

```
# 모듈 임포트

import glob
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# 특정 디렉토리에 있는 .csv 확장자 파일 모두 불러오기
files = glob.glob(r'D:\Wteckit_data\mid_project\pre_processed_data\*.csv')
files

['D:\Wteckit_data\mid_project\pre_processed_data\data_adult_final.csv',
 'D:\Wteckit_data\mid_project\pre_processed_data\data_child_final.csv',
 'D:\Wteckit_data\mid_project\pre_processed_data\data_elder_final.csv',
 'D:\Wteckit_data\mid_project\pre_processed_data\data_teen_final.csv']

# 불러온 파일 확인하기
for file in files:
    print(file)

D:\Wteckit_data\mid_project\pre_processed_data\data_adult_final.csv
D:\Wteckit_data\mid_project\pre_processed_data\data_child_final.csv
D:\Wteckit_data\mid_project\pre_processed_data\data_elder_final.csv
D:\Wteckit_data\mid_project\pre_processed_data\data_teen_final.csv

# 불러온 파일을 raw_data 파일 하나로 합치기
raw_data = pd.DataFrame()

for file in files:
    new_data = pd.read_csv(file, parse_dates=['MESURE_DE'])
    raw_data = pd.concat([raw_data, new_data])
raw_data
```

Unnamed: 0						
0						
0	1	남양주	성인	32.0	3	2013-0
1	2	원주	성인	44.0	4	2013-0

```
raw_data.info()
raw_data.describe()

<class 'pandas.core.frame.DataFrame'>
Index: 807609 entries, 0 to 201458
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                -
0   Unnamed: 0                            807609 non-null  int64
1   CINTER_NM                            807609 non-null  object
2   AGRDE_FLAG_NM                        807609 non-null  object
3   MESURE_AGE_CO                        807609 non-null  float64
4   CRTFC_FLAG_NM                        807609 non-null  int64
5   MESURE_DE                            807609 non-null  datetime64[ns]
6   SEXDSTN_FLAG_CD                     807609 non-null  int64
7   MESURE_IEM_001_VALUE                 807609 non-null  float64
8   MESURE_IEM_002_VALUE                 807609 non-null  float64
9   MESURE_IEM_012_VALUE                 807609 non-null  float64
10  MESURE_IEM_018_VALUE                 807609 non-null  float64
11  MESURE_IEM_019_VALUE                 377502 non-null  float64
12  MESURE_IEM_020_VALUE                 610339 non-null  float64
13  MESURE_IEM_021_VALUE                 377502 non-null  float64
14  MESURE_IEM_022_VALUE                 610339 non-null  float64
15  MVM_PRSCRIPTN_CN                    807609 non-null  object
16  Unnamed: 0.1                        197270 non-null  float64
17  MESURE_IEM_023_VALUE                 197270 non-null  float64
18  MESURE_IEM_025_VALUE                 197270 non-null  float64
19  MESURE_IEM_027_VALUE                 197270 non-null  float64
dtypes: datetime64[ns](1), float64(13), int64(3), object(3)
memory usage: 129.4+ MB
```

Unnamed: 0					
MESURE_AGE_CO					
CRTFC_FLAG_NM					
MESURE_DE					
SEXDSTN_FLAG_CD					
count	807609.000000	807609.000000	807609.000000	807609	807609.000000
mean	10899.756303	39.415742	3.352980	2018-06-25 09:58:23.528563200	0.53114
min	0.000000	11.000000	1.000000	2013-03-05 00:00:00	0.000000
25%	3623.000000	17.000000	3.000000	2016-08-12 00:00:00	0.000000
50%	8134.000000	32.000000	4.000000	2018-09-10 00:00:00	1.000000
75%	15718.000000	64.000000	4.000000	2019-12-04 00:00:00	1.000000
max	49210.000000	102.000000	4.000000	2022-12-31 00:00:00	1.000000
std	9535.546151	23.442137	0.901834	NaN	0.499000

- 유소년
 - 평균대체: 1, 2
 - BMI 계산: 18
 - 핫덱대체: 12, 20, 22
- 청소년
 - 평균대체: 1, 2
 - BMI 계산: 18
 - 핫덱대체: 12, 20, 22
- 성인
 - 평균대체: 1, 2
 - BMI 계산: 18
 - 핫덱대체: 12, 19, 20, 21, 22

- 노인
 - 평균대체: 1, 2
 - BMI 계산: 18
 - 핫덱대체: 12, 23, 25, 27

```
data=raw_data[['AGRDE_FLAG_NM', 'MESURE_DE', 'MESURE_IEM_001_VALUE', 'MESURE_IEM_002_VALUE', 'MESURE_IEM_020_VALUE', 'MESURE_IEM_023_VALUE']]
```

```
data
```

	AGRDE_FLAG_NM	MESURE_DE	MESURE_IEM_001_VALUE	MESURE_IEM_002_VALUE	MESURE_IEM_020_VALUE	MESURE_IEM_023_VALUE
0	성인	2013-03-05	170.0	80.00		
1	성인	2013-03-05	157.4	66.60		
2	성인	2013-03-05	161.9	76.40		
3	성인	2013-03-05	179.2	98.30		
4	성인	2013-03-05	155.8	58.24		
...
201454	청소년	2022-12-28	162.7	46.70		
201455	청소년	2022-12-28	169.5	71.50		

```
data_child = data[data['AGRDE_FLAG_NM']=='유소년']
data_teen = data[data['AGRDE_FLAG_NM']=='청소년']
data_adult = data[data['AGRDE_FLAG_NM']=='성인']
data_elder = data[data['AGRDE_FLAG_NM']=='노인']
```

```
data_child=data_child.drop(['AGRDE_FLAG_NM', 'MESURE_IEM_023_VALUE'], axis=1)
```

```
data_child['year']=data_child['MESURE_DE'].dt.year
data_teen['year']=data_teen['MESURE_DE'].dt.year
data_adult['year']=data_adult['MESURE_DE'].dt.year
data_elder['year']=data_elder['MESURE_DE'].dt.year
```

C:\Users\W224km\AppData\Local\Temp\ipykernel_18612\W2158952909.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
data_teen['year']=data_teen['MESURE_DE'].dt.year
C:\Users\W224km\AppData\Local\Temp\ipykernel_18612\W2158952909.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
data_adult['year']=data_adult['MESURE_DE'].dt.year
C:\Users\W224km\AppData\Local\Temp\ipykernel_18612\W2158952909.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
data_elder['year']=data_elder['MESURE_DE'].dt.year
```

```
print(data_child['year'].value_counts(ascending=False))
print(data_teen['year'].value_counts(ascending=False))
print(data_adult['year'].value_counts(ascending=False))
print(data_elder['year'].value_counts(ascending=False))
```

```
year
2022    26798
2021     4089
2020      475
2019       16
Name: count, dtype: int64
year
2019     59932
2022     52041
2018     48341
2021     28573
2020     7679
```

```

2017    4892
2015         1
Name: count, dtype: int64
year
2019    71966
2015    57401
2018    57125
2014    46886
2017    44932
2016    42423
2020    29848
2013    21809
2022     2608
2021     2504
Name: count, dtype: int64
year
2019    33617
2018    32558
2017    25262
2022    20501
2016    20089
2015    20084
2014    16886
2021    10263
2020    10012
2013     7998
Name: count, dtype: int64

```

체력 평균 항목(유소년, 청소년, 성인: 왕복오래달리기(회) // 노인: 2분 제자리 걷기(회))

- 유소년
 - 2019 2020 2021 2022
- 청소년
 - 2017 2018 2019 2020 2021 2022
- 성인
 - 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022
- 노인
 - 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022

유소년 평균값

```

child_2019 = data_child.loc[data_child.year == 2019, :]
child_2019_mean = child_2019.MESURE_IEM_020_VALUE.mean()
print("2019 ==> ", child_2019_mean)

```

```

child_2020 = data_child.loc[data_child.year == 2020, :]
child_2020_mean = child_2020.MESURE_IEM_020_VALUE.mean()
print("2020 ==> ", child_2020_mean)

```

```

child_2021 = data_child.loc[data_child.year == 2021, :]
child_2021_mean = child_2021.MESURE_IEM_020_VALUE.mean()
print("2021 ==> ", child_2021_mean)

```

```

child_2022 = data_child.loc[data_child.year == 2022, :]
child_2022_mean = child_2022.MESURE_IEM_020_VALUE.mean()
print("2022 ==> ", child_2022_mean)

```

```

child_average=pd.DataFrame({"year": [2019, 2020, 2021, 2022], "average": [child_2019_mean, child_2020_mean, child_2021_mean, child_2022_mean]})
print(child_average)

```

```

2019 ==> 89.1875
2020 ==> 53.68631578947368
2021 ==> 41.29579359256542
2022 ==> 49.62463989849989
   year  average
0  2019   89.187500
1  2020   53.686316
2  2021   41.295794
3  2022   49.624640

```

청소년

```

teen_2017 = data_teen.loc[data_teen.year == 2017, :]
teen_2017_mean = teen_2017.MESURE_IEM_020_VALUE.mean()
print("2017 ==> ", teen_2017_mean)

```

```

teen_2018 = data_teen.loc[data_teen.year == 2018, :]
teen_2018_mean = teen_2018.MESURE_IEM_020_VALUE.mean()
print("2018 ==> ". teen 2018 mean)

```

```

teen_2019 = data_teen.loc[data_teen.year == 2019, :]
teen_2019_mean = teen_2019.MESURE_IEM_020_VALUE.mean()
print("2019 ==> ", teen_2019_mean)

teen_2020 = data_teen.loc[data_teen.year == 2020, :]
teen_2020_mean = teen_2020.MESURE_IEM_020_VALUE.mean()
print("2020 ==> ", teen_2020_mean)

teen_2021 = data_teen.loc[data_teen.year == 2021, :]
teen_2021_mean = teen_2021.MESURE_IEM_020_VALUE.mean()
print("2021 ==> ", teen_2021_mean)

teen_2022 = data_teen.loc[data_teen.year == 2022, :]
teen_2022_mean = teen_2022.MESURE_IEM_020_VALUE.mean()
print("2022 ==> ", teen_2022_mean)

teen_average=pd.DataFrame({"year":[2017, 2018, 2019, 2020, 2021, 2022], W
                           "average":[teen_2017_mean, teen_2018_mean, teen_2019_mean, teen_2020_mean, teen_2021_mean, teen_2022_mean]})
print(teen_average)

    2017 ==>  36.250940310711364
    2018 ==>  37.24031008874454
    2019 ==>  37.13173179603551
    2020 ==>  30.60764031774971
    2021 ==>  35.14360375179365
    2022 ==>  33.71655079648738
      year    average
0  2017  36.250940
1  2018  37.240310
2  2019  37.131732
3  2020  30.607640
4  2021  35.143604
5  2022  33.716551

# 성인
adult_2013 = data_adult.loc[data_adult.year == 2013, :]
adult_2013_mean = adult_2013.MESURE_IEM_020_VALUE.mean()
print("2013 ==> ", adult_2013_mean)

adult_2014 = data_adult.loc[data_adult.year == 2014, :]
adult_2014_mean = adult_2014.MESURE_IEM_020_VALUE.mean()
print("2014 ==> ", adult_2014_mean)

adult_2015 = data_adult.loc[data_adult.year == 2015, :]
adult_2015_mean = adult_2015.MESURE_IEM_020_VALUE.mean()
print("2015 ==> ", adult_2015_mean)

adult_2016 = data_adult.loc[data_adult.year == 2016, :]
adult_2016_mean = adult_2016.MESURE_IEM_020_VALUE.mean()
print("2016 ==> ", adult_2016_mean)

adult_2017 = data_adult.loc[data_adult.year == 2017, :]
adult_2017_mean = adult_2017.MESURE_IEM_020_VALUE.mean()
print("2017 ==> ", adult_2017_mean)

adult_2018 = data_adult.loc[data_adult.year == 2018, :]
adult_2018_mean = adult_2018.MESURE_IEM_020_VALUE.mean()
print("2018 ==> ", adult_2018_mean)

adult_2019 = data_adult.loc[data_adult.year == 2019, :]
adult_2019_mean = adult_2019.MESURE_IEM_020_VALUE.mean()
print("2019 ==> ", adult_2019_mean)

adult_2020 = data_adult.loc[data_adult.year == 2020, :]
adult_2020_mean = adult_2020.MESURE_IEM_020_VALUE.mean()
print("2020 ==> ", adult_2020_mean)

adult_2021 = data_adult.loc[data_adult.year == 2021, :]
adult_2021_mean = adult_2021.MESURE_IEM_020_VALUE.mean()
print("2021 ==> ", adult_2021_mean)

adult_2022 = data_adult.loc[data_adult.year == 2022, :]
adult_2022_mean = adult_2022.MESURE_IEM_020_VALUE.mean()
print("2022 ==> ", adult_2022_mean)

adult_average=pd.DataFrame({"year":[2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022], W
                           "average":[adult_2013_mean, adult_2014_mean, adult_2015_mean, adult_2016_mean, adult_2017_mean, adult_2018_mean, adult_2019_mean, adult_2020_mean, adult_2021_mean, adult_2022_mean]})
print(adult_average)

    2013 ==>  23.603649869320005
    2014 ==>  26.77175702768417
    2015 ==>  25.62337677043954

```

```

2016 ==> 27.1187532706315
2017 ==> 27.857669144485
2018 ==> 27.19742389496718
2019 ==> 29.419679292999472
2020 ==> 35.8281456043956
2021 ==> 42.445926517571884
2022 ==> 42.65671012269939
   year    average
0  2013    23.603650
1  2014    26.771757
2  2015    25.623377
3  2016    27.118753
4  2017    27.857669
5  2018    27.197424
6  2019    29.419679
7  2020    35.828146
8  2021    42.445927
9  2022    42.656710

# 노인
elder_2013 = data_elder.loc[data_elder.year == 2013, :]
elder_2013_mean = elder_2013.MESURE_IEM_023_VALUE.mean()
print("2013 ==> ", elder_2013_mean)

elder_2014 = data_elder.loc[data_elder.year == 2014, :]
elder_2014_mean = elder_2014.MESURE_IEM_023_VALUE.mean()
print("2014 ==> ", elder_2014_mean)

elder_2015 = data_elder.loc[data_elder.year == 2015, :]
elder_2015_mean = elder_2015.MESURE_IEM_023_VALUE.mean()
print("2015 ==> ", elder_2015_mean)

elder_2016 = data_elder.loc[data_elder.year == 2016, :]
elder_2016_mean = elder_2016.MESURE_IEM_023_VALUE.mean()
print("2016 ==> ", elder_2016_mean)

elder_2017 = data_elder.loc[data_elder.year == 2017, :]
elder_2017_mean = elder_2017.MESURE_IEM_023_VALUE.mean()
print("2017 ==> ", elder_2017_mean)

elder_2018 = data_elder.loc[data_elder.year == 2018, :]
elder_2018_mean = elder_2018.MESURE_IEM_023_VALUE.mean()
print("2018 ==> ", elder_2018_mean)

elder_2019 = data_elder.loc[data_elder.year == 2019, :]
elder_2019_mean = elder_2019.MESURE_IEM_023_VALUE.mean()
print("2019 ==> ", elder_2019_mean)

elder_2020 = data_elder.loc[data_elder.year == 2020, :]
elder_2020_mean = elder_2020.MESURE_IEM_023_VALUE.mean()
print("2020 ==> ", elder_2020_mean)

elder_2021 = data_elder.loc[data_elder.year == 2021, :]
elder_2021_mean = elder_2021.MESURE_IEM_023_VALUE.mean()
print("2021 ==> ", elder_2021_mean)

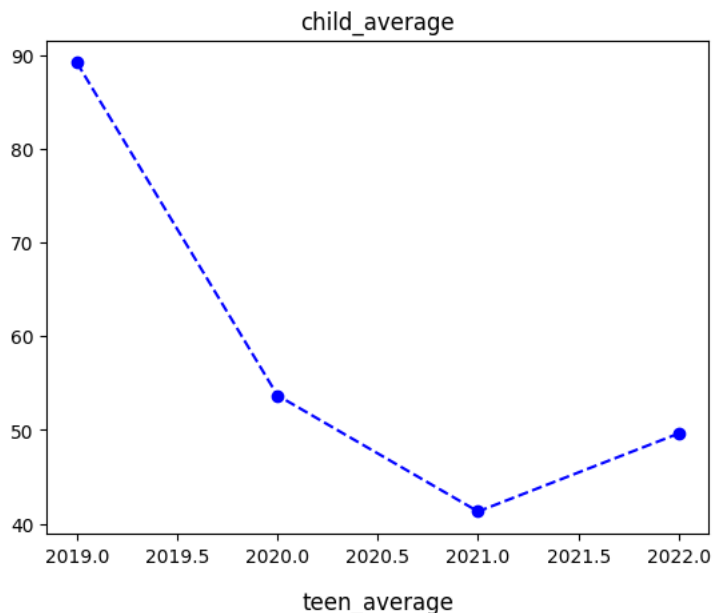
elder_2022 = data_elder.loc[data_elder.year == 2022, :]
elder_2022_mean = elder_2022.MESURE_IEM_023_VALUE.mean()
print("2022 ==> ", elder_2022_mean)

elder_average=pd.DataFrame({"year": [2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022],
                             "average": [elder_2013_mean, elder_2014_mean, elder_2015_mean, elder_2016_mean, elder_2017_mean, elder_2018_mean, elder_2019_mean, elder_2020_mean, elder_2021_mean, elder_2022_mean]})
print(elder_average)

2013 ==> 18.302700675168794
2014 ==> 18.620691697264007
2015 ==> 19.171429994025093
2016 ==> 19.52585992334113
2017 ==> 19.52070303222231
2018 ==> 19.86826586399656
2019 ==> 20.253651426361664
2020 ==> 22.786755892928486
2021 ==> 21.723764980999707
2022 ==> 22.07438661528706
   year    average
0  2013    18.302701
1  2014    18.620692
2  2015    19.171430
3  2016    19.525860
4  2017    19.520703
5  2018    19.868266
6  2019    20.253651
7  2020    22.786756
8  2021    21.723765
9  2022    22.074387

```

```
plt.plot(child_average['year'],child_average['average'],color='blue',linestyle='--',marker='o')
plt.title('child_average')
plt.show()
plt.plot(teen_average['year'],teen_average['average'],color='blue',linestyle='--',marker='o')
plt.title('teen_average')
plt.show()
plt.plot(adult_average['year'],adult_average['average'],color='blue',linestyle='--',marker='o')
plt.title('adult_average')
plt.show()
plt.plot(elder_average['year'],elder_average['average'],color='blue',linestyle='--',marker='o')
plt.title('elder_average')
plt.show()
```



▼ 각 연령별 키 평균

청소년 년도별 키와 체중 13부터 매년 청소년들의 평균 키 체중은 지속적으로 상승했을것이다
유소년 키

```
child_2019_cm = data_child.loc[data_child.year == 2019, :]
child_2019_mean_cm = child_2019_cm.MESURE_IEM_001_VALUE.mean()
print("2019 ==> ", child_2019_mean_cm)
```

```
child_2020_cm = data_child.loc[data_child.year == 2020, :]
child_2020_mean_cm = child_2020_cm.MESURE_IEM_001_VALUE.mean()
print("2020 ==> ", child_2020_mean_cm)
```

```
child_2021_cm = data_child.loc[data_child.year == 2021, :]
child_2021_mean_cm = child_2021_cm.MESURE_IEM_001_VALUE.mean()
print("2021 ==> ", child_2021_mean_cm)
```

```
child_2022_cm = data_child.loc[data_child.year == 2022, :]
child_2022_mean_cm = child_2022_cm.MESURE_IEM_001_VALUE.mean()
print("2022 ==> ", child_2022_mean_cm)
```

```
child_average_cm=pd.DataFrame({"year":[2019, 2020, 2021, 2022], "average":[child_2019_mean_cm, child_2020_mean_cm,child_2021_mean_cm,child_2022_mean_cm]})
print(child_average_cm)
```

```
2019 ==> 150.39375
2020 ==> 154.04715789473684
2021 ==> 151.8235754463194
2022 ==> 151.52162474811553
   year  average
0  2019  150.393750
1  2020  154.047158
2  2021  151.823575
3  2022  151.521625
```


청소년

```
teen_2017_cm = data_teen.loc[data_teen.year == 2017, :]
teen_2017_mean_cm = teen_2017_cm.MESURE_IEM_001_VALUE.mean()
print("2017 ==> ", teen_2017_mean_cm)
```

```
teen_2018_cm = data_teen.loc[data_teen.year == 2018, :]
teen_2018_mean_cm = teen_2018_cm.MESURE_IEM_001_VALUE.mean()
print("2018 ==> ", teen_2018_mean_cm)
```

```
teen_2019_cm = data_teen.loc[data_teen.year == 2019, :]
teen_2019_mean_cm = teen_2019_cm.MESURE_IEM_001_VALUE.mean()
print("2019 ==> ", teen_2019_mean_cm)
```

```
teen_2020_cm = data_teen.loc[data_teen.year == 2020, :]
teen_2020_mean_cm = teen_2020_cm.MESURE_IEM_001_VALUE.mean()
print("2020 ==> ", teen_2020_mean_cm)
```

```
teen_2021_cm = data_teen.loc[data_teen.year == 2021, :]
teen_2021_mean_cm = teen_2021_cm.MESURE_IEM_001_VALUE.mean()
print("2021 ==> ", teen_2021_mean_cm)
```

```
teen_2022_cm = data_teen.loc[data_teen.year == 2022, :]
teen_2022_mean_cm = teen_2022_cm.MESURE_IEM_001_VALUE.mean()
print("2022 ==> ", teen_2022_mean_cm)
```

```
teen_average_cm=pd.DataFrame({"year":[2017, 2018, 2019, 2020, 2021, 2022], W
                             "average":[teen_2017_mean_cm, teen_2018_mean_cm, teen_2019_mean_cm, teen_2020_mean_cm, teen_2021_mean_cm, teen_2022_mean_
print(teen_average_cm)
```

```
2017 ==> 165.20179885527392
2018 ==> 165.89043668935273
2019 ==> 165.73786791697256
2020 ==> 164.02499023310327
2021 ==> 167.22105134217617
2022 ==> 166.36465286985262
   year  average
0  2017  165.201799
1  2018  165.890437
2  2019  165.737868
3  2020  164.024990
4  2021  167.221051
5  2022  166.364653
```

```

# 성인
adult_2013_cm = data_adult.loc[data_adult.year == 2013, :]
adult_2013_mean_cm = adult_2013_cm.MESURE_IEM_001_VALUE.mean()
print("2013 ==> ", adult_2013_mean_cm)

adult_2014_cm = data_adult.loc[data_adult.year == 2014, :]
adult_2014_mean_cm = adult_2014_cm.MESURE_IEM_001_VALUE.mean()
print("2014 ==> ", adult_2014_mean_cm)

adult_2015_cm = data_adult.loc[data_adult.year == 2015, :]
adult_2015_mean_cm = adult_2015_cm.MESURE_IEM_001_VALUE.mean()
print("2015 ==> ", adult_2015_mean_cm)

adult_2016_cm = data_adult.loc[data_adult.year == 2016, :]
adult_2016_mean_cm = adult_2016_cm.MESURE_IEM_001_VALUE.mean()
print("2016 ==> ", adult_2016_mean_cm)

adult_2017_cm = data_adult.loc[data_adult.year == 2017, :]
adult_2017_mean_cm = adult_2017_cm.MESURE_IEM_001_VALUE.mean()
print("2017 ==> ", adult_2017_mean_cm)

adult_2018_cm = data_adult.loc[data_adult.year == 2018, :]
adult_2018_mean_cm = adult_2018_cm.MESURE_IEM_001_VALUE.mean()
print("2018 ==> ", adult_2018_mean_cm)

adult_2019_cm = data_adult.loc[data_adult.year == 2019, :]
adult_2019_mean_cm = adult_2019_cm.MESURE_IEM_001_VALUE.mean()
print("2019 ==> ", adult_2019_mean_cm)

adult_2020_cm = data_adult.loc[data_adult.year == 2020, :]
adult_2020_mean_cm = adult_2020_cm.MESURE_IEM_001_VALUE.mean()
print("2020 ==> ", adult_2020_mean_cm)

adult_2021_cm = data_adult.loc[data_adult.year == 2021, :]
adult_2021_mean_cm = adult_2021_cm.MESURE_IEM_001_VALUE.mean()
print("2021 ==> ", adult_2021_mean_cm)

adult_2022_cm = data_adult.loc[data_adult.year == 2022, :]
adult_2022_mean_cm = adult_2022_cm.MESURE_IEM_001_VALUE.mean()
print("2022 ==> ", adult_2022_mean_cm)

adult_average_cm=pd.DataFrame({"year":[2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022], W
                                "average":[adult_2013_mean_cm, adult_2014_mean_cm, adult_2015_mean_cm, adult_2016_mean_cm, adult_2017_mean_cm, adult_2018
print(adult_average_cm)

2013 ==> 163.4583153743867
2014 ==> 163.7495691677686
2015 ==> 164.3033518579815
2016 ==> 165.1718926054263
2017 ==> 164.90366331345146
2018 ==> 165.25048927789933
2019 ==> 166.7781966484173
2020 ==> 169.89055213079604
2021 ==> 170.9907747603834
2022 ==> 171.46959355828218
   year  average
0  2013  163.458315
1  2014  163.749569
2  2015  164.303352
3  2016  165.171893
4  2017  164.903663
5  2018  165.250489
6  2019  166.778197
7  2020  169.890552
8  2021  170.990775
9  2022  171.469594

```

```

# 노인
elder_2013_cm = data_elder.loc[data_elder.year == 2013, :]
elder_2013_mean_cm = elder_2013_cm.MESURE_IEM_001_VALUE.mean()
print("2013 ==> ", elder_2013_mean_cm)

elder_2014_cm = data_elder.loc[data_elder.year == 2014, :]
elder_2014_mean_cm = elder_2014_cm.MESURE_IEM_001_VALUE.mean()
print("2014 ==> ", elder_2014_mean_cm)

elder_2015_cm = data_elder.loc[data_elder.year == 2015, :]
elder_2015_mean_cm = elder_2015_cm.MESURE_IEM_001_VALUE.mean()
print("2015 ==> ", elder_2015_mean_cm)

elder_2016_cm = data_elder.loc[data_elder.year == 2016, :]
elder_2016_mean_cm = elder_2016_cm.MESURE_IEM_001_VALUE.mean()
print("2016 ==> ", elder_2016_mean_cm)

elder_2017_cm = data_elder.loc[data_elder.year == 2017, :]
elder_2017_mean_cm = elder_2017_cm.MESURE_IEM_001_VALUE.mean()
print("2017 ==> ", elder_2017_mean_cm)

elder_2018_cm = data_elder.loc[data_elder.year == 2018, :]
elder_2018_mean_cm = elder_2018_cm.MESURE_IEM_001_VALUE.mean()
print("2018 ==> ", elder_2018_mean_cm)

elder_2019_cm = data_elder.loc[data_elder.year == 2019, :]
elder_2019_mean_cm = elder_2019_cm.MESURE_IEM_001_VALUE.mean()
print("2019 ==> ", elder_2019_mean_cm)

elder_2020_cm = data_elder.loc[data_elder.year == 2020, :]
elder_2020_mean_cm = elder_2020_cm.MESURE_IEM_001_VALUE.mean()
print("2020 ==> ", elder_2020_mean_cm)

elder_2021_cm = data_elder.loc[data_elder.year == 2021, :]
elder_2021_mean_cm = elder_2021_cm.MESURE_IEM_001_VALUE.mean()
print("2021 ==> ", elder_2021_mean_cm)

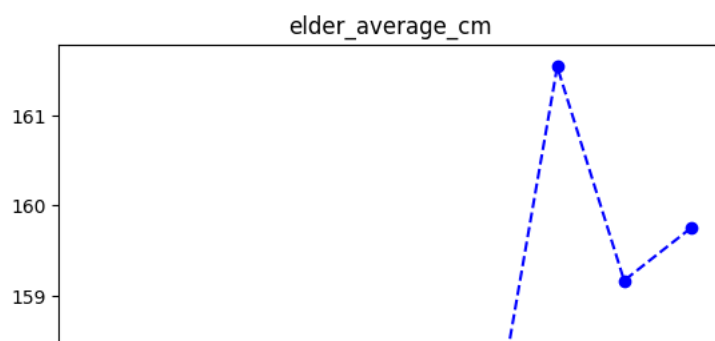
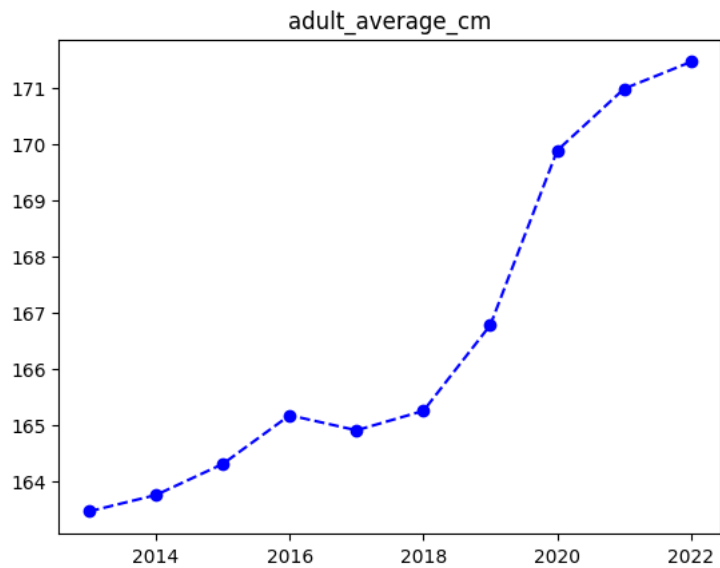
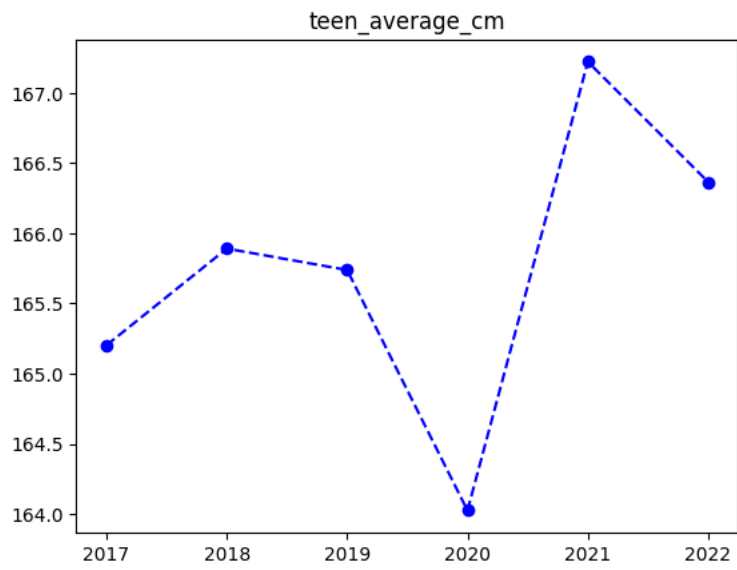
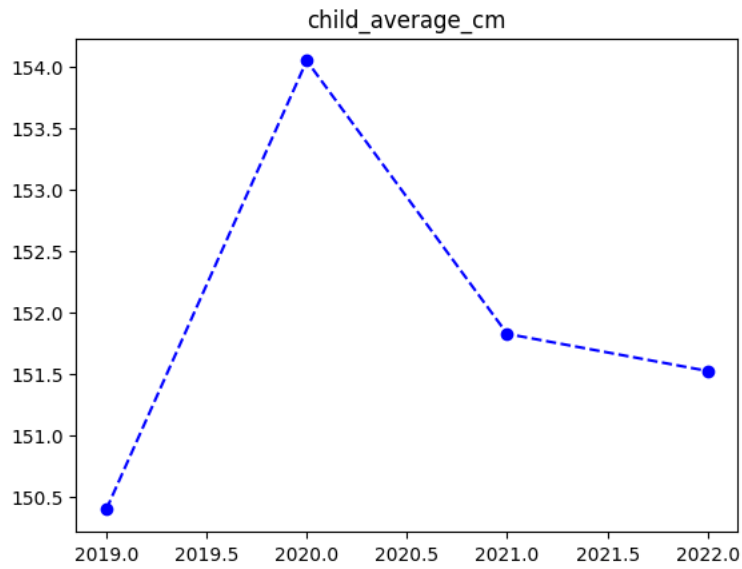
elder_2022_cm = data_elder.loc[data_elder.year == 2022, :]
elder_2022_mean_cm = elder_2022_cm.MESURE_IEM_001_VALUE.mean()
print("2022 ==> ", elder_2022_mean_cm)

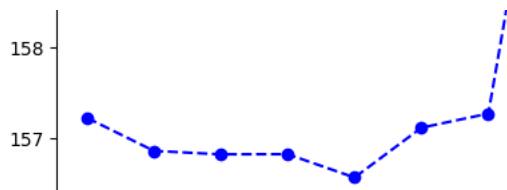
elder_average_cm=pd.DataFrame({"year": [2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022], W
                                "average": [elder_2013_mean_cm, elder_2014_mean_cm, elder_2015_mean_cm, elder_2016_mean_cm, elder_2017_mean_cm, elder_2018
print(elder_average_cm)

2013 ==> 157.22320580145038
2014 ==> 156.8573611275613
2015 ==> 156.81887074287988
2016 ==> 156.82050873612425
2017 ==> 156.5619309635025
2018 ==> 157.11400577431044
2019 ==> 157.2703572597198
2020 ==> 161.53604674390732
2021 ==> 159.16631589203936
2022 ==> 159.74887566460174
   year  average
0  2013  157.223206
1  2014  156.857361
2  2015  156.818871
3  2016  156.820509
4  2017  156.561931
5  2018  157.114006
6  2019  157.270357
7  2020  161.536047
8  2021  159.166316
9  2022  159.748876

plt.plot(child_average_cm['year'], child_average_cm['average'], color='blue', linestyle='--', marker='o')
plt.title('child_average_cm')
plt.show()
plt.plot(teen_average_cm['year'], teen_average_cm['average'], color='blue', linestyle='--', marker='o')
plt.title('teen_average_cm')
plt.show()
plt.plot(adult_average_cm['year'], adult_average_cm['average'], color='blue', linestyle='--', marker='o')
plt.title('adult_average_cm')
plt.show()
plt.plot(elder_average_cm['year'], elder_average_cm['average'], color='blue', linestyle='--', marker='o')
plt.title('elder_average_cm')
plt.show()

```





▼ 각 연령별 몸무게 년별 평균

유소년 평균값

```
child_2019_kg = data_child.loc[data_child.year == 2019, :]
child_2019_mean_kg = child_2019_kg.MESURE_IEM_002_VALUE.mean()
print("2019 ==> ", child_2019_mean_kg)
```

```
child_2020_kg = data_child.loc[data_child.year == 2020, :]
child_2020_mean_kg = child_2020_kg.MESURE_IEM_002_VALUE.mean()
print("2020 ==> ", child_2020_mean_kg)
```

```
child_2021_kg = data_child.loc[data_child.year == 2021, :]
child_2021_mean_kg = child_2021_kg.MESURE_IEM_002_VALUE.mean()
print("2021 ==> ", child_2021_mean_kg)
```

```
child_2022_kg = data_child.loc[data_child.year == 2022, :]
child_2022_mean_kg = child_2022_kg.MESURE_IEM_002_VALUE.mean()
print("2022 ==> ", child_2022_mean_kg)
```

```
child_average_kg=pd.DataFrame({"year":[2019, 2020, 2021, 2022], "average":[child_2019_mean_kg, child_2020_mean_kg,child_2021_mean_kg,child_2022_mean_kg])
print(child_average_kg)
```

```
2019 ==> 46.056250000000006
2020 ==> 50.35052631578947
2021 ==> 49.103396918562
2022 ==> 47.499916411672515
   year  average
0  2019  46.056250
1  2020  50.350526
2  2021  49.103397
3  2022  47.499916
```

청소년

```
teen_2017_kg = data_teen.loc[data_teen.year == 2017, :]
teen_2017_mean_kg = teen_2017_kg.MESURE_IEM_002_VALUE.mean()
print("2017 ==> ", teen_2017_mean_kg)
```

```
teen_2018_kg = data_teen.loc[data_teen.year == 2018, :]
teen_2018_mean_kg = teen_2018_kg.MESURE_IEM_002_VALUE.mean()
print("2018 ==> ", teen_2018_mean_kg)
```

```
teen_2019_kg = data_teen.loc[data_teen.year == 2019, :]
teen_2019_mean_kg = teen_2019_kg.MESURE_IEM_002_VALUE.mean()
print("2019 ==> ", teen_2019_mean_kg)
```

```
teen_2020_kg = data_teen.loc[data_teen.year == 2020, :]
teen_2020_mean_kg = teen_2020_kg.MESURE_IEM_002_VALUE.mean()
print("2020 ==> ", teen_2020_mean_kg)
```

```
teen_2021_kg = data_teen.loc[data_teen.year == 2021, :]
teen_2021_mean_kg = teen_2021_kg.MESURE_IEM_002_VALUE.mean()
print("2021 ==> ", teen_2021_mean_kg)
```

```
teen_2022_kg = data_teen.loc[data_teen.year == 2022, :]
teen_2022_mean_kg = teen_2022_kg.MESURE_IEM_002_VALUE.mean()
print("2022 ==> ", teen_2022_mean_kg)
```

```
teen_average_kg=pd.DataFrame({"year":[2017, 2018, 2019, 2020, 2021, 2022], "average":[teen_2017_mean_kg, teen_2018_mean_kg, teen_2019_mean_kg, teen_2020_mean_kg, teen_2021_mean_kg, teen_2022_mean_kg])
print(teen_average_kg)
```

```
2017 ==> 60.64573589533933
2018 ==> 60.97032601725243
2019 ==> 60.90915754521791
2020 ==> 59.839828102617524
2021 ==> 63.19776992265425
2022 ==> 61.77565842316635
   year  average
0  2017  60.645736
1  2018  60.970326
2  2019  60.909158
3  2020  59.839828
```

```

4 2021 63.197770
5 2022 61.775658

# 성인
adult_2013_kg = data_adult.loc[data_adult.year == 2013, :]
adult_2013_mean_kg = adult_2013_kg.MESURE_IEM_002_VALUE.mean()
print("2013 ==> ", adult_2013_mean_kg)

adult_2014_kg = data_adult.loc[data_adult.year == 2014, :]
adult_2014_mean_kg = adult_2014_kg.MESURE_IEM_002_VALUE.mean()
print("2014 ==> ", adult_2014_mean_kg)

adult_2015_kg = data_adult.loc[data_adult.year == 2015, :]
adult_2015_mean_kg = adult_2015_kg.MESURE_IEM_002_VALUE.mean()
print("2015 ==> ", adult_2015_mean_kg)

adult_2016_kg = data_adult.loc[data_adult.year == 2016, :]
adult_2016_mean_kg = adult_2016_kg.MESURE_IEM_002_VALUE.mean()
print("2016 ==> ", adult_2016_mean_kg)

adult_2017_kg = data_adult.loc[data_adult.year == 2017, :]
adult_2017_mean_kg = adult_2017_kg.MESURE_IEM_002_VALUE.mean()
print("2017 ==> ", adult_2017_mean_kg)

adult_2018_kg = data_adult.loc[data_adult.year == 2018, :]
adult_2018_mean_kg = adult_2018_kg.MESURE_IEM_002_VALUE.mean()
print("2018 ==> ", adult_2018_mean_kg)

adult_2019_kg = data_adult.loc[data_adult.year == 2019, :]
adult_2019_mean_kg = adult_2019_kg.MESURE_IEM_002_VALUE.mean()
print("2019 ==> ", adult_2019_mean_kg)

adult_2020_kg = data_adult.loc[data_adult.year == 2020, :]
adult_2020_mean_kg = adult_2020_kg.MESURE_IEM_002_VALUE.mean()
print("2020 ==> ", adult_2020_mean_kg)

adult_2021_kg = data_adult.loc[data_adult.year == 2021, :]
adult_2021_mean_kg = adult_2021_kg.MESURE_IEM_002_VALUE.mean()
print("2021 ==> ", adult_2021_mean_kg)

adult_2022_kg = data_adult.loc[data_adult.year == 2022, :]
adult_2022_mean_kg = adult_2022_kg.MESURE_IEM_002_VALUE.mean()
print("2022 ==> ", adult_2022_mean_kg)

adult_average_kg=pd.DataFrame({"year":[2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022], W
                                "average":[adult_2013_mean_kg, adult_2014_mean_kg, adult_2015_mean_kg, adult_2016_mean_kg, adult_2017_mean_kg, adult_2018
print(adult_average_kg)

2013 ==> 62.28185107065891
2014 ==> 62.52843706010322
2015 ==> 63.325003693315445
2016 ==> 64.35649553308346
2017 ==> 64.08023902786432
2018 ==> 64.44047824945297
2019 ==> 64.92107384042463
2020 ==> 68.92953732243366
2021 ==> 70.21178115015974
2022 ==> 71.07181365030675
   year  average
0  2013  62.281851
1  2014  62.528437
2  2015  63.325004
3  2016  64.356496
4  2017  64.080239
5  2018  64.440478
6  2019  64.921074
7  2020  68.929537
8  2021  70.211781
9  2022  71.071814

```