t1_p2

문제 1. [정렬 및 상위 추출]

- 다음 조건을 만족하는 평균값을 계산하시오.
  - mtcars.csv 데이터에서 mpg 값을 기준으로 내림차순 정렬한 후, 상위 10개의 mpg 평균을 소수 둘째 자리까지 반올림하여 출력하시오.

```
import pandas as pd
import numpy as np
```

```
df= pd.read_csv("data/mtcars.csv")
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  32 non-null     object 
 1   mpg         32 non-null     float64
 2   cyl         32 non-null     int64  
 3   disp        32 non-null     float64
 4   hp          32 non-null     int64  
 5   drat        32 non-null     float64
 6   wt          32 non-null     float64
 7   qsec        32 non-null     float64
 8   vs          32 non-null     int64  
 9   am          32 non-null     int64  
 10  gear        32 non-null     int64  
 11  carb        32 non-null     int64  
dtypes: float64(5), int64(6), object(1)
memory usage: 3.1+ KB
```

```
df.head(2)
```

	Unnamed: 0	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.9	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.9	2.875	17.02	0	1	4	4

```
df.rename(columns={'Unnamed: 0':'car_name'},inplace=True)
```

```
df.head(2)
```

	car_name	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.9	2.620	16.46	0	1	4	4

	car_name	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1	Mazda RX4 Wag	21.0	6	160.0	110	3.9	2.875	17.02	0	1	4	4

```
round(df.sort_values(['mpg'],ascending=False).head(10)['mpg'].mean(),2)
```

```
27.19
```

```
# dir(df)
```

문제 2. [스케일링]

- 다음 조건을 만족하는 개수를 계산하시오.
  - qsec 컬럼을 MinMaxScaler로 0~1 범위로 정규화한 후, 변환된 값이 0.5 이상인 행의 개수를 출력하시오.

```
# import scipy.stats
# dir(scipy.stats)
```

```
# import sklearn.preprocessing
# dir(sklearn.preprocessing)
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler= MinMaxScaler()
scaled= scaler.fit(df[['qsec']]).transform(df[['qsec']])
df['scaled_qsec'] = scaled

(df['scaled_qsec'] >0.5).sum()
```

```
9
```

```
scaled
```

```
array([[0.23333333],
       [0.3       ],
       [0.48928571],
       [0.58809524],
       [0.3       ],
       [0.68095238],
       [0.15952381],
       [0.6547619 ],
       [1.        ],
       [0.45238095],
       [0.52380952],
       [0.3452381 ],
       [0.36904762],
       [0.41666667],
       [0.41428571],
       [0.3952381 ],
```

```
       [0.34761905],
       [0.59166667],
       [0.47857143],
       [0.64285714],
       [0.65595238],
       [0.28214286],
       [0.33333333],
       [0.10833333],
       [0.30357143],
       [0.52380952],
       [0.26190476],
       [0.28571429],
       [0.        ],
       [0.11904762],
       [0.01190476],
       [0.48809524]])
```

```
# help(scaler)
```

문제 3. [날짜/시간 처리]

- 다음 조건을 만족하는 평균값을 계산하시오.
  - 데이터에 2022-01-01부터 시작하는 주간 날짜를 date라는 컬럼으로 추가하시오. 이후 2022년 6월~12월 기간의 mpg 평균을 소수 둘째 자리까지 반올림하여 출력하시오.

```
df["date"] = pd.date_range(start="2022-01-01", periods=len(df), freq="W")
```

```
# df['date']
```

```
df['year']=df['date'].dt.year
df['month']=df['date'].dt.month
df['dayofweek']=df['date'].dt.weekday
df['week']=df['date'].dt.isocalendar().week
```

```
# help(df['date'].dt)
```

```
filtered_df=df[(df['year']==2022)& (df['month'].between(6,12))]
avg_mpg_6_to_12 = round(filtered_df["mpg"].mean(), 2)
```

```
df.head(2)
```

	car_name	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	scaled_qsec	date	year	m
0	Mazda RX4	21.0	6	160.0	110	3.9	2.620	16.46	0	1	4	4	0.233333	2022-01-02	2022	1
1	Mazda RX4 Wag	21.0	6	160.0	110	3.9	2.875	17.02	0	1	4	4	0.300000	2022-01-09	2022	1

```
filtered_df.shape
```

```
(10, 18)
```

```
### 날짜 파생변수 생성해서 머신러닝 작업 2유형에서 중요한 변수로 사용 가능
```

변수	처리 방식	이유
`month`, `weekday`, `season`	**범주형 → 원-핫 인코딩**	순서 의미 없음, 트리 모델 분할에 유리
`year`	**연속형 그대로 사용 가능**	시간 흐름 반영 (예: 2021→2022)

모델	인코딩 방식	비고
RandomForest, XGBoost	보통 get_dummies 사용 권장	순서 의미 제거, 성능 안정성 ↑
LogisticRegression, SVM, NN	무조건 인코딩 필요	숫자 외 입력 불가
CatBoost, LightGBM	`categorical_features` 지정 시 **자동 처리 가능**	원-핫 없이도 처리 가능 (성능 좋음)

문제 4. [파생변수 생성]

- 다음 조건을 만족하는 새로운 컬럼 2개를 생성하시오.
  - power_to_weight: hp 값을 wt로 나눈 비율
  - mpg_label: mpg가 20 이상이면 "high", 미만이면 "low"로 구분

```
df= pd.read_csv("data/mtcars.csv")
```

```
df.head(3)
```

	Unnamed: 0	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1

```
df.shape[0],len(df)
```

```
(32, 32)
```

```
df['power_to_weight']=df['hp']/df['wt']
```

```
df.head(2)
```

	Unnamed: 0	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	power_to_weight
0	Mazda RX4	21.0	6	160.0	110	3.9	2.620	16.46	0	1	4	4	41.984733
1	Mazda RX4 Wag	21.0	6	160.0	110	3.9	2.875	17.02	0	1	4	4	38.260870

```
df['mpg_label']=np.where(df['mpg']>=20,'high','low')
```

```
df.head(2)
```

	Unnamed: 0	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	power_to_weight	mpg_labe
0	Mazda RX4	21.0	6	160.0	110	3.9	2.620	16.46	0	1	4	4	41.984733	high
1	Mazda RX4 Wag	21.0	6	160.0	110	3.9	2.875	17.02	0	1	4	4	38.260870	high