





- ✓ 서울 지하철 역별 시간별 승차 · 하차 인원
- ★ 서울 지하철 역별 유임 · 무임 승차 · 하차 인원



# 학습목표

- ☑ 임의시간 지하철 승차 · 하차 최대 역을 검색할 수 있다.
- ☑ 서울 지하철 시간별 승차 · 하차 인원이 가장 많은 역을 찾아 데이터 시각화할 수 있다.
- ☑ 서울 지하철 유임 · 무임 승차 · 하차 가장 많은 역 찾을 수 있다.
- ☑ 서울 지하철 유임·무임 승차·하차 비율 pie 차트를 그릴 수 있다.

LESSON 01

# 지하철 시간대별 이용 역 데이터 시각화 들



#### 지하철 시간대별 이용 역 데이터 시각화



200,999 등차수열 설명해 줘

## → 밤 11시에 사람들이 가장 많이 타는 역은 어디일까 (1/2)

# ⇒ 승차 시간 t와 인덱스 i 사이의 관계식 찾기

Index 0 1 2 3 4 5 6 7 8 9 10 11 12 ·

	А	В	С	D	Е	F	G	н	1	J	К	L	М
1	사용월	호선명	역ID	지하철역	04:00:00~0	4:59:59	05:00:00~0	5:59:59	06:00:00~	0 <mark>6:59:59</mark>	07:00:00~0	7:59:59	08:00:00
2					승차	하차	승차	하차	승차	하차	승차	하차	승차
3	Aug-22	1호선	150	서울역	573	19	8,638	8,274	12,332	45,706	39,560	102,779	63,5
4	Aug-22	1호선	151	시청	39	0	2,005	4,665	3				
5	Aug-22	1호선	152	종각	54	4	3,356	4,382	3	등차수열(Arithn	netic Sequence)은 [	계속하는 두 항 사이	의 차이가 일정

승차 시간 t	인덱스 i	관계식
4	4	
5	6 n-1	이듯이 직관적으로 t-4라
6	8	
7	10	i = 4 + (t - 4) * 2 = 2t - 4
8	12	1 20 4
23	?	

등차수열(Arithmetic Sequence)은 연속하는 두 항 사이의 차이가 일정한 수열을 말합니다. 이 일정 한 차이를 공차(common difference)라고 부르며, 이를 d로 나타냅니다.

등차수열의 일반적인 항을 수식으로 나타내면 다음과 같습니다.

$$a_n = a_1 + (n-1) \times d$$

여기서:

- a<sub>n</sub>: 수열의 n 번째 8
- a<sub>1</sub>: 수열의 첫 번째 항
- *d*: 공치
- n: 항의 순서

예를 들어, 첫 번째 향이 3이고 공차가 5인 등차수열을 생각해 봅시다. 그러면 이 수열은 다음과 같습니다.

3, 8, 13, 18, 23, . . .

이 수열의 각 항은 다음과 같이 계산됩니다.

- a<sub>1</sub> = 3
- $a_2 = 3 + 5 = 8$
- $a_3 = 3 + 2 \times 5 = 13$
- $a_4 = 3 + 3 \times 5 = 18$



### → 밤 11시에 사람들이 가장 많이 타는 역은 어디일까 (2/2)

# 활 밤 11시에 사람들이 가장 많이 타는 역 찾기

```
import csv
f = open('subwaytime.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
next(data)
# print(header)
max people = 0
max station = ''
t = int(input('몇 시의 승차 인원이 가장 많은 역이 궁금하세요?: '))
for row in data:
   for i in range(4, 52):
       row[i] = row[i].replace('.'. "")
   row[4:] = map(int. row[4:])
   people = row[2 * t - 4]
   if people > max_people:
       max people = people
       max\_station = row[3] + '(' + row[1] + ')'
f.close()
print(max_station, ':', max_people, '명')
몇 시의 승차 인원이 가장 많은 역이 궁금하세요?: 23
홍대입구(2호선) : 81164 명
```

#### ☑ 지하철 시간대별 이용 역 데이터 시각화



#### → 시간대별로 사람들이 가장 많이 타고 내리는 역은 어디일까 (1/7)

- ✔ 시간대별 최대 승차 역 이름 및 승차 인원 출력하기
  - ☑ 새벽 4시~새벽 3시 → 24시간을 1시간 단위로 구분 → 24개 → for 반복문 사용
  - ☑ 변수 j와 인덱스 i 사이의 관계식 찾기

Index 0 1 2 3 4 5 6 7 8 9 10 11 12 ··

	А	В	С	D	E	F	G	н	1	J	К	L	М	N	ı
1	사용월	호선명	역ID	지하철역	04:00:00~0	4:59:59	05:00:00~0	5:59:59	06:00:00~0	6:59:59	07:00:00~0	7:59:59	08:00:00~0	8:59:59	C
2					승차	하차	ŧ								
3	Aug-22	1호선	150	서울역	573	19	8,638	8,274	12,332	45,706	39,560	102,779	63,523	200,999	
4	Aug-22	1호선	151	시청	39	0	2,005	4,665	3,404	23,606	6,430	65,621	8,401	181,920	

변수 j	인덱스 i	관계식
0	4	
1	6	
2	8	i = 2 * j + 4
	···	1-2 ]+4
22	48	
23	50	



#### → 시간대별로 사람들이 가장 많이 타고 내리는 역은 어디일까 (2/7)

# ✔ 시간대별 최대 승차 역 이름 및 승차 인원 출력하기

```
import csv
f = open('subwaytime.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
next(data)
# print(header)
max people = [0] * 24
max station = [''] * 24
for row in data:
   for i in range(4, 52):
        row[i] = row[i].replace(',', "")
   row[4:] = map(int, row[4:])
   for i in range(24):
        people = row[2 * j + 4]
        if people > max_people[j]:
            max people[i] = people
           max_station[j] = row[3]
f.close()
print(max station)
print(max people)
```

#### ☑ 지하철 시간대별 이용 역 데이터 시각화



#### → 시간대별로 사람들이 가장 많이 타고 내리는 역은 어디일까 (3/7)

# ✔ 시간대별 최대 승차 역 이름 및 승차 인원 출력하기(실행결과)

['구로', '신림', '신림', '신림', '신림', '신림', '신림', '신림', '잠실(송파구청)', '잠실(송파구청)', '강남', '공남', '공남', '장남', '강남', '강남', '공남', '공남', '공남', '동암', '용산'] [11898, 38345, 71039, 184073, 274665, 165979, 88153, 76852, 84468, 98834, 106471, 131677, 154492, 241721, 321759, 197282, 153516, 177 679, 167850, 81164, 14129, 29, 2, 1]

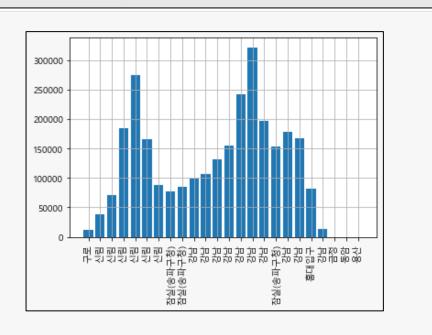


#### → 시간대별로 사람들이 가장 많이 타고 내리는 역은 어디일까 (4/7)



#### **≥ 시간대별 최대 승차 역 이름 및 승차 인원 막대그래프로 시각화하기**

```
import csv
import matplotlib.pyplot as plt
f = open('subwaytime.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
next (data)
# print(header)
max people = [0] * 24
max_station = [''] * 24
for row in data:
    for i in range(4, 52):
        row[i] = row[i].replace(',', "")
    row[4:] = map(int, row[4:])
    for i in range(24):
        people = row[2 * i + 4]
        if people > max_people[j]:
            max people[i] = people
            \max \text{ station[i]} = \text{row[3]}
f.close()
plt.rc('font', family='Malgun Gothic')
plt.bar(range(24), max people)
plt.xticks(range(24), max station, rotation=90)
plt.grid(True)
plt.show()
```



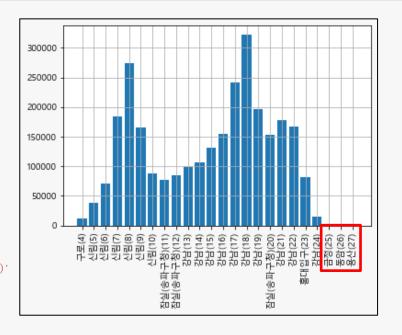
• x축에 몇 시인지 추가하면 좋을 것 같습니다.



## → 시간대별로 사람들이 가장 많이 타고 내리는 역은 어디일까 (5/7)

# **◇** 시간 정보 추가하기

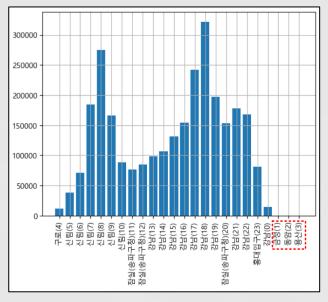
```
import csv
import matplotlib.pyplot as plt
f = open('subwaytime.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
next (data)
# print(header)
max_people = [0] * 24
max_station = [''] * 24
for row in data:
    for i in range(4, 52):
        row[i] = row[i].replace(',', "")
    row[4:] = map(int, row[4:])
   for i in range(24):
       people = row[2 * j + 4]
        if people > max_people[j]:
            max people[i] = people
            \max_{station[j]} = row[3] + '(' + str(j + 4) + ')'
f.close()
plt.rc('font', family='Malgun Gothic')
plt.bar(range(24), max_people)
plt.xticks(range(24), max station, rotation=90)
plt.grid(True)
plt.show()
```





#### → 시간대별로 사람들이 가장 많이 타고 내리는 역은 어디일까 (6/7)





```
# 시간대별 최대 승차 역 이름 및 승차 인원 막대그래프로 시각화
import csv
import matplotlib.pyplot as plt
f = open('subwaytime.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
next(data)
max people = [0] * 24
max_station = [''] * 24
for row in data :
    for i in range(4, 52):
       row[i] = row[i].replace(',', '')
   row[4:] = map(int, row[4:])
    for j in range(24):
        people = row[2 * j + 4]
       if people > max people[j]:
           max_people[j] = people
           max station[j] = row[3] + f
f.close()
plt.rc('font', family='Malgun Gothic')
plt.bar(range(24), max people)
plt.xticks(range(24), max station, rotation=90)
plt.grid(True)
plt.show()
```

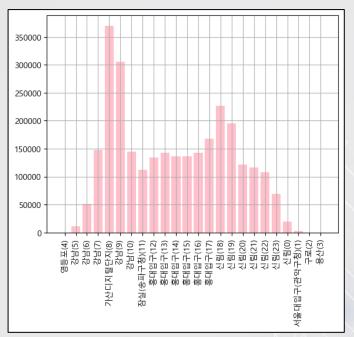


#### → 시간대별로 사람들이 가장 많이 타고 내리는 역은 어디일까 (7/7)



#### **≥ 시간대별 최대 하차 역 이름 및 하차 인원 막대그래프로 시각화하기**

```
# 시간대별 최대 하차 역 이름 및 하차 인원 막대그래프로 시각화
import csv
import matplotlib.pvplot as plt
f = open('subwaytime.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
next(data)
max people = [0] * 24
max_station = [''] * 24
for row in data :
    for i in range(4, 52):
       row[i] = row[i].replace(',', '')
    row[4:] = map(int, row[4:])
   for j in range(24):
       people = row[2 * j + 5]
       if people > max people[j]:
           max_people[j] = people
           \max_{station[j]} = row[3] + '(' + str((j + 4) % 24) + ')'
f.close()
plt.rc('font', family='Malgun Gothic')
plt.bar(range(24), max_people, color='pink')
plt.xticks(range(24), max station, rotation=90)
plt.grid(True)
plt.show()
```



#### 지하철 시간대별 이용 역 데이터 시각화



## → 모든 지하철역에서 시간대별 승하차 인원을 모두 더하면 (1/3)

- ❤️ 알고리즘(Algorithm) 설계하기
  - ☑ Step 1) 데이터를 읽어온다.
  - ☑ Step 2) 모든 역에 대해 시간대별 승차 인원과 하차 인원을 누적해서 더한다.
  - ☑ Step 3) 시간대별 승차 인원과 하차 인원을 그래프로 표현한다.



#### → 모든 지하철역에서 시간대별 승하차 인원을 모두 더하면 (2/3)

# **❷** 시간대별 승하차 인원을 저장할 리스트 만들기

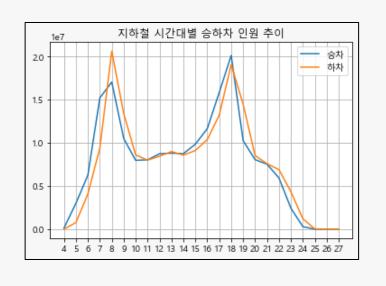
```
import csv
f = open('subwaytime.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
next (data)
s_in = [0] * 24 # 승차 인원을 저장할 리스트 초기화
s out = [0] * 24
                # 하차 인원을 저장할 리스트 초기화
for row in data:
   for i in range(4, 52):
       row[i] = row[i].replace(',', "")
   row[4:] = map(int, row[4:])
   for i in range(24):
       s_{in}[i] = s_{in}[i] + row[2 * i + 4]
       s out[i] = s out[i] + row[2 * i + 5]
f.close()
```



#### → 모든 지하철역에서 시간대별 승하차 인원을 모두 더하면 (3/3)



```
import csv
import matplotlib.pyplot as plt
f = open('subwaytime.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
next (data)
s in = [0] * 24 # 승차 인원을 저장할 리스트 초기화
s out = [0] * 24 # 하차 인원을 저장할 리스트 초기화
for row in data:
    for i in range(4, 52):
        row[i] = row[i].replace(',', "")
    row[4:] = map(int, row[4:])
    for i in range(24):
       s_{in}[i] = s_{in}[i] + row[2 * i + 4]
       s out[i] = s out[i] + row[2 * i + 5]
f.close()
plt.rc('font', family='Malgun Gothic')
plt.title('지하철 시간대별 승하차 인원 추이')
plt.plot(s_in, label='승차')
plt.plot(s_out, label='하차')
plt.legend()
plt.grid(True)
plt.xticks(range(24), range(4, 28))
plt.show()
```



LESSON 02

# 유임·무임승차 데이터 시각화





#### → 유무임 승하차 인원이 가장 많은 역은 어디일까 (1/4)

- 🤪 이번에는 유무임 승하차 인원이 가장 많은 역이 어디인지 찾아보겠습니다.
  - ☑ 유임 승차 인원이 가장 많은 역은?
  - ☑ 유임 하차 인원이 가장 많은 역은?
  - ☑ 무임 승차 인원이 가장 많은 역은?
  - ☑ 무임 하차 인원이 가장 많은 역은?

# ❤️ 알고리즘(Algorithm) 설계하기

- ☑ Step 1) 데이터를 읽어온다.
- ☑ Step 2) 모든 역의 데이터를 바탕으로 유임 승차, 유임 하차, 무임 승차, 무임 하차 인원이 가장 많은 역을 각각 찾는다.
- ☑ Step 3) 각각의 인원이 가장 많은 역을 출력한다.



- → 유무임 승하차 인원이 가장 많은 역은 어디일까 (2/4)
- ❷ 인원수를 저장할 리스트의 이름을 max\_people이라고 하겠습니다.
- ❤️ max\_people 리스트를 초기화 하는 방법을 살펴보겠습니다.

```
max_people = [0, 0, 0, 0]
   print(max people)
   max people = [0] * 4
   print(max people)
   max people = []
   for i in range(4):
       max people.append(0)
   print(max people)
   max_people = [0 for _ in range(4)]
   print(max people)
 ✓ 0.0s
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]
```

4가지 방법 모두초기화 방법으로 사용할 수 있습니다.



- → 유무임 승하차 인원이 가장 많은 역은 어디일까 (3/4)
- ❤️ 역이름을 저장할 리스트의 이름을 max\_station이라고 하겠습니다.
- ❤️ max\_station 리스트를 초기화 하는 방법을 살펴보겠습니다.

```
max_station = ['', '', '', '']
print(max_station)
max_station = [''] * 4
print(max station)
max station = []
for i in range(4):
    max station.append('')
print(max station)
max_station = ['' for _ in range(4)]
print(max station)
```

4가지 방법 모두초기화 방법으로 사용할 수 있습니다.



#### → 유무임 승하차 인원이 가장 많은 역은 어디일까 (4/4)

# → 유무임 승하차 인원이 가장 많은 역 찾기

```
import csv
   f = open('subwayfee.csv', encoding='cp949')
   data = csv.reader(f)
   header = next(data)
   max_station = ['' for _ in range(4)]
   max people = [0 for in range(4)]
   # index: 4, 5, 6, 7
  label = ['유임승차','유임하차','무임승차','무임하차']
   for row in data :
      for i in range(4, 8):
          row[i] = int(row[i].replace(',', ''))
          if row[i] > max_people[i-4] :
              max people[i-4] = row[i]
              \max \ station[i-4] = row[3] + ' ' + row[1]
   f.close()
   for i in range(4):
      print(f'{label[i]}: {max_station[i]}, {max_people[i]} 'B')
 ✓ 0.0s
                                                                   Python
유임승차: 잠실(송파구청) 2호선, 2106912 명
유임하차: 홍대입구 2호선, 2210524 명
무임승차: 종로3가 1호선, 318029 명
무임하차: 제기동 1호선, 303921 명
```



- → 모든 역의 유무임 승하차 비율은 어떻게 될까 (1/4)
- **❷** 데이터가 있는 모든 역에 대한 유무임 승하차 비율을 표현해 보겠습니다.
- ❤️ 역마다 4개의 정보가 표현될 테니 원그래프(Pie Chart)를 이용하겠습니다.
- ➡ 유무임 승하차 비율 원그래프로 시각화하기

```
import csv
import matplotlib.pyplot as plt
f = open('subwayfee.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
label = ['유임승차', '유임하차', '무임승차', '무임하차']
color = ['skyblue', 'palegreen', 'pink', 'beige']
plt.rc('font', family='Malgun Gothic')
for row in data:
   for i in range(4, 8):
       row[i] = int(row[i].replace(',', ""))
   plt.title(row[3] + ' ' + row[1])
   plt.pie(row[4:8], labels=label, colors=color, autopct='%1.f%%')
    nlt_show()
f.close()
```



#### → 모든 역의 유무임 승하차 비율은 어떻게 될까 (2/4)

# → 유무임 승하차 비율 원그래프로 시각화하기(실행결과)





#### → 모든 역의 유무임 승하차 비율은 어떻게 될까 (3/4)

# **◈** 각 그래프를 이미지 파일로 저장하기

```
import csv
import matplotlib.pyplot as plt
f = open('subwayfee.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
label = ['유임승차', '유임하차', '무임승차', '무임하차']
color = ['skyblue', 'palegreen', 'pink', 'beige']

    savefig 함수를 이용하면 그래프를 저장할 수 있습니다.

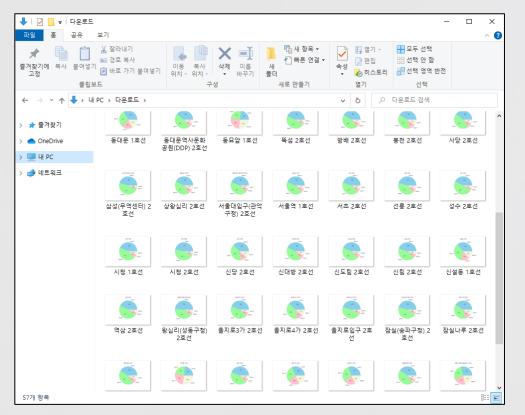
                                                             함수의 입력 값으로는 저장할 파일의 이름을 적어주면 됩니다.
plt.rc('font', family='Malgun Gothic')
for row in data:
   for i in range(4, 8):
       row[i] = int(row[i].replace(',', ""))
   plt.title(row[3] + ' ' + row[1])
   plt.pie(row[4:8], labels=label, colors=color, autopct='%1.f%%')
   plt.savefig(row[3] + ' ' + row[1] + '.png')
   nlt.show()
f.close()
```



#### → 모든 역의 유무임 승하차 비율은 어떻게 될까 (4/4)



#### **☞** 각 그래프를 이미지 파일로 저장하기(실행결과)



# SUMMARY

# 학습정긴





...

- 🧿 임의시간 지하철 승차 · 하차 최대 역을 검색
  - >> row[2\*t 4]: t시간 승차인원
- - max\_people = [0] \* 24
  - max\_station = ["] \* 24
- 전 서울 지하철 유임 · 무임 승차 · 하차 가장 많은 역 검색
  - max\_station = [" for \_ in range(4)]
  - max\_people = [0 for \_ in range(4)]
- び 서울 지하철 유임 · 무임 승차 · 하차 비율 pie 차트
  - » plt.pie(row[4:8], labels=label, colors=color, autopct='%1.f%%')



