # 각 시간대별 총 승차 인원 그래프 그리기
plt.plot(time,total_down,'orange',label = 'deboard') # 각 시간대별 총 하차 인원 그래프 그리기
plt.legend(loc = 'upper right')  # 범례를 그래프의 오른쪽 위에 표시

plt.show()  # 그래프 출력
```

[출력 결과]