각 데이터에 대한 이해하기

CSV 1. 대여소 정보 (usage_stations.csv)

[column 설명]

- 대여소 번호: 대여소의 코드 (301)
- 보관소(대여소명): 대여소 이름 문자열 (ex 경복궁역 7 번출구 앞)
- 자치구: 대여소가 포함된 서울시 자치구(ex 종로구)
- 상세주소: 도로명주소+상세한 위치(ex 서울특별시 종로구 사직로 지하 130 경복궁역 7 번출구 앞)
- 위도: 위도 실수형 숫자 (ex 37.57579422)
- 경도: 경도 실수형 숫자 (ex 126.9714508)
- 설치 시기: 대여소가 설치된 날짜 시계열 데이터 (ex 2015-10-07)
- QR: QR 방식으로 대여하는 자전거의 설치 대수
- LCD: LCD 방식으로 대여하는 자전거의 설치 대수
- 운영 방식: QR/LCD 중 운영하는 방식("QR", "LCD' 중 하나가 입력됨)

CSV 2. 시간대별 이용정보 (usage_time.csv)

[column 설명] 대여일자: 대여한 날짜의 시계열 데이터 (ex 2025-06-01) 대여시간: 0~23 숫자(시간 단위, ex 오후 1 시-오후 2 시 사이에 빌렸다면 13) 대여소번호: 대여가 발생한 대여소 번호 대여소명: 대여가 발생한 대여소의 이름(대여소 번호에 종속) 대여구분코드: 일일권/정기권/가족권 등 발생한 대여의 대여권 정보 성별: 대여한 사람의 성별(남자, 여자, 미입력 시 NaN) 연령대코드: 범주형, 문자열(~10 대/20 대/ ...) 이용건수: 대여일자, 대여시간, 대여소번호, 대여소명, 대여구분코드, 성별, 연령대코드 정보가 모두 동일한 대여들의 대여 건수 합산(ex 같은 날 같은 시간대에서 21 세 남자와 26 세 남자가 모두 정기권으로 대여를 했다면 이 행의 이용건수는 2 이다) 운동량: 위 이용건수 조건 하 운동량의 총합 탄소량: 위 이용건수 조건 하 탄소 감축량 효과의 총합 이동거리: 위 이용건수 조건 하 이용자들의 이용거리 총합

CSV 3, 4: 일별 이용정보, 월별 이용정보 (usage_day, usage_month)

[column 설명] CSV 2 에서 대여시간 열이 제거되고, 이용시간(분)이 추가된 형태. 이용시간(분): 이용건수 조건 하 이용자들의 이용시간 총합

운동량 및 탄소량

운동량: 운동량(kcal), 이동거리와 정비례하여 계산됨 탄소량: 탄소배출량(g), 이동거리와 정비례하여 계산됨

외국인 대여

외국인의 대여는 함께 집계되지 않음

분석해볼 만한 것

서비스 전체 패턴

- 시간대별 전체 이용 곡선
- 요일 x 시간 Heatmap
- 월별 계절성
- 권종별 비중 변화(일일권, 정기권, 가족권) + 시간대별 분해
- 성별/연령대별 이용 비중의 시간대 분포(예: 20 대 vs 50 대 피크 시간)

대여소 기준

- 자치구별 이용량
- 대여소별 이용량 분포
- 지도 시각화
- 핫스팟/콜드스팟: 상위 대여소와 하위 대여소의 공통점

이용자 기준

- 연령대별 이용 시간대 패턴(ex 출퇴근대 수요, 등하교 수요 등)
- 요일별 수요 패턴
- 성별 x 연령대 x 권종으로 구성비 비교(모자이크플롯/스택바)
- 권종별 평균 이동거리, 운동량, 탄소량
- 건당 이동거리, 이용시간
- 시간대별 수요 패턴 + 주중/주말 비교

모델링 시 무엇을 target(label)으로 할까? => 이용건수, 운동량, 탄소량, 이동거리

```
# 라이브러리, 데이터
import pandas as pd
import numpy as np

stations = pd.read_csv("dataset/usage_stations.csv")
time = pd.read_csv("dataset/usage_time.csv")
day = pd.read_csv("dataset/usage_day.csv")
month = pd.read_csv("dataset/usage_month.csv")

# 1. stations 분석
print(stations.shape)
print(stations.nunique())
print(stations['gu'].unique()) # 서울시25 개 구 모두에 설치되어 있음
print(stations['bike_type'].unique()) # LCD 만 배치, QR 만 배치, LCD 와QR 모두 배치하는 방식이 있음

# station id 는2780 개인데station name 은2779 개임
print(stations['station_name'].mode()) # 금천구와 중랑구에 각각 '한양수자인아파트 앞' 이라는 대여소가 존재함
```

```
(2780, 10)
station_id     2780
station_name    2779
gu                25
address         2700
latitude        2687
longitude       2693
installed        572
qr                26
lcd               41
bike_type          3
dtype: int64
['종로구' '중구' '용산구' '성동구' '광진구' '동대문구' '중랑구' '성북구' '강북구'
'도봉구' '노원구' '은평구'
 '서대문구' '마포구' '양천구' '강서구' '구로구' '금천구' '영등포구' '동작구' '관악
구' '서초구' '강남구' '송파구'
 '강동구']
['QR' 'LCD' 'LCD,QR']
0     한양수자인아파트 앞
Name: station_name, dtype: object
```

```python
# 2. 시간대별 사용자 분석
print(time.shape)
print(time.nunique())
print(time['rental_type'].unique()) # 총5 가지 대여 방식이 있음
print(time['gender'].unique()) # nan: 입력 정보 없음, M/m, F/f 가 있음
print(time['gender'].value_counts()) # m 과f 는 매우 적음, M 과F 로 통일하면 될 듯함
print(time['age'].unique()) # 기타: 입력 정보 없음
```

```
(3644614, 11)
date                   30
hour                   24
station_id           2742
station_name         2742
rental_type             5
gender                  4
age                     8
rentals                24
calories            70966
carbon_reduction     1429
distance           762237
dtype: int64
['정기권' '일일권' '일일권(비회원)' '가족권(2 시간)' '가족권']
[nan 'F' 'M' 'm' 'f']
gender
M    1716230
F     983359
m        321
```

```
f        176
Name: count, dtype: int64
['~10대' '20대' '30대' '40대' '50대' '60대' '70대이상' '기타']

print(time.isna().sum())

date                    0
hour                    0
station_id              0
station_name            0
rental_type             0
gender             944528
age                     0
rentals                 0
calories            16772
carbon_reduction    16772
distance                0
dtype: int64

# 3. 일별, 월별 사용자 분석
print(day.shape)
print(day.nunique())
print(day['gender'].unique())
print(day.isna().sum())

(2146663, 11)
date                   30
station_id           2742
station_name         2742
rental_type             5
gender                  4
age                     8
rentals                69
calories            90604
carbon_reduction     2331
distance           856548
time                 1162
dtype: int64
[nan 'F' 'f' 'M' 'm']
date                    0
station_id              0
station_name            0
rental_type             0
gender             588812
age                     0
rentals                 0
calories             6695
carbon_reduction     6695
distance                0
```

```
time                        0
dtype: int64

print(month.shape)
print(month.nunique())
print(month.isna().sum())

(600437, 11)
date                        6
station_id               2760
station_name             2760
rental_type                 5
gender                      4
age                         8
rentals                   880
calories               283999
carbon_reduction        17878
distance               558568
time                      8916
dtype: int64
date                        0
station_id                  0
station_name                0
rental_type                 0
gender                 185977
age                         0
rentals                     0
calories                    0
carbon_reduction            0
distance                    0
time                        0
dtype: int64
```