t1_past_my_ans

문제1

- 데이터셋(basic1.csv)의 'f5' 컬럼을 기준으로 상위 10개의 데이터를 구하고,
- 'f5'컬럼 10개 중 최소값으로 데이터를 대체한 후,
- 'age'컬럼에서 80 이상인 데이터의'f5 컬럼 평균값 구하기

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

```
# 엑셀 파일 읽기
df_raw = pd.read_excel("data/basic1.xlsx", sheet_name='basic1')

# 첫 번째 열을 쉼표 기준으로 분리
df = df_raw.iloc[:, 0].str.split(",", expand=True)

# 컬럼명 설정
df.columns = ['id', 'age', 'city', 'f1', 'f2', 'f3', 'f4', 'f5']

# 타입 변환 (숫자 컬럼)
cols_to_numeric = ['age', 'f1', 'f2', 'f5']
df[cols_to_numeric] = df[cols_to_numeric].apply(pd.to_numeric, errors='coerce')
```

```
df1=df.copy()
```

```
df_sorted=df1.sort_values(['f5'],ascending=False).reset_index(drop=True)
```

```
f5min=df_sorted.head(10)['f5'].min()
df_sorted.loc[:9,'f5']=f5min
result= df_sorted[df_sorted['age']>=80]['f5'].mean()
result
```

```
62.49774712521738
```

문제2

- 데이터셋(basic1.csv)의 앞에서 순서대로 70% 데이터만 활용해서,
- 'f1'컬럼 결측치를 중앙값으로 채우기 전후의 표준편차를 구하고
- 두 표준편차 차이 계산하기

```
# 라이브러리 및 데이터 불러오기
import pandas as pd
import numpy as np
```

```
df2=df.copy()
```

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 8 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   id      100 non-null    object 
 1   age     100 non-null    float64
 2   city    100 non-null    object 
 3   f1      69 non-null     float64
 4   f2      100 non-null    int64  
 5   f3      100 non-null    object 
 6   f4      100 non-null    object 
 7   f5      100 non-null    float64
dtypes: float64(3), int64(1), object(4)
memory usage: 6.4+ KB
```

```
df2=df2.loc[:69,:]
```

```
df2.shape
```

```
(70, 8)
```

```
df2['f1'].isnull().sum()
```

```
23
```

```
df2['f1'].describe() # std = 17.987276
```

```
count     47.000000
mean      68.744681
std       17.987276
min       34.000000
25%       56.000000
50%       68.000000
75%       83.500000
max      111.000000
Name: f1, dtype: float64
```

```
std1= df2['f1'].agg('std')
median=df2['f1'].median()
```

```
df2.loc[:,'f1']= df2['f1'].fillna(median)
```

```
std2 = df2['f1'].std()
```

```
abs(std2-std1)
```

```
3.2965018033960725
```

```
df2.isnull().sum()
```

```
id      0
age     0
city    0
f1      0
f2      0
f3      0
f4      0
f5      0
dtype: int64
```

문제3

- 데이터셋(basic1.csv)의 'age'컬럼의 이상치를 더하시오!
- 단, 평균으로부터 1.5*표준편차를 벗어나는 영역을 이상치라고 판단함

```
df3=df.copy()
```

```
mean=df3['age'].mean()
```

```
std=df3['age'].std()
```

```
df3[(df3['age']>mean+1.5*std)|(df3['age']<mean-1.5*std)]['age'].sum()
```

```
473.5
```

```
# 혼자 연습
df4=df.copy()
df4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 8 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   id      100 non-null    object 
 1   age     100 non-null    float64
 2   city    100 non-null    object 
 3   f1      69 non-null     float64
 4   f2      100 non-null    int64  
 5   f3      100 non-null    object 
 6   f4      100 non-null    object 
```

```
 7   f5      100 non-null    float64
dtypes: float64(3), int64(1), object(4)
memory usage: 6.4+ KB
```

```
df4.head(2)
```

	id	age	city	f1	f2	f3	f4	f5
0	id01	2.0	서울	NaN	0		ENFJ	91.297791
1	id02	9.0	서울	70.0	1		ENFJ	60.339826

```
df4.groupby(['city','f4'])['age'].mean().reset_index()
```

	city	f4	age
0	경기	ENTJ	61.250000
1	경기	ENTP	63.500000
2	경기	ESFJ	1.000000
3	경기	ESFP	48.666667
4	경기	ESTJ	64.000000
5	경기	ESTP	36.666667
6	경기	INFJ	83.600000
7	경기	INFP	78.250000
8	경기	INTJ	66.333333
9	경기	INTP	52.750000
10	경기	ISFJ	82.500000
11	경기	ISFP	41.614286
12	경기	ISTP	-9.000000
13	대구	ENFJ	77.000000
14	대구	ENFP	53.333333
15	대구	ENTP	75.000000
16	대구	ESFJ	64.000000
17	대구	ESTJ	62.200000
18	대구	ESTP	39.000000
19	대구	INTP	59.500000
20	대구	ISFJ	50.000000
21	대구	ISTJ	36.000000
22	대구	ISTP	23.000000
23	부산	ENFJ	47.000000
24	부산	ENFP	24.350000
25	부산	ENTP	12.250000
26	부산	ESFJ	86.000000
27	부산	ESTJ	23.000000
28	부산	ESTP	64.500000
29	부산	INFJ	68.000000
30	부산	INFP	65.000000
31	부산	INTP	68.000000
32	부산	ISFJ	25.000000
33	부산	ISFP	90.000000

	city	f4	age
34	부산	ISTP	34.000000
35	서울	ENFJ	5.500000
36	서울	ENFP	40.000000
37	서울	ENTJ	77.000000
38	서울	ENTP	22.000000
39	서울	ESFJ	9.150000
40	서울	ESFP	68.000000
41	서울	ESTP	20.000000
42	서울	INFJ	38.000000
43	서울	INFP	75.000000
44	서울	INTJ	11.000000
45	서울	INTP	22.000000
46	서울	ISFJ	33.766667
47	서울	ISFP	74.000000
48	서울	ISTJ	27.000000
49	서울	ISTP	74.000000

```
df4.groupby(['city','f4'])['age'].mean().head()
```

```
city  f4
경기    ENTJ    61.250000
      ENTP    63.500000
      ESFJ     1.000000
      ESFP    48.666667
      ESTJ    64.000000
Name: age, dtype: float64
```