





- ☑ 대중교통 데이터 준비하기
- ☑ 대중교통 데이터분석 기초



학습목표

- ☑ 티머니에서 대중교통 데이터를 내려 받을 수 있다.
 - subwayfee.xls, subwaytime.xls
- ☑ 대중교통 데이터에서 데이터의 구조를 이해할 수 있다.
 - 유임-무임 승차와 하차 인원, 시간대별 승차와 하차 인원
- 대중교통 데이터에서 지하철 유임승차 비율이 높은 역을 찾을 수 있다.
 - <u>- 0으로 나누는 문</u>제점 해결

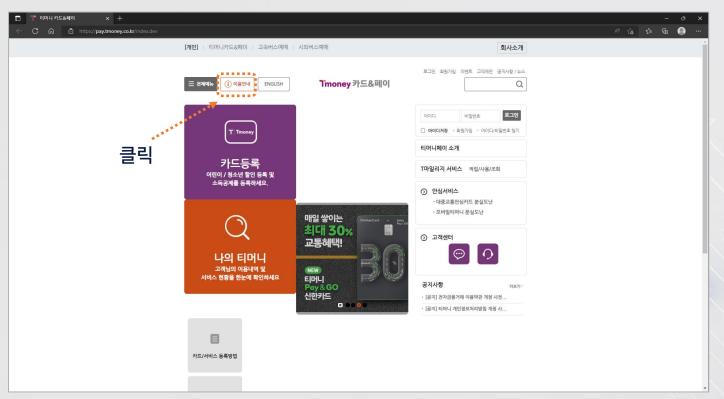
LESSON 01

대중교통 데이터 준비하기





- → 대중교통 데이터 내려받기 (1/4)
- 대중교통 통계자료 수집 → 티머니 홈페이지(https://pay.tmoney.co.kr/index.dev) - [①이용안내]



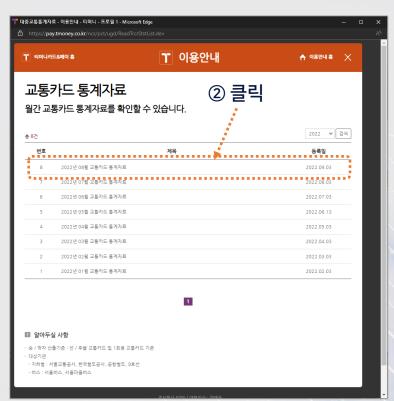


→ 대중교통 데이터 내려받기 (2/4)

② [대중교통 통계자료] 버튼 클릭 → [2022년 08월 교통카드 통계자료] 버튼 클릭



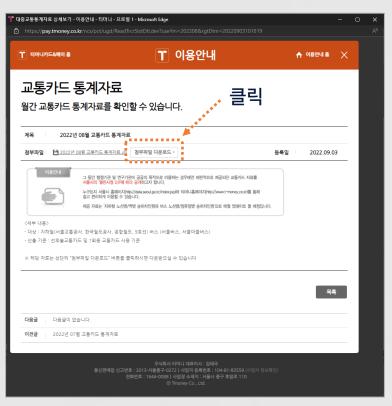






→ 대중교통 데이터 내려받기 (3/4)

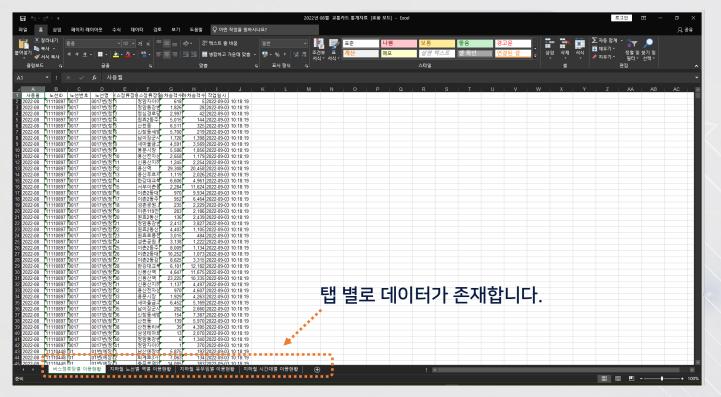
❷ [첨부파일 다운로드] 버튼 클릭



다운로드(Downloads) 폴더에 "2022년 08월 교통카드 통계자료.xls" 엑셀 파일이 저장됩니다.

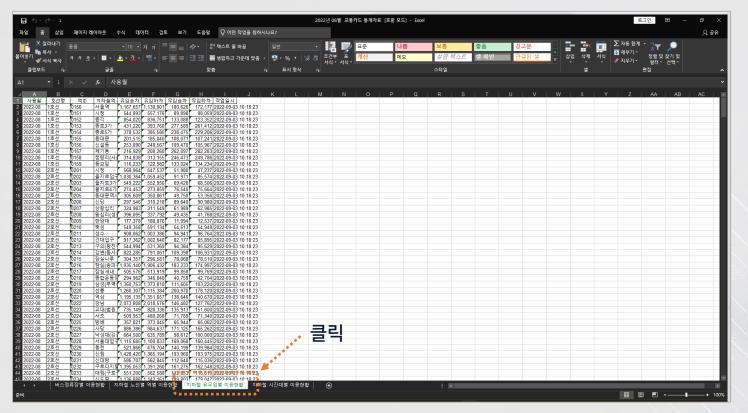


- → 대중교통 데이터 내려받기 (4/4)
- → 버스정류장별, 지하철 노선별 역별, 지하철 유무임별, 지하철 시간대별 이용현황 데이터가 존재합니다.



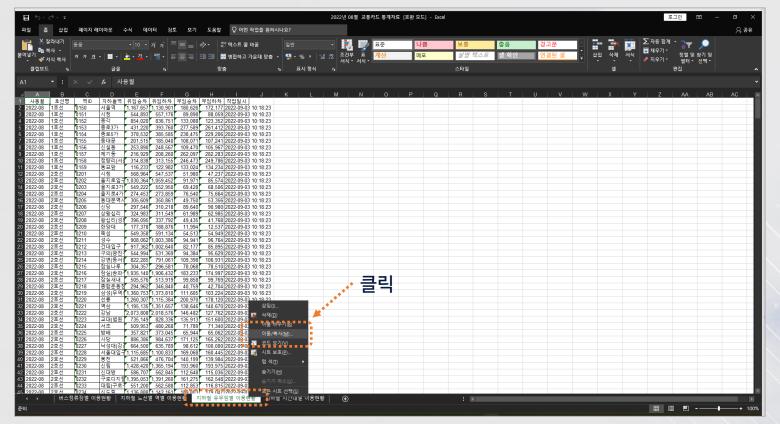


- → 지하철 유무임별 이용현황 데이터 정제하기 (1/6)
- ❤️ 4개의 탭 중 "지하철 유무임별 이용현황" 탭을 클릭합니다.





- → 지하철 유무임별 이용현황 데이터 정제하기 (2/6)
- ❤️ 워크시트 이름 탭을 마우스 오른쪽 단추로 클릭하고, [이동/복사(M)] 버튼을 클릭합니다.

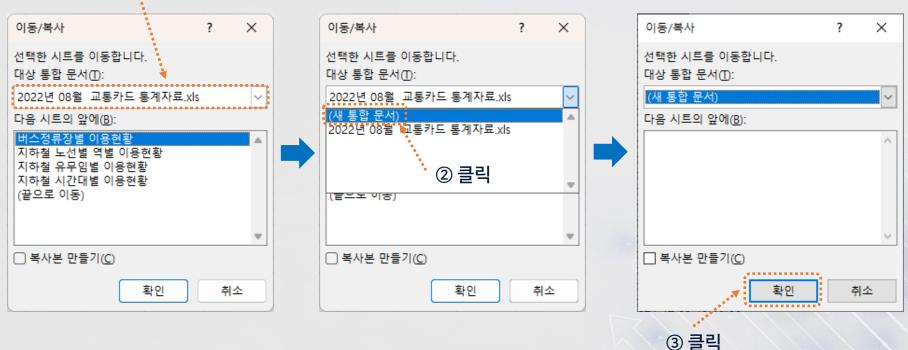




→ 지하철 유무임별 이용현황 데이터 정제하기 (3/6)

② [대상 통합 문서(T)] - [(새 통합 문서)] - [확인] 버튼을 클릭합니다.

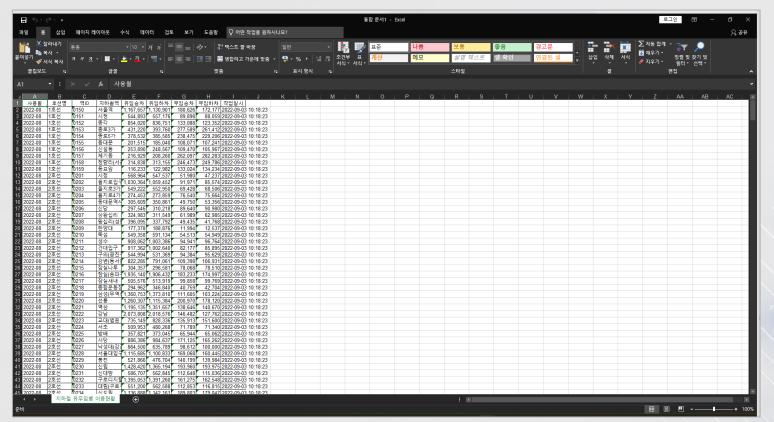
① 클릭





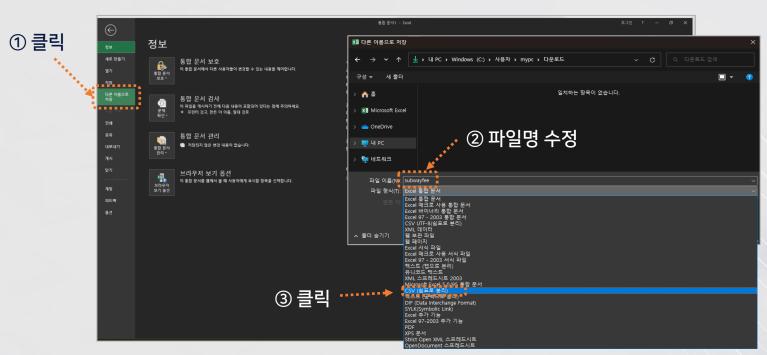
→ 지하철 유무임별 이용현황 데이터 정제하기 (4/6)

❷ 새 통합 문서가 열립니다.



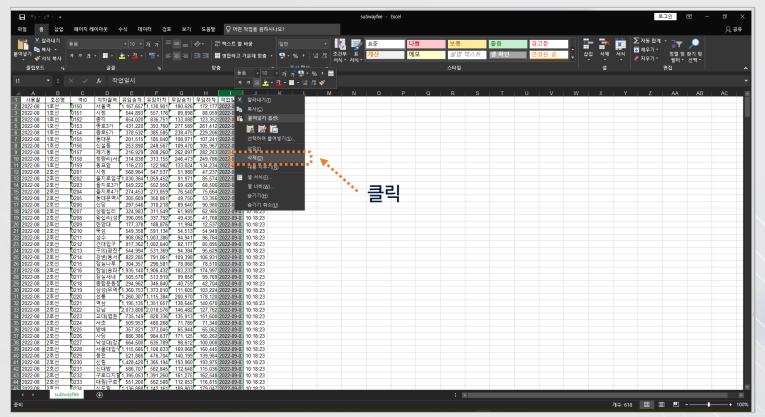


- → 지하철 유무임별 이용현황 데이터 정제하기 (5/6)
- ❤️ [파일] [다른 이름으로 저장] 버튼을 클릭하고, 파일 이름을 subwayfee로 수정합니다.
- **▶ 파일 형식을 "CSV (쉼표로 분리)"로 선택한 후 저장 버튼을 클릭합니다.**(저장 경로는 다운로드 폴더)



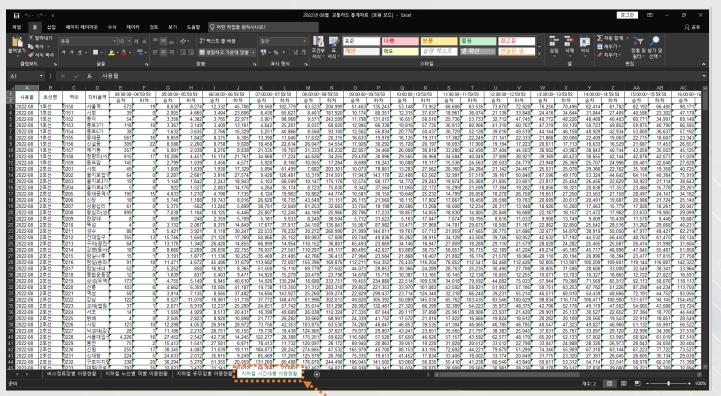


- → 지하철 유무임별 이용현황 데이터 정제하기 (6/6)
- ❤️ subwayfee.csv 파일을 열고, I열(Column)을 삭제 및 저장하고 파일을 닫습니다.





→ 지하철 시간대별 이용 현황 데이터 정제하기 (1/6)

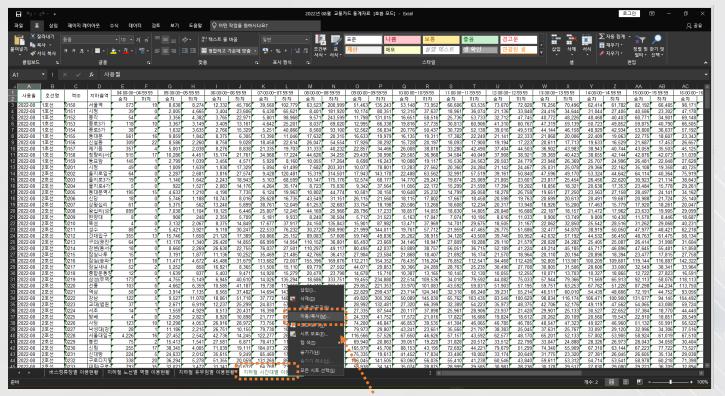






→ 지하철 시간대별 이용 현황 데이터 정제하기 (2/6) /

❤️ 워크시트 이름 탭을 마우스 오른쪽 단추로 클릭하고, [이동/복사(M)] 버튼을 클릭합니다.

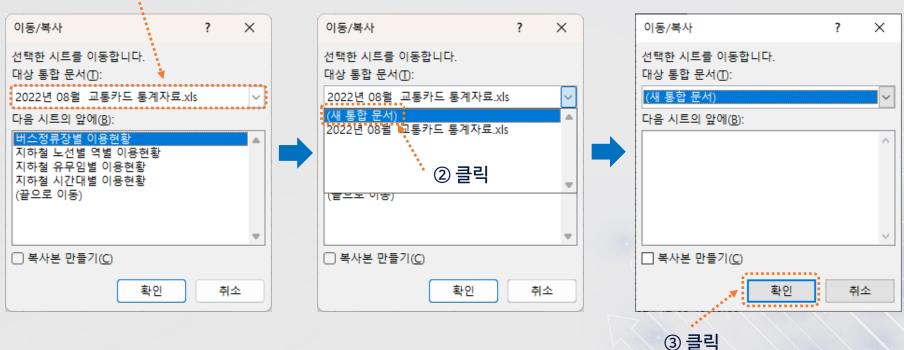




→ 지하철 시간대별 이용 현황 데이터 정제하기 (3/6) /

② [대상 통합 문서(T)] - [(새 통합 문서)] - [확인] 버튼을 클릭합니다.

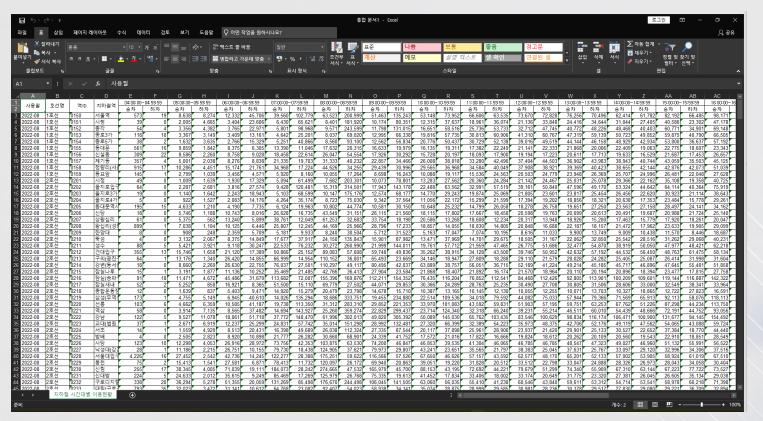
① 클릭





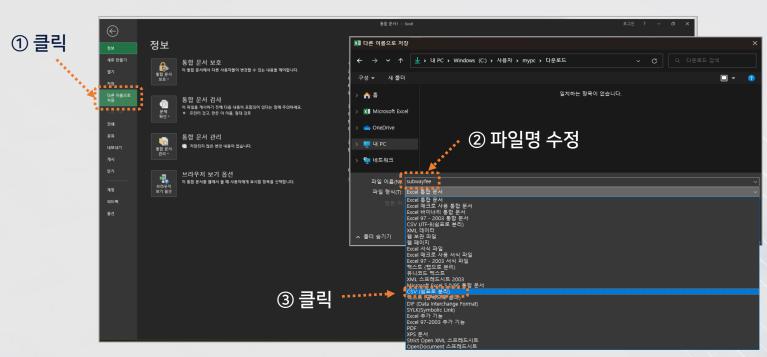
→ 지하철 시간대별 이용 현황 데이터 정제하기 (4/6) /

⇒ 새 통합 문서가 열립U다.



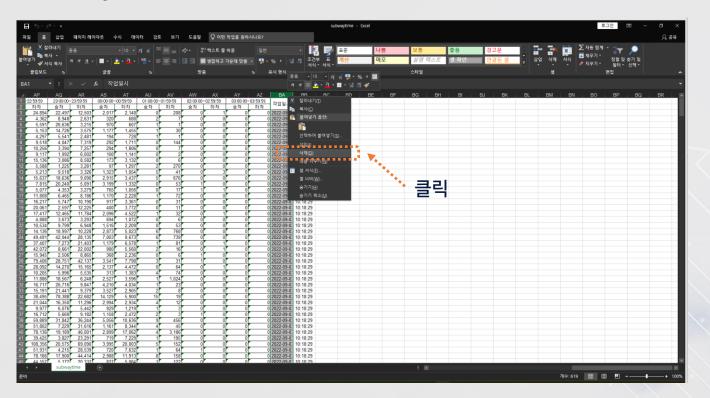


- ⊸ 지하철 시간대별 이용 현황 데이터 정제하기 (5/6) /
- 🧼 [파일] − [다른 이름으로 저장] 버튼을 클릭하고, 파일 이름을 subwaytime으로 수정
- **▶ 파일 형식을 "CSV (쉼표로 분리)"로 선택한 후 저장 버튼을 클릭합니다.**(저장 경로는 다운로드 폴더)





- ⊸ 지하철 시간대별 이용 현황 데이터 정제하기 (6/6) /
- ❤️ subwaytime.csv 파일을 열고, BA열(Column)을 삭제 및 저장하고 파일을 닫습니다.





⊸ 생각해 보기 (1/3)



🍑 출근 시간대 사람들이 가장 많이 타고 내리는 역은 어디일까요?

오전 7시 승차 데이터의 위치를 확인하니 10번 인덱스(Index)에 저장되어 있습니다.

3 5 11 12 Index 0 6 7 8 9 10 4 ... В м 사용월 호선명 역ID 지하철역 04:00:00~04:59:59 05:00:00~05:59:59 06:00:00~06:59:59 07:00:00~07:59:59 08:00:00~08:59:59 승차 하차 승차 하차 승차 하차 승차 하차 승차 하차 Aug-22 1호선 150 서울역 573 8,274 12,332 45,706 39,560 102,779 200,999 19 8,638 63,523 Aug-22 1호선 151 시청 23,606 39 2,005 4,665 3,404 6,430 65,621 8,401 181,920 152 종각 22,971 Aug-22 1호선 54 3.356 4.382 3.765 5.801 98,968 9.571 243,599 Aug-22 1호선 153 종로3가 118 10 3.367 3,149 3,409 13,161 4.642 25,201 8,037 69,020 Aug-22 1호선 154 종로5가 2.766 15.329 5.251 93.100 38 2 1.632 3.635 40.866 8.560 Aug-22 1호선 155 동대문 561 16 1,842 8,375 6,305 13,390 11,046 20,315 9,859 17,632 Aug-22 1호선 156 신설동 309 22 8,586 2.260 8,758 9,028 18,458 22.614 26,047 54,554 Aug-22 1호선 157 제기동 8,276 40,232 357 5,001 2,038 8,838 21,335 19,703 31,333 Aug-22 1호선 158 청량리(서 915 17 10,286 4.451 15,174 21,761 34,968 17,224 44,626 34,255 Aug-22 1호선 159 동묘앞 145 2.799 1,039 3,456 4.571 5,920 8.160 10,055 17,264 Aug-22 2호선 201 시청 49 1.639 1.930 17.329 5.094 61,499 203.301 0 1.009 7.682 Aug-22 2호선 202 을지로입구 64 2.287 2.681 3,816 27,574 9,428 120,481 15,319 314,501 0 203 을지로3가 Aug-22 2호선 19 1,140 1,642 2,243 18,943 5,103 68,599 10,147 175,176 204 을지로4가 14,176 4,264 Aug-22 2호선 0 922 1,527 2,083 35,174 8,723 75,030 205 동대문역시 15 1,210 7,735 6,124 44,774 Aug-22 2호선 195 4,633 4.190 19,963 10,802 Διια-22 2호선 206 시단 18 0 5 746 1 188 10.743 8.016 26.628 16 735 43 549 31 151

DMUAi

→ 생각해 보기 (2/3)



빨 함 11시에 사람들이 가장 많이 타는 역은 어디일까요?

승차 시간 t와 인덱스 i 사이의 관계식 찾기

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | ••• | |
|-------|--------|-----|-----|------|------------|---------|-----------|----------|------------|---------|------------|---------|------------|---------|---|
| | A | В | С | D | E | F | G | н | 1 | J | К | L | М | N | |
| 1 | 사용월 | 호선명 | 역ID | 지하철역 | 04:00:00~0 | 4:59:59 | 05:00:00~ |)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 | 승차 | 하차 | 승차 | 하차 | 승차 | 하차 | 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 | |
| 5 | Aug-22 | 1호선 | 152 | 종각 | 54 | 4 | 3,356 | 4,382 | 3,765 | 22,971 | 5,801 | 98,968 | 9,571 | 243,599 | |

| 승차 시간 t | 인덱스 i | 관계식 | | | | |
|---------|-------|---------------------------------|--|--|--|--|
| 4 | 4 | | | | | |
| 5 | 6 | | | | | |
| 6 | 8 | | | | | |
| 7 | 10 | i = 4 + (t - 4) * 2 = 2t - 4 | | | | |
| 8 | 12 | | | | | |
| ••• | ••• | | | | | |
| 23 | ? | | | | | |



⊸ 생각해 보기 (3/3)



❷ 시간대별로 사람들이 가장 많이 타고 내리는 역은 어디일까요?

시간대별 최대 승차 역 이름 및 승차 인원 출력하기

새벽 4시~새벽 3시 → 24시간을 1시간 단위로 구분 → 24개 → for 반복문 사용 변수 i와 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 | I | J | К | L | М | N | A |
| 1 | 사용월 | 호선명 | 역ID | 지하철역 | 04:00:00~0 | 4:59:59 | 05:00:00~ |)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 | 승차 | 하차 | 승차 | 하차 | 승차 | 하차 | 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 | |
| 5 | Aug-22 | 1호선 | 152 | 종각 | 54 | 4 | 3,356 | 4,382 | 3,765 | 22,971 | 5,801 | 98,968 | 9,571 | 243,599 | |

| 변수 j | 인덱스 i | 관계식 |
|------|-------|---------------|
| 0 | 4 | |
| 1 | 6 | |
| 2 | 8 | :-2*:/ |
| ••• | ••• | i = 2 * j + 4 |
| 22 | 48 | |
| 23 | 50 | |

LESSON 02

대중교통 데이터분석 기초





→ 지하철 유무임별 이용현황 데이터 정제하기 (1/2)



```
import csv
f = open('subwayfee.csv', encoding='cp949')
data = csv.reader(f)
                                                     헤더가 존재하고 있습니다.
for row in data:
    print(row)
f.close()
['사용월', '호선명', '역ID', '지하철역', '유임승차', '유임하차', '무임승차', '무임하차']
['2021-11', '1호선', '1', '서울역', '1,086,374', '1,044,750', '180,495', '170,842']
                     '10', '동묘앞', '119,396', '128,531', '137,406', '138,097']
                     '2', '시청', '516,353', '528,021', '72,373', '72,182']
                     '3', '종각', '797,960', '781,329', '118,815', '110,536']
['2021-11', '1호선', '4', '종로3가', '433,793', '397,349', '272,242', '254,421']
['2021-11', '1호선', '5', '종로5가', '362,377', '370,057', '230,840', '220,847']
['2021-11', '1호선', '6', '동대문', '210.503', '195.301', '112.597', '110.925']
['2021-11', '1호선', '7', '신설동', '271,006', '262,798', '108,130', '103,695']
                     '8', '제기동', '222,630', '214,314', '247,332', '263,992']
                          '청량리(서울시립대입구)'. '315.789'. '317.666'. '232.661'. '237.208']
```

인원수 데이터가 문자열 자료형으로 저장되어 있습니다. 그리고 쉼표(,)도 존재합니다.



→ 지하철 유무임별 이용현황 데이터 정제하기 (2/2)

● 인원수 데이터를 정수 자료형으로 바꾸고, 데이터 출력하기

```
import csv
f = open('subwayfee.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
for row in data:
   for i in range(4, 8):
       row[i] = int(row[i].replace('.'. ""))
   print(row)
f.close()
['2021-11', '1호선', '1', '서울역', 1086374, 1044750, 180495, 170842]
['2021-11', '1호선', '10', '동묘앞', 119396, 128531, 137406, 138097]
['2021-11', '1호선', '2', '시청', 516353, 528021, 72373, 72182]
['2021-11'. '1호선'. '3'. '종각', 797960, 781329, 118815, 110536]
['2021-11', '1호선', '4', '종로3가', 433793, 397349, 272242, 254421]
['2021-11', '1호선', '5', '종로5가', 362377, 370057, 230840, 220847]
['2021-11', '1호선', '6', '동대문', 210503, 195301, 112597, 110925]
「'2021-11', '1호선', '7', '신설동', 271006, 262798, 108130, 103695]
['2021-11', '1호선', '8', '제기동', 222630, 214314, 247332, 263992]
['2021-11', '1호선', '9', '청량리(서울시립대입구)', 315789, 317666, 232661, 237208]
```

☑근 대중교통 데이터분석 기초



→ 유임 승차 비율이 가장 높은 역은 어디일까 (1/13)

🤪 유임 승차 비율이 가장 높은 역은 어디일까요?

알고리즘(Algorithm)으로 생각하기

- Step 1) 데이터를 읽어온다.
- Step 2) 모든 역의 데이터를 바탕으로 각 역의 유임 승차 비율(Rate)을 계산한다.
- Step 3) 비율이 가장 높은 역을 찾는다.
- Step 4) 비율이 가장 높은 역이 어디인지, 그 비율이 얼마인지 출력한다.

☑ 유임(有賃): 값을 치름

☑ 무임(有賃): 값을 치르지

않음

Rate = 유임 승차 인원 / 무임 승차 인원 rate = row[4] / row[6]



→ 유임 승차 비율이 가장 높은 역은 어디일까 (2/13)

⇒ 유임 승차 인원/무임 승차 인원 비율의 최대값 찾기

```
import csv
f = open('subwayfee.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
print(header)
rate = 0
max rate = 0
for row in data:
    for i in range(4, 8):
        row[i] = int(row[i].replace(',', ""))
    rate = row[4] / row[6]
    if rate > max_rate:
       max_rate = rate
f.close()
print(max_rate)
```



- → 유임 승차 비율이 가장 높은 역은 어디일까 (3/13)
- ⇒ 유임 승차 인원/무임 승차 인원 비율의 최대값 찾기(실행결과)
 - ☑ 오류 메시지를 읽고, 오류 원인 생각해 보기

오류 문구를 확인해 보니, 0으로 값을 나누려고 했다고 하네요. row[6]의 무임 승차 인원 데이터에 0값이 존재하는 것 같네요.



→ 유임 승차 비율이 가장 높은 역은 어디일까 (4/13)

❤️ row[6]가 0인 데이터 출력하기

```
import csv
f = open('subwayfee.csv', encoding='cp949')
data = csv.reader(f)
header = ne \times t(data)
print(header)
rate = 0
max rate = 0
for row in data:
    for i in range(4, 8):
        row[i] = int(row[i].replace(',', ""))
    if row[6] == 0:
        print(row)
f.close()
```

row[6]가 0인 역이 존재하네요.

```
['사용월', '호선명', '역ID', '지하철역', '유임승차', '유임하차', '무임승차', '무임하차']
['2022-08', '경원선', '1022', '창동', 10, 0, 0, 0]
['2022-08', '분당선', '1031', '복정', 13, 0, 0, 0]
['2022-08', '경의선', '1297', '검암', 2, 0, 0, 0]
['2022-08', '7호선', '2756', '신중동', 1, 0, 0, 0]
['2022-08', '7호선', '2758', '상동', 2, 0, 0, 0]
['2022-08', '7호선', '2759', '삼산체육관', 1, 0, 0, 0]
['2022-08', '7호선', '2760', '골포천', 2, 0, 0, 0]
['2022-08', '7호선', '2761', '부평구청', 3, 0, 0, 0]
```



→ 유임 승차 비율이 가장 높은 역은 어디일까 (5/13)

❤️ row[6]가 0인 역은 제외하고, 유임 승차 인원/무임 승차 인원 비율의 최대값 찾기

```
import csv
f = open('subwayfee.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
print(header)
rate = 0
max rate = 0
for row in data:
    for i in range(4, 8):
        row[i] = int(row[i].replace(',', ""))
    if row[6] != 0:
        rate = row[4] / row[6]
        if rate > max rate:
            max rate = rate
            print(row, max_rate)
f.close()
```



- → 유임 승차 비율이 가장 높은 역은 어디일까 (6/13)
- → row[6]가 0인 역은 제외하고, 유임 승차 인원/무임 승차 인원 비율의 최대값 찾기 (실행결과)

```
['사용월', '호선명', '역ID', '지하철역', '유임승차', '유임하차', '무임승차', '무임하차']
['2022-08', '1호선', '0150', '서울역', 1167657, 1130901, 180626, 172177] 6.464501234595241
['2022-08', '2호선', '0201', '시청', 568964, 547537, 51980, 47237] 10.945825317429781
['2022-08', '2호선', '0202', '을지로입구', 1030364, 1059452, 91971, 85574] 11.203140120255298
['2022-08', '2호선', '0209', '한양대', 177378, 188876, 11994, 12537] 4.788894447223612
['2022-08', '2호선', '0239', '홍대입구', 1717050, 1851268, 90063, 86905] 19.064987841844044
```

max_rate 값의 소수점 아래 수가 너무 길어서 보기가 힘드네요.



→ 유임 승차 비율이 가장 높은 역은 어디일까 (7/13)

⇒ 유임 승차 인원/무임 승차 인원 비율의 최대값을 소수 둘째 자리 까지만 표현하기

```
import csv
f = open('subwayfee.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
print(header)
rate = 0
max rate = 0
for row in data:
    for i in range(4, 8):
        row[i] = int(row[i].replace(',', ''))
    if row[6] != 0 :
        rate = row[4] / row[6]
        if rate > max rate :
            max rate = rate
            # print(row, round(max rate, 2))
            print(row, f'{max rate:.2f}')
f.close()
```

round(max_rate, 2)는 max_rate 값을 소수점 셋째 자리에서 반올림하여 둘째 자리까지 표현하는 명령어입니다.



→ 유임 승차 비율이 가장 높은 역은 어디일까 (8/13)

⇒ 유임 승차 인원/무임 승차 인원 비율의 최대값을 소수 둘째 자리 까지만 표현하기(실행결과)

```
['사용월', '호선명', '역ID', '지하철역', '유임승차', '유임하차', '무임승차', '무임하차']
['2022-08', '1호선', '0150', '서울역', 1167657, 1130901, 180626, 172177] 6.46
['2022-08', '2호선', '0201', '시청', 568964, 547537, 51980, 47237] 10.95
['2022-08', '2호선', '0202', '을지로입구', 1030364, 1059452, 91971, 85574] 11.2
['2022-08', '2호선', '0209', '한양대', 177378, 188876, 11994, 12537] 14.79
['2022-08', '2호선', '0239', '홍대입구', 1717050, 1851268, 90063, 86905] 19.06
['2022-08', '6호선', '2615', '연신내', 52, 0, 2, 0] 26.0
```

비율의 최대값은 26.0으로 해당 역은 "연신내"역입니다. 그런데 "연신내"역의 정보가 이상한 것 같습니다.

데이터에 따르면, 유임승차 인원수가 52명이고 유임하차, 무임하차 인원수각 0명이네요.

이런 "연신내"역을 유임 승차 비율이 가장 높은 역이라고 부르는 것이 적절할까요?



→ 유임 승차 비율이 가장 높은 역은 어디일까 (9/13)

- **❷** 새로운 기준으로 유임 승차 비율 최대값 찾기
 - ☑ 기존 유임 승차 비율(Rate) 계산 방법

Rate = 유임 승차 인원 / 무임 승차 인원 rate = row[4] / row[6]

- ☑ 새로운 유임 승차 비율(Rate) 계산 방법
 - ◆ 유임 + 무임승차 인원을 합해서 100,000명 이상인 경우로만 한정

Rate = 유임 승차 인원 / 전체 (유임 + 무임승차) 인원 rate = row[4] / (row[4] + row[6])



→ 유임 승차 비율이 가장 높은 역은 어디일까 (10/13)

❷ 새로운 기준으로 유임 승차 비율 최대값 찾기

```
import csv
f = open('subwayfee.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
print(header)
rate = 0
max rate = 0
for row in data :
    for i in range(4, 8):
        row[i] = int(row[i].replace(',', ''))
    if row[6] != 0 and row[4] + row[6] > 100000:
        rate = row[4] / (row[4] + row[6])
        if rate > max_rate :
            max_rate = rate
            # print(row, round(max rate, 2))
            print(row, f'{max rate:.2f}')
f.close()
```

☑ 대중교통 데이터분석 기초



→ 유임 승차 비율이 가장 높은 역은 어디일까 (11/13) /

☑ 새로운 기준으로 유임 승차 비율 최대값 찾기(실행결과)

```
['사용월', '호선명', '역ID', '지하철역', '유임승차', '유임하차', '무임승차', '무임하차']
['2022-08', '1호선', '0150', '서울역', 1167657, 1130901, 180626, 172177] 0.87
['2022-08', '2호선', '0201', '시청', 568964, 547537, 51980, 47237] 0.92
['2022-08', '2호선', '0202', '을지로입구', 1030364, 1059452, 91971, 85574] 0.92
['2022-08', '2호선', '0209', '한양대', 177378, 188876, 11994, 12537] 0.94
['2022-08', '2호선', '0239', '홍대입구', 1717050, 1851268, 90063, 86905] 0.95
['2022-08', '공항철도 1호선', '4203', '홍대입구', 334381, 344582, 17164, 16166] 0.95
```

"연신대"역이 제외되었습니다. 새로운 기준으로 유임 승차 비율 최대값을 찾은 결과 최대값은 0.95이고, 해당 역은 "홍대입구" 역입니다.



→ 유입 승차 비율이 가장 높은 역은 어디일까 (12/13)

역을 살펴볼까요?

→ rate 〉 0.92로 수정 후, 코드 실행해 보기

```
import csv
f = open('subwayfee.csv', encoding='cp949')
data = csv.reader(f)
header = next(data)
print(header)
                                         유임 승차 비율이 0.92 이상인
rate = 0
max rate = 0
for row in data:
    for i in range(4, 8):
       row[i] = int(row[i].replace(',', ''))
    if row[6] != 0 and row[4] + row[6] > 100000:
       rate = row[4] / (row[4] + row[6])
       if rate > 0.92:
           max rate = rate
           # print(row, round(max rate, 2))
           print(row, f'{max rate:.2f}')
f.close()
```



→ 유임 승차 비율이 가장 높은 역은 어디일까 (13/13)

→ rate 〉 0.92로 수정 후, 코드 실행해 보기(실행결과)

```
[['사용월', '호선명', '역ID', '지하철역', '유임승차', '유임하차', '무임승차', '무임하차']
[['2022-08', '2호선', '0209', '한양대', 177378, 188876, 11994, 12537] 0.94
['2022-08', '2호선', '0219', '삼성(무역센터)', 1360753, 1373810, 111605, 103224] 0.92
['2022-08', '2호선', '0222', '강남', 2073808, 2018576, 146482, 127762] 0.93
「'2022-08', '2호선', '0238', '함정', 841083, 921865, 62256, 628391 0.93
                                                                           재미있는 결과가 나왔습니다.
['2022-08', '2호선', '0239', '홍대입구', 1717050, 1851268, 90063, 86905] 0.95
['2022-08', '3호선', '0327', '신사', 728935, 769927, 62606, 61011] 0.92
['2022-08', '경부선', '1001', '서울역', 246839, 70773, 13293, 2816] 0.95
['2022-08', '경부선', '1702', '가산디지털단지', 410976, 487813, 32364, 35714] 0.93
['2022-08', '분당선', '1848', '압구정로데오', 502688, 575961, 43710, 41839] 0.92
['2022-08', '5호선', '2527', '여의도', 710781, 755045, 50386, 47769] 0.93
['2022-08', '5호선', '2528', '여의나루', 333874, 381345, 28279, 27372] 0.92
['2022-08', '6호선', '2624', '상수', 241040, 285596, 20003, 20534] 0.92
['2022-08', '6호선', '2632', '한강진', 224575, 266059, 14949, 14544] 0.94
['2022-08', '7호선', '2748', '가산디지털단지', 1011630, 1004540, 80508, 75856] 0.93
['2022-08', '9호선', '4115', '여의도', 721152, 680330, 46281, 42069] 0.94
['2022-08', '9호선', '4125', '신논형', 790971, 787147, 48569, 46859] 0.94
['2022-08', '공항철도 1호선', '4203', '홍대입구', 334381, 344582, 17164, 16166] 0.95
['2022-08', '공항철도 1호선', '4204', '디지털미디어시티', 338524, 313402, 24556, 23008] 0.93
['2022-08', '공항철도 1호선', '4206', '마곡나루(서울식물원)', 216283, 206531, 12852, 12660] 0.94
['2022-08', '공항철도 1호선', '4207', '김포공항', 337165, 182448, 22741, 14639] 0.94
['2022-08', '공항철도 1호선', '4210', '청라국제도시', 178960, 168163, 12695, 12606] 0.93
```

SUMMARY

학습정긴





...

🧔 티머니에서 대중교통 데이터 내려받기

이용안내 > 대중교통 통계자료 > 월별 통계자료

🔅 열의 제목줄과 데이터 내용 검토

필요 없는 줄은 제거

🧿 유임 승차 비율이 높은 지하철 역 찾기

- >> 잘못된 계산 방법
 - Rate = 유임 승차 인원 / 무임 승차 인원(0이 아니어야 함)
 - rate >= 1
- 계산 방법 수정
 - Rate = 유임 승차 인원 / 전체 (유임 + 무임승차) 인원
 - -0 < rate <= 1
 - rate >= .92인 모든 지하철역 검색



