데이터 과학

2주차

tashu

201202156 오희빈

합치는 과정

- tashu.csv => 년도별로 tashu데이터를 csv파일로 통합한 후 인코딩을 Linux 커멘드를 사용하여 UFT-8로 변환 후 필요한 4개의 칼럼만 남도록 통합
특히 15년도 파일의 날짜 데이터 앞의 '는 notepad++을 사용하여 모두바꾸기를 사용하여 '를 제거하였다.

- station.csv => 인코딩을 Linux 커멘드를 사용하여 UTF-8로 바꾸고 '명칭'을 제외한 나머지 칼럼들을 삭제해서 사용
-

1. 가장 인기 있는 정류장 Top10

```
def get_top10_station(tashu_dict, station_dict):

    j = 0
    rent_station = []
    return_station = []
    name = []
    station_count = [0 for i in range(250)]
    station_num = [q for q in range(250)]
    result = [['' for col in range(3)] for row in range(10)]

    for rent in tashu_dict:
        rent_station.append(rent['RENT_STATION'])
        return_station.append(rent['RETURN_STATION'])

    for station in station_dict:
        name.append(station['명칭'])
```

- rent_station => RENT_STATION의 값들을 저장하는 list
- return_station => RETURN_STATION의 값들을 저장하는 list
- station_count => 모든 정류장의 count값을 저장하기 위한 list
- name => 모든 정류장의 이름을 저장하기 위한 list
- station_num => 모든 정류장의 정류장번호를 저장하기 위한 list

- result => 가장 인기 있는 정류장 10개를 저장하기 위한 list

for rent in tashu_dict 와 for station in station_dict를 이용하고 append()함수를 이용해서 RENT_STATION과 RETURN_STATION, 명칭을 각각의 list에 저장한다.

```
    for z in rent_station:
        if rent_station[j] != '':
            if return_station[j] != '':
                station_count[int(rent_station[j])] += 1
        if return_station[j] != '':
            if rent_station[j] != '':
                station_count[int(return_station[j])] += 1
        j = j + 1

    counting_sort(station_count,station_num)

    for a in range(10):
        result[a][0] = name[station_num[a]-1]
        result[a][1] = str(station_num[a])
        result[a][2] = station_count[a]

    return result
```

for문을 이용해서 빌리고 반납한 station의 수를 count하는데 빌리고 반납한 station중 하나라도 station이 비어있으면 그 row는 count하지 않기에 if rent_station[]과 return_station[]을 하나씩 읽어서 둘 중 하나가 비어있는지 확인하고 station_count[int(rent_station[j])]을 이용해서 station_count list에 station번호를 list번호에 맞춰 count를 하나씩 증가시키고 return_station인 경우도 위와 마찬가지로 count를 하나씩 증가시켜 빌리고 반납한 station의 count를 저장한다.

count_sort함수를 만들어서 station_count를 sort해서 station_num과 함께 값을 return시킨다.
그리고 마지막으로 그 중 가장 count가 큰 10개를 result에 2차원 list 형식으로 저장시키고 return result를 해서 인기 있는 top10을 구한다.

counting_sort

```
def counting_sort(A,B):
    for i in range(1,250):
        j = i
        temp_value = A[i]
        temp_num = B[i]
        while temp_value > A[j-1] and j > 0:
            A[j] = A[j-1]
            B[j] = B[j-1]
            j -= 1
        A[j] = temp_value
        B[j] = temp_num

    return A,B
```

결과

```
C:\Users\heebin\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/heebin/PycharmProjects/tashu_test/sort.py
[' 한밭수목원(정문입구)', '3', 348977]
[' 충대정문(장대네거리)', '56', 182114]
['유성구청', '31', 166866]
['타임월드 앞 ', '17', 165778]
[' 홈플러스(유성점)', '32', 147063]
['월평역', '33', 142310]
['둔산 하이마트 앞', '14', 114878]
['카이스트 서쪽 쪽문', '105', 112921]
['카이스트 학사식당 앞', '21', 111715]
[' 충대정문오거리 1', '55', 110045]

Process finished with exit code 0
```

2. 가장 인기 있는 경로 Top10

```
def get_top10_trace(tashu_dict, station_dict):

    num = 0
    num2 = 0
    len_station = 228*228
    num3 = 0
    rent_station = []
    return_station = []
    name = []
    station_name = ['' for w in range(228)]
    station = [[0 for col in range(228)] for row in range(228)]
    station_result = [[0 for col in range(5)] for row in range(228*228)]
    result = [['' for col in range(5)] for row in range(10)]

    for rent in tashu_dict:
        rent_station.append(rent['RENT_STATION'])
        return_station.append(rent['RETURN_STATION'])

    for station_tashu in station_dict:
        name.append(station_tashu['명칭'])
```

- num, num2, num3 => 뒤에 list에 값들을 저장하기 위해 사용하는 변수
- rent_station => RENT_STATION의 값들을 저장하는 list
- return_station => RETURN_STATION의 값들을 저장하는 list
- station_count => 모든 정류장의 count값을 저장하기 위한 list
- station_num => 모든 정류장의 정류장번호를 저장하기 위한 list
- name => 정류장의 이름을 저장하기 위한 list
- station => 모든 정류장의 경로 경우의 수와 맞게 count 하기 위한 list
- station_result => 모든 정류장의 경로의 경우의 수와 이름 번호를 저장하기 위한 list
- result => 가장 인기 있는 경로 10개를 저장하기 위한 list

for rent in tashu_dict 와 for station in station_dict를 이용하고

append()함수를 이용해서 RENT_STATION과 RETURN_STATION, 명칭을 각각의 list에 저장한다.

```
cnt = len(rent_station)
cnt_name = len(name)

for y in range(0,cnt_name):
    station_name[y+1] = name[y]

for j in range(0,cnt):
    if rent_station[j] != '' :
        if return_station[j] != '':
            station[int(rent_station[j])][int(return_station[j])] += 1
```

모든 정류장 이름을 station_name에 저장한다.
모든 경로의 경우의 수에 해당하는 값들과 RENT_STATION과 RETURN_STATION을 한 줄씩 읽어서 그 경로와 맞는 2차원 list의 자리의 값을 하나씩 증가시켜 모든 경로의 count값을 저장한다.

```
    for j in range(len_station):
        if num2 != 228:
            string = str(num)
            string2 = str(num2)
            station_result[num3][0] = station_name[num]
            station_result[num3][1] = string
            station_result[num3][2] = station_name[num2]
            station_result[num3][3] = string2
            station_result[num3][4] = int(station[num][num2])
            num2 = num2 +1
            num3 = num3 + 1
        else :
            num = num + 1
            num2 = 0

    insert(station_result)

    for j in range(0,10):
        result[j] = station_result[j]

    return result
```

station_result list에 빌린 정류장 이름과 정류장 번호 그리고 반납한 정류장 이름과 정류장 번호, 그 경로의 count 수를 저장한다. 총 경우의 수는 228*228의 수가 되고 list에 빌린 정류장을 0부터 반납 정류장을 0부터 227까지 하나씩 증가 시키고 그에 맞는 경로의 count 수를 저장하고 빌린 정류장을 하나 증가시키고 다시 이를 반복해서 모든 경로를 list에 저장시킨다. 그리고 insert() 함수를 사용해 count 수를 중점으로 station_result list를 count시킨다.
마지막으로 count한 station_result를 10개 result list에 저장시키고 return시킨다.

```
def insert(array):
    for i in range(len(array)):
        if i > 0:
            while array[i-1][4] <= array[i][4] and i > 0:
                array[i-1], array[i] = array[i], array[i-1]
                i = i-1

    return array
```

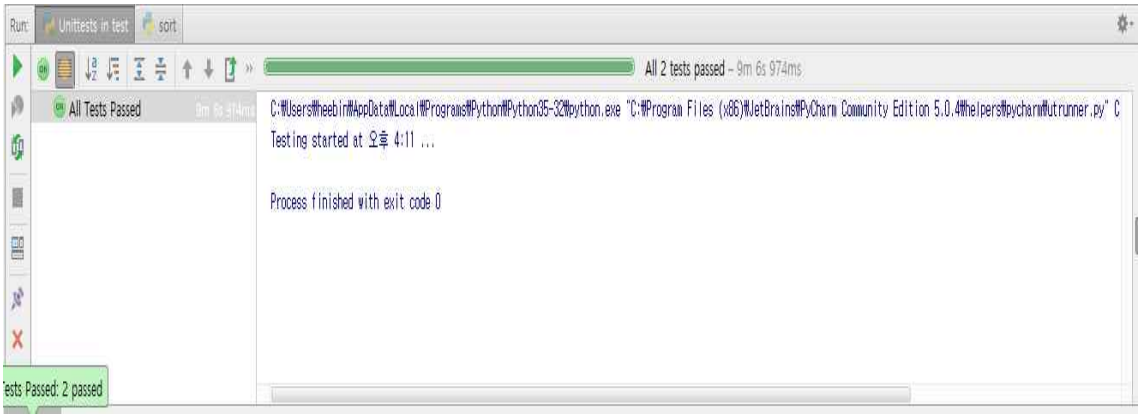
=> list의 count 수를 이용해서 list를 큰 순서로 sort한다.

결과

```
C:\Users\heebin\AppData\Local\Programs\Python\Python35-32\python.exe C:/Users/heebin/PycharmProjects/tashu_test/test2.py
[' 한밭수목원(정문입구)', '3', ' 한밭수목원(정문입구)', '3', 84496]
['유성구청', '31', '유성구청', '31', 21749]
[' 충대정문(장대네거리)', '56', ' 충대정문(장대네거리)', '56', 18343]
['카이스트 학사식당 앞', '21', '카이스트 서쪽 쪽문', '105', 17220]
['무역전시관입구(택시승강장 앞)', '1', '무역전시관입구(택시승강장 앞)', '1', 14489]
[' 홈플러스(유성점)', '32', ' 홈플러스(유성점)', '32', 12177]
['카이스트 서쪽 쪽문', '105', '카이스트 학사식당 앞', '21', 12154]
['월평역', '33', '월평역', '33', 11973]
['타임월드 앞 ', '17', '타임월드 앞 ', '17', 11966]
[' 충대정문(장대네거리)', '56', ' 홈플러스(유성점)', '32', 11868]

Process finished with exit code 0
```

test결과



=> 이상 없이 1, 2번 둘 다 test를 통과했다.