

▼ 레이더 차트 만들기

▼ 1. 합쳐진 데이터 불러오기

```
import pandas as pd
raw_data = pd.read_csv(r"E:\Downloads\combined_data.csv")
```

▼ 2. 데이터 기본정보 확인하기

- 총 데이터는 1727278행 51열이 있음

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



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1727278 entries, 0 to 1727277
Data columns (total 47 columns):
#   Column                Dtype
---  -
0    MBER_SEQ_NO_VALUE      object
1    MESURE_SEQ_NO          int64
2    CINTER_NM              object
3    AGRDE_FLAG_NM          object
4    MESURE_PLACE_FLAG_NM  object
5    MFSURF_AGE_CO          float64
```

▼ 3. 데이터 전처리

3.1 불필요한 행 제거 및 결측치 처리

- 불필요한 정보인 회원번호, 측정회원번호, 입력구분명, 측정장소구분명은 제거
- 향후 서비스 이용자가 스스로 측정 불가한 측정항목 제거(3, 4, 5, 6, 7, 8, 10, 14, 15, 16, 17, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40)
- data_child 9번 열, data_teen 9번, 13번, 15번 열, data_elder 26번 열을 되살림
- 기본정보를 토대로 결측치가 많은 측정값은(24) 제거
- 중요 변수인 나이와 운동처방에 결측이 있는 경우, 해당되는 행 제거
- 성인데이터 중, 등급이 '금상', '은상', '동상'인 경우 행 제거
- 성별(남:0, 여:1), 운동등급(1등급:1, 2등급:2, 3등급:3, 참가증:4)로 인코딩

```
# 회원일련번호값, 측정일련번호, 입력구분명 제거
raw_data.drop(['MBER_SEQ_NO_VALUE', 'MESURE_SEQ_NO', 'INPT_FLAG_NM'], axis = 1, inplace=True)

# 향후 서비스 이용자가 이용하지 못하는 측정값 제거
col = []
num = [3, 4, 5, 6, 7, 8, 14, 16, 17, 28, 30, 31, 32, 33, 34, 35, 36, 37, 40, 41]
for i in num:
    if len(str(i))==1:
        dt = 'MESURE_IEM_00' + str(i) + '_VALUE'
        col.append(dt)
    else:
        dt = 'MESURE_IEM_0' + str(i) + '_VALUE'
        col.append(dt)

raw_data.drop(col, axis = 1, inplace=True)

# 나이, 운동처방에 결측이 있는 경우, 행을 제거
raw_data.dropna(subset=['MVM_PRSCRPTN_CN', 'MESURE_AGE_CO'], inplace=True)

# 성인 데이터 중 등급이 '금상', '은상', '동상'인 행 제거
raw_data.drop(raw_data[(raw_data['CRTFC_FLAG_NM']=='금상') | (raw_data['CRTFC_FLAG_NM']=='동상')].index, inplace=True)

# 결측치 제거 여부 확인하기
print(raw_data['MESURE_AGE_CO'].isnull().sum())
print(raw_data['MVM_PRSCRPTN_CN'].isnull().sum())

# raw 데이터 결측치 제거 확인
#raw_data.info()
print(raw_data.columns)

Index(['CINTER_NM', 'AGRDE_FLAG_NM', 'MESURE_AGE_CO', 'CRTFC_FLAG_NM',
      'MESURE_DE', 'SEXOSTN_FLAG_CD', 'MESURE_IEM_001_VALUE',
      'MESURE_IEM_002_VALUE', 'MESURE_IEM_009_VALUE', 'MESURE_IEM_010_VALUE',
      'MESURE_IEM_012_VALUE', 'MESURE_IEM_013_VALUE', 'MESURE_IEM_015_VALUE',
      'MESURE_IEM_018_VALUE', 'MESURE_IEM_019_VALUE', 'MESURE_IEM_020_VALUE',
      'MESURE_IEM_021_VALUE', 'MESURE_IEM_022_VALUE', 'MESURE_IEM_023_VALUE',
      'MESURE_IEM_025_VALUE', 'MESURE_IEM_026_VALUE', 'MESURE_IEM_027_VALUE',
      'MVM_PRSCRPTN_CN'],
      dtype='object')

# 성별 인코딩하기
raw_data['SEXOSTN_FLAG_CD'] = raw_data['SEXOSTN_FLAG_CD'].map({'M': 0, 'F': 1})

# 등급표 인코딩하기
raw_data['CRTFC_FLAG_NM'] = raw_data['CRTFC_FLAG_NM'].map({'참가증':4, '3등급':3, '2등급':2, '1등급':1})
```

- 연령별 특성을 고려한 알고리즘 모델을 학습시키기 위하여, 연령별로 데이터를 분할
- 데이터를 분할한 뒤, 연령별 측정하지 않는 측정항목은 제거

[illegible]

▼ 3.3 연령별 데이터 내 결측치 대체

- ```
import numpy as np
#유소년(child) 결측치 대체
언더샘플링
데이터 보존량이 적은 MESURE_IEM_020_VALUE을 기준으로 언더 샘플링
data_child.dropna(subset=['MESURE_IEM_020_VALUE'], inplace=True)

신장, 체중에 결측치가 있는 경우, 평균대체로 결측치 처리
col_to_fill = ['MESURE_IEM_001_VALUE', 'MESURE_IEM_002_VALUE']
for col_name in col_to_fill:
 data_child[col_name].fillna(data_child[col_name].mean())

BMI (18) 계산(체중(kg)/ 신장(m)**2)
data_child['MESURE_IEM_018_VALUE'].fillna(data_child['MESURE_IEM_002_VALUE'] / ((data_child['MESURE_IEM_001_VALUE']/100)**2), inplace=True)

측정 데이터에 결측치가 있는 경우, 핫덱 대체로 결측치 처리
data_child = hot_deck(data_child)

결측치 확인
data_child.info()

<class 'pandas.core.frame.DataFrame'>
Index: 33513 entries, 1211965 to 1727038
```

```

Data columns (total 14 columns):
Column Non-Null Count Dtype
--- -
0 CNTER_NM 33513 non-null object
1 AGRDE_FLAG_NM 33513 non-null object
2 MESURE_AGE_CO 33513 non-null float64
3 CRTFC_FLAG_NM 33513 non-null int64
4 MESURE_DE 33513 non-null int64
5 SEXDSTN_FLAG_CD 33513 non-null int64
6 MESURE_IEM_001_VALUE 33513 non-null float64
7 MESURE_IEM_002_VALUE 33513 non-null float64
8 MESURE_IEM_009_VALUE 33513 non-null float64
9 MESURE_IEM_012_VALUE 33513 non-null float64
10 MESURE_IEM_018_VALUE 33513 non-null float64
11 MESURE_IEM_020_VALUE 33513 non-null float64
12 MESURE_IEM_022_VALUE 33513 non-null float64
13 MVM_PRSCRIPTN_CN 33513 non-null object
dtypes: float64(8), int64(3), object(3)
memory usage: 3.8+ MB
C:\Users\WPublic\Documents\WESTsoft\WCreatorTemp\Wipykernel_24528\W2814172312.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data_child.dropna(subset=['MESURE_IEM_020_VALUE'], inplace=True)
C:\Users\WPublic\Documents\WESTsoft\WCreatorTemp\Wipykernel_24528\W2814172312.py:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data_child['MESURE_IEM_018_VALUE'].fillna(data_child['MESURE_IEM_002_VALUE'] / ((data_child['MESURE_IEM_001_VALUE']/100)**2), inplace=True)

청소년(teen) 결측치 대체
언더샘플링
데이터 보존량이 적은 MESURE_IEM_022_VALUE를 기준으로 언더 샘플링
data_teen.dropna(subset=['MESURE_IEM_022_VALUE'], inplace=True)

신장, 체중에 결측치가 있는 경우, 평균대체로 결측치 처리
col_to_fill = ['MESURE_IEM_001_VALUE', 'MESURE_IEM_002_VALUE']
for col_name in col_to_fill:
 data_teen[col_name].fillna(data_teen[col_name].mean())

BMI(18) 계산(체중(kg)/ 신장(m)**2)
data_teen['MESURE_IEM_018_VALUE'].fillna(data_teen['MESURE_IEM_002_VALUE'] / ((data_teen['MESURE_IEM_001_VALUE']/100)**2), inplace=True)

측정 데이터에 결측치가 있는 경우, 핫덱 대체로 결측치 처리
data_teen = hot_deck(data_teen)

결측치 확인
data_teen.info()

C:\Users\WPublic\Documents\WESTsoft\WCreatorTemp\Wipykernel_24528\W2620873156.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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.dropna(subset=['MESURE_IEM_022_VALUE'], inplace=True)
C:\Users\WPublic\Documents\WESTsoft\WCreatorTemp\Wipykernel_24528\W2620873156.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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['MESURE_IEM_018_VALUE'].fillna(data_teen['MESURE_IEM_002_VALUE'] / ((data_teen['MESURE_IEM_001_VALUE']/100)**2), inplace=True)
<class 'pandas.core.frame.DataFrame'>
Index: 235788 entries, 248803 to 1726553
Data columns (total 16 columns):
Column Non-Null Count Dtype
--- -
0 CNTER_NM 235788 non-null object
1 AGRDE_FLAG_NM 235788 non-null object
2 MESURE_AGE_CO 235788 non-null float64
3 CRTFC_FLAG_NM 235788 non-null int64
4 MESURE_DE 235788 non-null int64
5 SEXDSTN_FLAG_CD 235788 non-null int64
6 MESURE_IEM_001_VALUE 235788 non-null float64
7 MESURE_IEM_002_VALUE 235788 non-null float64
8 MESURE_IEM_009_VALUE 235788 non-null float64
9 MESURE_IEM_012_VALUE 235788 non-null float64
10 MESURE_IEM_013_VALUE 235788 non-null float64
11 MESURE_IEM_015_VALUE 235788 non-null float64
12 MESURE_IEM_018_VALUE 235788 non-null float64
13 MESURE_IEM_020_VALUE 235788 non-null float64
14 MESURE_IEM_022_VALUE 235788 non-null float64
15 MVM_PRSCRIPTN_CN 235788 non-null object
dtypes: float64(10), int64(3), object(3)
memory usage: 30.6+ MB

```

```

성인(adult) 결측치 대체
언더샘플링
데이터 보존량이 적은 MESURE_IEM_020_VALUE를 기준으로 언더 샘플링
data_adult.dropna(subset=['MESURE_IEM_020_VALUE'], inplace=True)

신장, 체중에 결측치가 있는 경우, 평균대체로 결측치 처리
col_to_fill = ['MESURE_IEM_001_VALUE', 'MESURE_IEM_002_VALUE']
for col_name in col_to_fill:
 data_adult[col_name].fillna(data_adult[col_name].mean())

BMI(18) 계산(체중(kg)/ 신장(m)**2)
data_adult['MESURE_IEM_018_VALUE'].fillna(data_adult['MESURE_IEM_002_VALUE'] / ((data_adult['MESURE_IEM_001_VALUE']/100)**2), inplace=True)

측정 데이터에 결측치가 있는 경우, 핫덱 대체로 결측치 처리
data_adult = hot_deck(data_adult)

결측치 확인
data_adult.info()

C:\Users\WPublic\Documents\WESTsoft\WCreatorTemp\Wipykernel_24528\W907542849.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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.dropna(subset=['MESURE_IEM_020_VALUE'], inplace=True)
C:\Users\WPublic\Documents\WESTsoft\WCreatorTemp\Wipykernel_24528\W907542849.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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['MESURE_IEM_018_VALUE'].fillna(data_adult['MESURE_IEM_002_VALUE'] / ((data_adult['MESURE_IEM_001_VALUE']/100)**2), inplace=True)
<class 'pandas.core.frame.DataFrame'>
Index: 423847 entries, 1 to 1727222
Data columns (total 15 columns):
Column Non-Null Count Dtype
--- ---
0 CNTER_NM 423847 non-null object
1 AGRDE_FLAG_NM 423847 non-null object
2 MESURE_AGE_CO 423847 non-null float64
3 CRTFC_FLAG_NM 423847 non-null int64
4 MESURE_DE 423847 non-null int64
5 SEXDSTN_FLAG_CD 423847 non-null int64
6 MESURE_IEM_001_VALUE 423847 non-null float64
7 MESURE_IEM_002_VALUE 423847 non-null float64
8 MESURE_IEM_012_VALUE 423847 non-null float64
9 MESURE_IEM_018_VALUE 423847 non-null float64
10 MESURE_IEM_019_VALUE 423847 non-null float64
11 MESURE_IEM_020_VALUE 423847 non-null float64
12 MESURE_IEM_021_VALUE 423847 non-null float64
13 MESURE_IEM_022_VALUE 423847 non-null float64
14 MVM_PRSCRIPTN_CN 423847 non-null object
dtypes: float64(9), int64(3), object(3)
memory usage: 51.7+ MB

노인(elder) 결측치 대체
언더샘플링
데이터 보존량이 적은 MESURE_IEM_027_VALUE를 기준으로 언더 샘플링
data_elder.dropna(subset=['MESURE_IEM_027_VALUE'], inplace=True)

신장, 체중에 결측치가 있는 경우, 평균대체로 결측치 처리
col_to_fill = ['MESURE_IEM_001_VALUE', 'MESURE_IEM_002_VALUE']
for col_name in col_to_fill:
 data_elder[col_name].fillna(data_elder[col_name].mean())

BMI(18) 계산(체중(kg)/ 신장(m)**2)
data_elder['MESURE_IEM_018_VALUE'].fillna(data_elder['MESURE_IEM_002_VALUE'] / ((data_elder['MESURE_IEM_001_VALUE']/100)**2), inplace=True)

측정 데이터에 결측치가 있는 경우, 핫덱 대체로 결측치 처리
data_elder = hot_deck(data_elder)

결측치 확인
data_elder.info()

<class 'pandas.core.frame.DataFrame'>
Index: 237241 entries, 0 to 172722
Data columns (total 15 columns):
Column Non-Null Count Dtype
--- ---
0 CNTER_NM 237241 non-null object
1 AGRDE_FLAG_NM 237241 non-null object
2 MESURE_AGE_CO 237241 non-null float64
3 CRTFC_FLAG_NM 237241 non-null int64
4 MESURE_DE 237241 non-null int64
5 SEXDSTN_FLAG_CD 237241 non-null int64
6 MESURE_IEM_001_VALUE 237241 non-null float64
7 MESURE_IEM_002_VALUE 237241 non-null float64
8 MESURE_IEM_012_VALUE 237241 non-null float64
9 MESURE_IEM_018_VALUE 237241 non-null float64

```

```

10 MEASURE_IEM_023_VALUE 237241 non-null float64
11 MEASURE_IEM_025_VALUE 237241 non-null float64
12 MEASURE_IEM_026_VALUE 237241 non-null float64
13 MEASURE_IEM_027_VALUE 237241 non-null float64
14 MVM_PRSCRIPTN_CN 237241 non-null object
dtypes: float64(9), int64(3), object(3)
memory usage: 29.0+ MB
C:\Users\Public\Documents\WESTsoft\WCreatorTemp\Wipykernel_24528\W2291520091.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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.dropna(subset=['MEASURE_IEM_027_VALUE'], inplace=True)
C:\Users\Public\Documents\WESTsoft\WCreatorTemp\Wipykernel_24528\W2291520091.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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['MEASURE_IEM_018_VALUE'].fillna(data_elder['MEASURE_IEM_002_VALUE'] / ((data_elder['MEASURE_IEM_001_VALUE']/100)**2), inplace=True)

```

```

print("유소년 데이터 수", data_child.shape)
print("청소년 데이터 수", data_teen.shape)
print("성인 데이터 수", data_adult.shape)
print("노인 데이터 수", data_elder.shape)

```

```

유소년 데이터 수 (33513, 14)
청소년 데이터 수 (235788, 16)
성인 데이터 수 (423847, 15)
노인 데이터 수 (237241, 15)

```

## ▼ 이상치 확인하기

박스플롯(boxplot과 상관도를 통해 IQR을 초과하는 데이터 검출하기)

```
이상치 확인하기
```

```
boxplot
```

```
print(data_child[6:21].head(10))
```

|         | CNTNR_NM | AGRDE_FLAG_NM | MEASURE_AGE_CO | CRTFC_FLAG_NM | MEASURE_DE | W |
|---------|----------|---------------|----------------|---------------|------------|---|
| 1211979 | 안동       | 유소년           | 11.0           | 3             | 20191210   |   |
| 1211982 | 안동       | 유소년           | 11.0           | 2             | 20191210   |   |
| 1211988 | 안동       | 유소년           | 12.0           | 4             | 20191210   |   |
| 1211996 | 삼척       | 유소년           | 12.0           | 4             | 20191210   |   |
| 1211998 | 삼척       | 유소년           | 11.0           | 4             | 20191210   |   |
| 1211999 | 삼척       | 유소년           | 12.0           | 4             | 20191210   |   |
| 1212000 | 삼척       | 유소년           | 12.0           | 4             | 20191210   |   |
| 1212001 | 삼척       | 유소년           | 12.0           | 4             | 20191210   |   |
| 1212002 | 삼척       | 유소년           | 11.0           | 4             | 20191210   |   |
| 1212003 | 삼척       | 유소년           | 12.0           | 4             | 20191210   |   |

|         | SEXSTN_FLAG_CD | MEASURE_IEM_001_VALUE | MEASURE_IEM_002_VALUE | W |
|---------|----------------|-----------------------|-----------------------|---|
| 1211979 | 0              | 160.4                 | 41.6                  |   |
| 1211982 | 1              | 160.4                 | 51.2                  |   |
| 1211988 | 0              | 156.8                 | 45.5                  |   |
| 1211996 | 0              | 153.0                 | 57.5                  |   |
| 1211998 | 0              | 140.7                 | 35.3                  |   |
| 1211999 | 0              | 165.1                 | 50.8                  |   |
| 1212000 | 0              | 159.7                 | 50.1                  |   |
| 1212001 | 0              | 152.0                 | 65.6                  |   |
| 1212002 | 0              | 148.1                 | 50.9                  |   |
| 1212003 | 0              | 151.1                 | 41.9                  |   |

|         | MEASURE_IEM_009_VALUE | MEASURE_IEM_012_VALUE | MEASURE_IEM_018_VALUE | W |
|---------|-----------------------|-----------------------|-----------------------|---|
| 1211979 | 80.0                  | 6.3                   | 16.2                  |   |
| 1211982 | 31.0                  | 16.1                  | 19.9                  |   |
| 1211988 | 80.0                  | 3.3                   | 18.5                  |   |
| 1211996 | 9.0                   | -1.1                  | 24.6                  |   |
| 1211998 | 4.0                   | -1.9                  | 17.8                  |   |
| 1211999 | 49.0                  | -3.3                  | 18.6                  |   |
| 1212000 | 26.0                  | -2.0                  | 19.6                  |   |
| 1212001 | 23.0                  | 5.0                   | 28.4                  |   |
| 1212002 | 14.0                  | -0.2                  | 23.2                  |   |
| 1212003 | 26.0                  | -12.4                 | 18.4                  |   |

|         | MEASURE_IEM_020_VALUE | MEASURE_IEM_022_VALUE | W |
|---------|-----------------------|-----------------------|---|
| 1211979 | 101.0                 | 168.0                 |   |
| 1211982 | 81.0                  | 185.0                 |   |
| 1211988 | 152.0                 | 173.0                 |   |
| 1211996 | 70.0                  | 132.0                 |   |
| 1211998 | 95.0                  | 152.0                 |   |
| 1211999 | 121.0                 | 189.0                 |   |
| 1212000 | 112.0                 | 197.0                 |   |
| 1212001 | 65.0                  | 164.0                 |   |
| 1212002 | 68.0                  | 151.0                 |   |
| 1212003 | 113.0                 | 133.0                 |   |

MVM\_PRSCRIPTN\_CN

```

1211979 준비운동:걷기, 동적 스트레칭 루틴프로그램 / 본운동:요가 및 필라테스 루틴프로그램...
1211982 준비운동:동적 스트레칭 루틴프로그램 / 본운동:순간반응 콘 찍기, 앉았다 일어나며 점...
1211988 준비운동:걷기, 동적 스트레칭 루틴프로그램 / 본운동:요가 및 필라테스 루틴프로그램...
1211996 준비운동:동적 스트레칭 루틴프로그램 / 본운동:짐볼 이용 근력운동 루틴 프로그램, 의...
1211998 준비운동:동적 스트레칭 루틴프로그램 / 본운동:짐볼 이용 근력운동 루틴 프로그램, 의...
1211999 준비운동:동적 스트레칭 루틴프로그램 / 본운동:폼롤러 이용 유연성운동 루틴 프로그램...
1212000 준비운동:동적 스트레칭 루틴프로그램 / 본운동:폼롤러 이용 유연성운동 루틴 프로그램...
1212001 준비운동:동적 스트레칭 루틴프로그램 / 본운동:폼롤러 이용 유연성운동 루틴 프로그램...
1212002 준비운동:동적 스트레칭 루틴프로그램 / 본운동:폼롤러 이용 유연성운동 루틴 프로그램...

```

```

import matplotlib.pyplot as plt
유소년 데이터 기준 박스플롯
그림 초기화
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(10, 10))

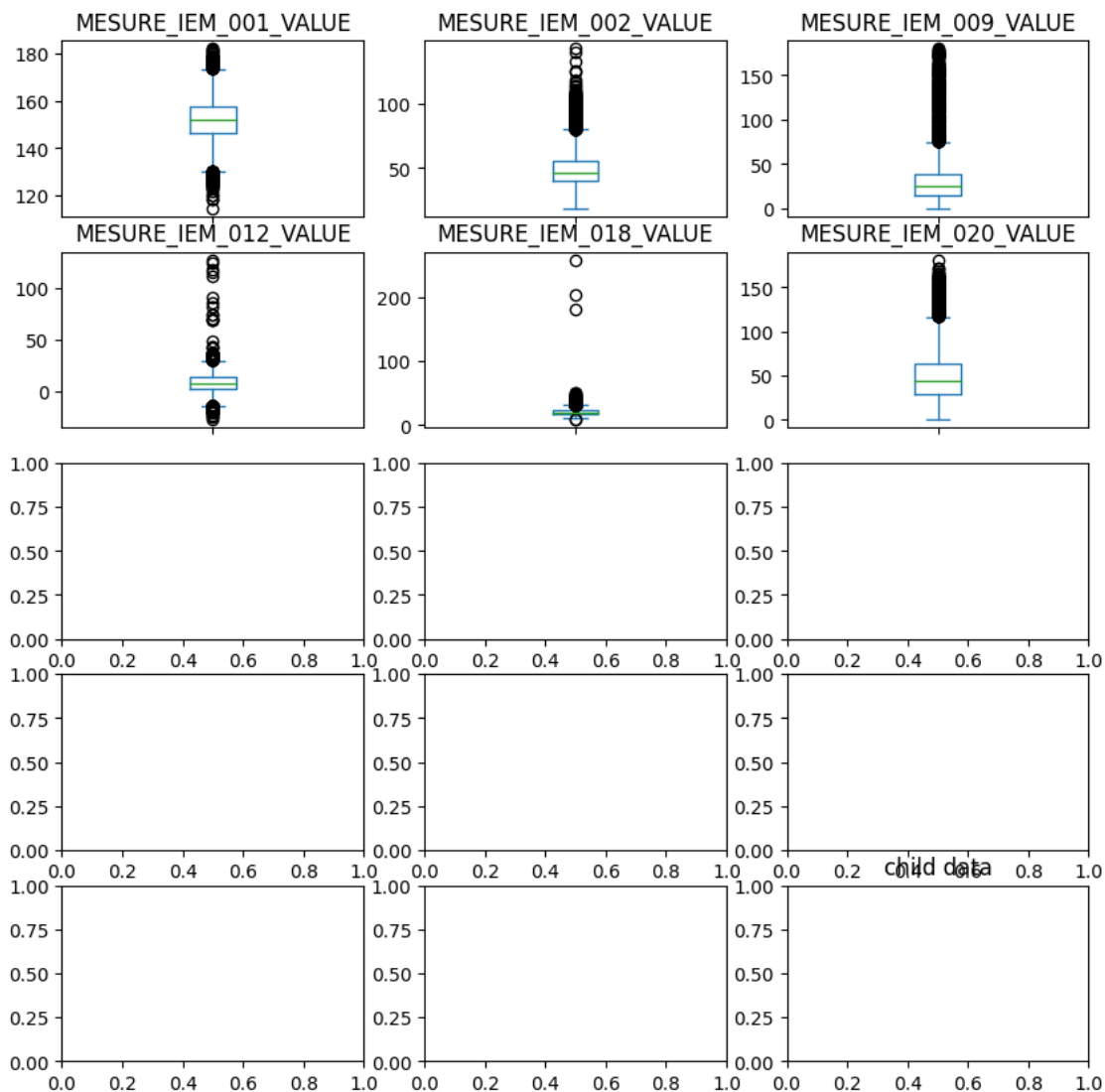
모든 열에 대해 박스플롯 그리기
for i, column in enumerate(data_child.columns[6:12]):
 ax = axes[i // 3, i % 3]
 data_child[column].plot(kind='box', ax=ax)
 ax.set_title(column)
 ax.set_xticklabels([]) # x축 레이블 제거

plt.suptitle('Child Data', fontsize=16)

레이아웃 조정
plt.title('child data')
plt.show()

```

Child Data



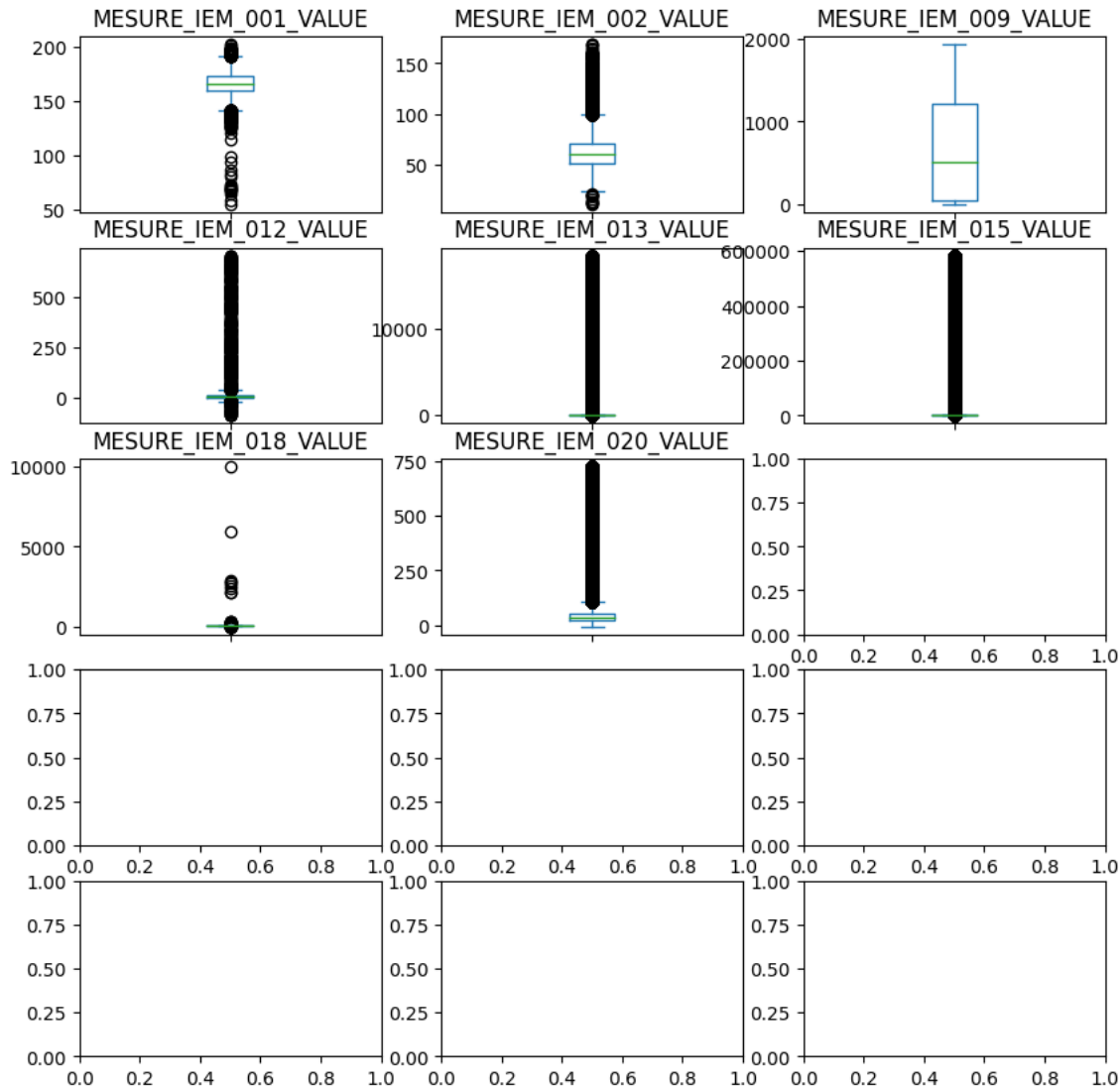


```
청소년(teen) 데이터 기준 박스플롯
그림 초기화
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(10, 10))

모든 열에 대해 박스플롯 그리기
for i, column in enumerate(data_teen.columns[6:14]):
 ax = axes[i // 3, i % 3]
 data_teen[column].plot(kind='box', ax=ax)
 ax.set_title(column)
 ax.set_xticklabels([]) # x축 레이블 제거

plt.suptitle('Teen Data', fontsize=16)
plt.show()
```

Teen Data

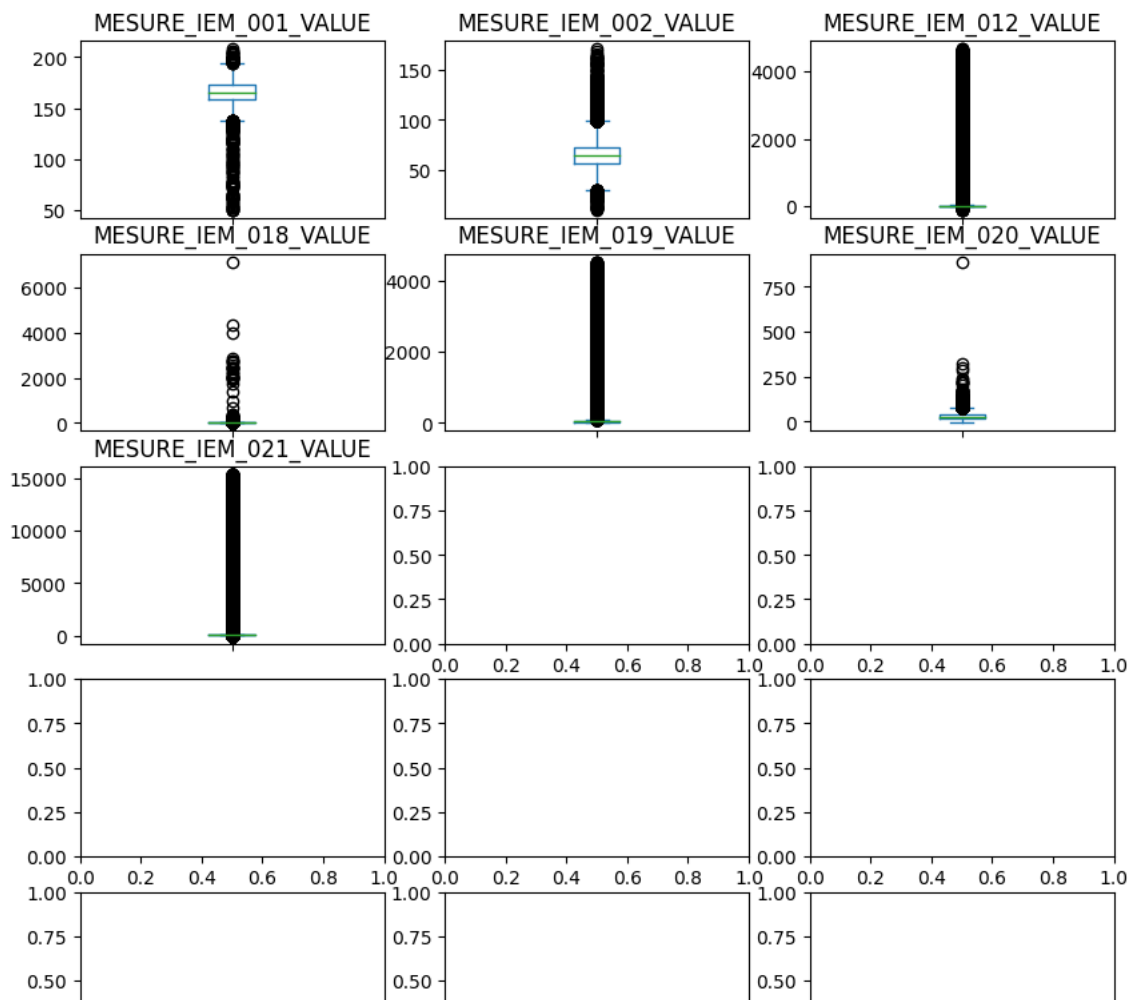


```
성인(adult) 데이터 기준 박스플롯
그림 초기화
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(10, 10))

모든 열에 대해 박스플롯 그리기
for i, column in enumerate(data_adult.columns[6:13]):
 ax = axes[i // 3, i % 3]
 data_adult[column].plot(kind='box', ax=ax)
 ax.set_title(column)
 ax.set_xticklabels([]) # x축 레이블 제거

plt.suptitle('Adult Data', fontsize=16)
plt.show()
```

## Adult Data

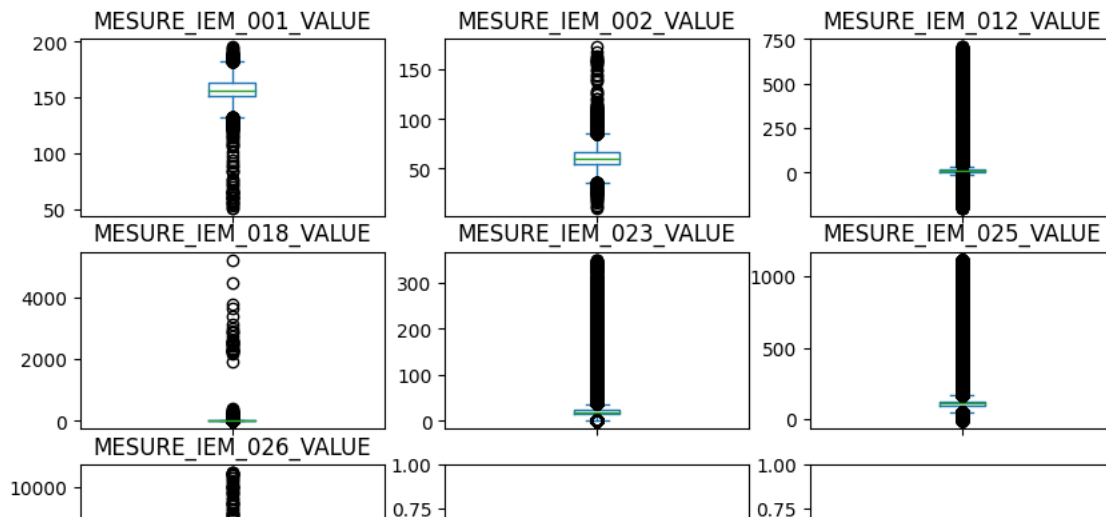


```
노인(elder) 데이터 기준 박스플롯
그림 초기화
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(10, 10))
```

```
모든 열에 대해 박스플롯 그리기
for i, column in enumerate(data_elder.columns[6:13]):
 ax = axes[i // 3, i % 3]
 data_elder[column].plot(kind='box', ax=ax)
 ax.set_title(column)
 ax.set_xticklabels([]) # x축 레이블 제거
```

```
plt.suptitle('Elder Data', fontsize=16)
plt.show()
```

## Elder Data



## ▼ IQR을 활용한 이상치 제거

```

iqr을 이용한 이상치 제거
IQR 기준 이상치 제거 함수
def remove_outliers(df, column):
 q1 = df.loc[:, column].quantile(0.25) # 1사분위수
 q3 = df.loc[:, column].quantile(0.75) # 3사분위수
 iqr = q3 - q1 # IQR 계산
 lower_bound = q1 - 1.5 * iqr # 하위 이상치 제거 기준
 upper_bound = q3 + 1.5 * iqr # 상위 이상치 제거 기준
 df_filtered = df[(df[column] >= lower_bound) & (df[column] <= upper_bound)] # 이상치 제거
 return df_filtered

data_child 상관도 분석하기
col_list = data_child.columns[6:-1].tolist()
col_list.append('CRTFC_FLAG_NM')
print(col_list)
data_child[col_list].corr()
20,9,12,18,2,22,1

['MESURE_IEM_001_VALUE', 'MESURE_IEM_002_VALUE', 'MESURE_IEM_009_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_018_VALUE', 'MESURE_IEM_020_VALUE',
MESURE_IEM_001_VALUE MESURE_IEM_002_VALUE MESURE_IEM_009_VALUE MESURE_IEM_012_VALUE MESURE_IEM_018_V
MESURE_IEM_001_VALUE 1.000000 0.648739 0.096333 -0.027223 0.27
MESURE_IEM_002_VALUE 0.648739 1.000000 -0.120610 -0.079085 0.84
MESURE_IEM_009_VALUE 0.096333 -0.120610 1.000000 0.122383 -0.18
MESURE_IEM_012_VALUE -0.027223 -0.079085 0.122383 1.000000 -0.07
MESURE_IEM_018_VALUE 0.278060 0.845093 -0.185921 -0.076387 1.00
MESURE_IEM_020_VALUE -0.045239 -0.309410 0.331523 0.085407 -0.33
MESURE_IEM_022_VALUE 0.118686 -0.059122 0.242941 0.093092 -0.12
CRTFC_FLAG_NM 0.048770 0.246726 -0.377167 -0.336875 0.26

data_teen 상관도 분석하기
col_list = data_teen.columns[6:-1].tolist()
col_list.append('CRTFC_FLAG_NM')
print(col_list)
data_teen[col_list].corr()
22,2,12,13,15,18,9,20,1

```

```
['MESURE_IEM_001_VALUE', 'MESURE_IEM_002_VALUE', 'MESURE_IEM_009_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_013_VALUE', 'MESURE_IEM_015_VALUE',
MESURE_IEM_001_VALUE MESURE_IEM_002_VALUE MESURE_IEM_009_VALUE MESURE_IEM_012_VALUE MESURE_IEM_013_V

MESURE_IEM_001_VALUE 1.000000 0.574218 -0.040056 -0.095575 0.05
MESURE_IEM_002_VALUE 0.574218 1.000000 -0.038216 -0.053555 0.04
MESURE_IEM_009_VALUE -0.040056 -0.038216 1.000000 0.006591 -0.02
MESURE_IEM_012_VALUE -0.095575 -0.053555 0.006591 1.000000 0.01
MESURE_IEM_013_VALUE 0.051005 0.041958 -0.025739 0.016471 1.00
MESURE_IEM_015_VALUE 0.020816 0.004460 0.013205 0.011304 0.41

data_adult 상관도 분석하기
col_list = data_adult.columns[6:-1].tolist()
col_list.append('CRTFC_FLAG_NM')
print(col_list)
data_adult[col_list].corr()
20, 2, 22,18,21,19,1,12

['MESURE_IEM_001_VALUE', 'MESURE_IEM_002_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_018_VALUE', 'MESURE_IEM_019_VALUE', 'MESURE_IEM_020_VALUE',
MESURE_IEM_001_VALUE MESURE_IEM_002_VALUE MESURE_IEM_012_VALUE MESURE_IEM_018_VALUE MESURE_IEM_019_V

MESURE_IEM_001_VALUE 1.000000 0.657343 -0.008080 -0.017849 0.07
MESURE_IEM_002_VALUE 0.657343 1.000000 -0.009378 0.123127 0.04
MESURE_IEM_012_VALUE -0.008080 -0.009378 1.000000 -0.000837 0.07
MESURE_IEM_018_VALUE -0.017849 0.123127 -0.000837 1.000000 0.00
MESURE_IEM_019_VALUE 0.072380 0.046620 0.073240 0.000291 1.00
MESURE_IEM_020_VALUE 0.526361 0.249610 -0.000398 -0.012126 0.08
MESURE_IEM_021_VALUE 0.060735 0.059142 0.010136 0.003682 0.05
MESURE_IEM_022_VALUE 0.318533 0.194424 0.021607 0.001804 0.13
CRTFC_FLAG_NM -0.022096 0.138708 -0.007595 0.033360 -0.02

data_elder 상관도 분석하기
col_list = data_elder.columns[6:-1].tolist()
col_list.append('CRTFC_FLAG_NM')
print(col_list)
data_elder[col_list].corr()
23,12,25,2,1,27,18,26

['MESURE_IEM_001_VALUE', 'MESURE_IEM_002_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_018_VALUE', 'MESURE_IEM_023_VALUE', 'MESURE_IEM_025_VALUE',
MESURE_IEM_001_VALUE MESURE_IEM_002_VALUE MESURE_IEM_012_VALUE MESURE_IEM_018_VALUE MESURE_IEM_023_V

MESURE_IEM_001_VALUE 1.000000 0.614083 -0.125143 -0.049567 0.09
MESURE_IEM_002_VALUE 0.614083 1.000000 -0.097385 0.073992 0.02
MESURE_IEM_012_VALUE -0.125143 -0.097385 1.000000 -0.003428 0.08
MESURE_IEM_018_VALUE -0.049567 0.073992 -0.003428 1.000000 -0.00
MESURE_IEM_023_VALUE 0.098953 0.029391 0.085225 -0.008344 1.00
MESURE_IEM_025_VALUE 0.030570 -0.004810 0.034742 -0.005517 0.13
MESURE_IEM_026_VALUE -0.004882 0.001789 0.037606 0.000453 0.08
MESURE_IEM_027_VALUE -0.029159 -0.012401 -0.007663 0.001506 -0.03
CRTFC_FLAG_NM -0.060032 0.109537 -0.170508 0.023275 -0.26

#data_child 이상치 제거
20,9,12,18,2,22,1
columns= ['MESURE_IEM_020_VALUE', 'MESURE_IEM_009_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_018_VALUE', 'MESURE_IEM_002_VALUE', 'MESURE_IEM_022_VALUE'
]

for i in columns:
 if i == "MESURE_IEM_020_VALUE":
 data_child_outlier = remove_outliers(data_child, i)
 else:
 data_child_outlier = remove_outliers(data_child_outlier,i)
print(data_child_outlier.shape)
```

(29277, 14)

```
data_teen 이상치 제거
22,2,12,13,15,18,9,20,1
columns= ['MESURE_IEM_022_VALUE', 'MESURE_IEM_002_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_013_VALUE', 'MESURE_IEM_015_VALUE', 'MESURE_IEM_018_VALUE',
 'MESURE_IEM_009_VALUE',

 'MESURE_IEM_020_VALUE', 'MESURE_IEM_001_VALUE'

]

for i in columns:
 if i == "MESURE_IEM_022_VALUE":
 data_teen_outlier = remove_outliers(data_teen, i)
 else:
 data_teen_outlier = remove_outliers(data_teen_outlier,i)
print(data_teen_outlier.shape)

(181169, 16)
```

#### ▼ data\_teen에 대해 해당 연령값에서 벗어난 데이터 삭제

```
data_teen=data_teen[data_teen['MESURE_AGE_C0']>=13]
data_teen=data_teen[data_teen['MESURE_AGE_C0']<=18]
data_teen['MESURE_AGE_C0'].describe()
```

```
count 235781.000000
mean 15.402259
std 1.586854
min 13.000000
25% 14.000000
50% 16.000000
75% 17.000000
max 18.000000
Name: MESURE_AGE_C0, dtype: float64
```

```
data_teen['MESURE_AGE_C0'].value_counts()
```

```
MESURE_AGE_C0
16.0 50231
17.0 49515
13.0 41780
15.0 41531
14.0 32257
18.0 20467
Name: count, dtype: int64
```

```
data_adult 이상치 제거
20, 2, 22,18,21,19,1,12
columns= ['MESURE_IEM_020_VALUE', 'MESURE_IEM_002_VALUE', 'MESURE_IEM_022_VALUE', 'MESURE_IEM_018_VALUE', 'MESURE_IEM_021_VALUE', 'MESURE_IEM_019_VALUE',
 'MESURE_IEM_012_VALUE'

]

for i in columns:
 if i == "MESURE_IEM_020_VALUE":
 data_adult_outlier = remove_outliers(data_adult, i)
 else:
 data_adult_outlier = remove_outliers(data_adult_outlier,i)
print(data_adult_outlier.shape)

(377609, 15)
```

#### ▼ data\_adult에 대해 음수가 나올 수 없는 데이터의 음수 측정값 삭제

```
data_adult=data_adult[data_adult['MESURE_IEM_020_VALUE']>0]
data_adult=data_adult[data_adult['MESURE_IEM_021_VALUE']>0]
print(data_adult['MESURE_IEM_020_VALUE'].describe())
print(data_adult['MESURE_IEM_021_VALUE'].describe())
```

```
count 419850.000000
mean 29.202686
std 19.333800
min 0.100000
25% 15.000000
50% 24.000000
75% 40.000000
```

```

max 890.000000
Name: MESURE_IEM_020_VALUE, dtype: float64
count 419850.000000
mean 211.724217
std 1413.842521
min 0.090000
25% 11.160000
50% 12.850000
75% 14.530000
max 15399.000000
Name: MESURE_IEM_021_VALUE, dtype: float64

```

```

data_adult에 대해 나이 이상치 제거
data_adult=data_adult[data_adult['MESURE_AGE_C0']>=19]
data_adult=data_adult[data_adult['MESURE_AGE_C0']<=65]
print(data_adult['MESURE_AGE_C0'].describe())

```

```

count 419797.000000
mean 37.124122
std 15.082095
min 19.000000
25% 22.000000
50% 35.000000
75% 51.000000
max 65.000000
Name: MESURE_AGE_C0, dtype: float64

```

```

data_elder 이상치 제거
23,12,25,2,1,27,18,26
columns= ['MESURE_IEM_023_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_025_VALUE', 'MESURE_IEM_002_VALUE', 'MESURE_IEM_001_VALUE',
 'MESURE_IEM_027_VALUE', 'MESURE_IEM_018_VALUE', 'MESURE_IEM_026_VALUE'

```

```

]

for i in columns:
 if i == "MESURE_IEM_023_VALUE":
 data_elder_outlier = remove_outliers(data_elder, i)
 else:
 data_elder_outlier = remove_outliers(data_elder_outlier,i)
print(data_elder_outlier.shape)

(193444, 15)

```

#### ▼ data\_elder에 대해 음수가 나올 수 없는 데이터의 음수 측정값 삭제

```

data_elder = data_elder[data_elder['MESURE_IEM_026_VALUE']>0].describe()
data_elder['MESURE_IEM_026_VALUE'].describe()

```

```

count 8.000000
mean 31038.334194
std 83349.513020
min 0.100000
25% 5.900000
50% 8.537471
75% 2873.073960
max 237098.000000
Name: MESURE_IEM_026_VALUE, dtype: float64

```

```

data_elder=data_elder[data_elder['MESURE_AGE_C0']>=65]
data_elder.drop(data_elder[data_elder['MESURE_AGE_C0'] == 173].index,inplace=True)
data_elder.drop(data_elder[data_elder['MESURE_AGE_C0'] == 118].index,inplace=True)

```

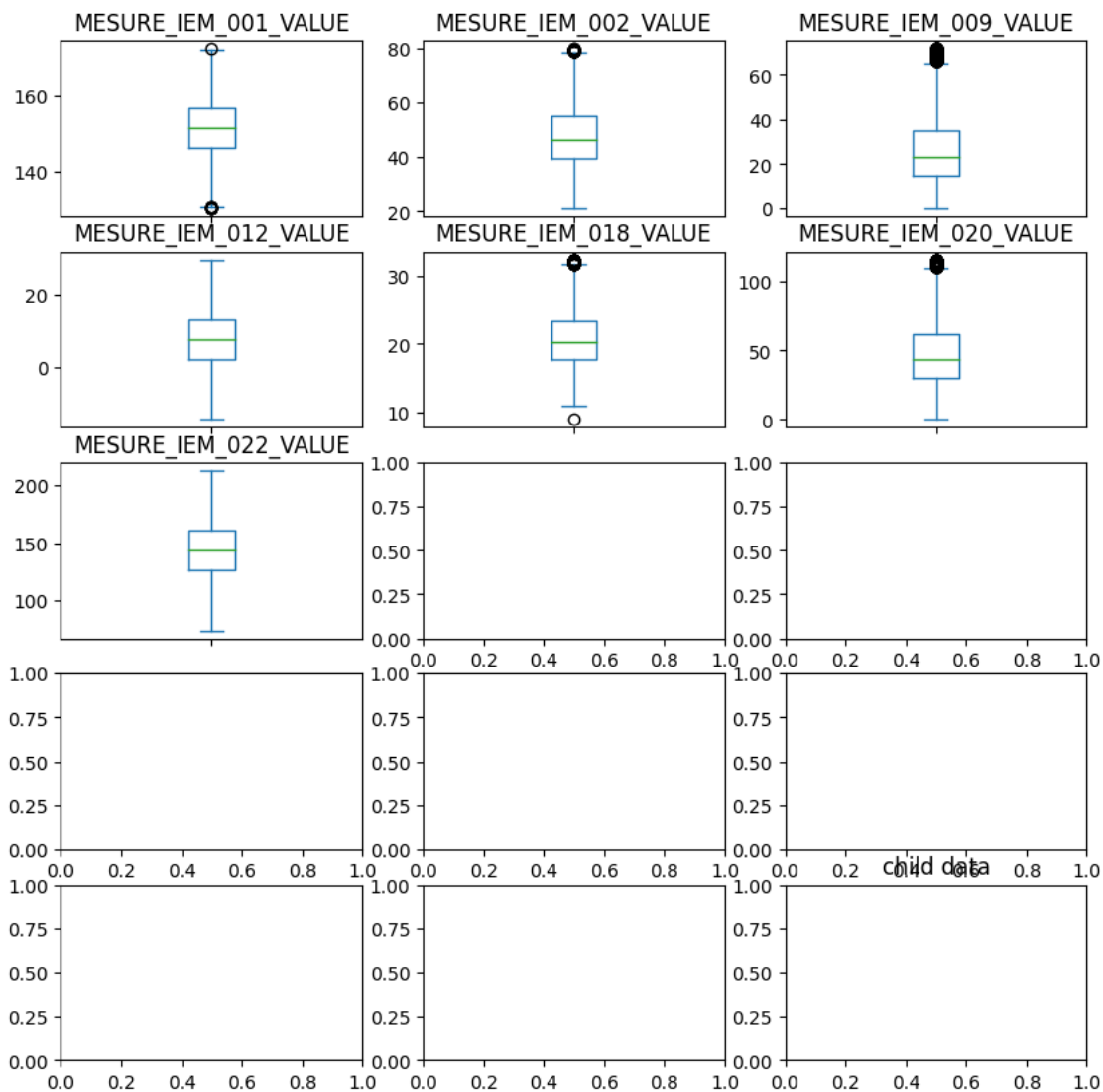
```
유소년 데이터 기준 박스플롯
그림 초기화
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(10, 10))

모든 열에 대해 박스플롯 그리기
for i, column in enumerate(data_child_outlier.columns[6:-1]):
 ax = axes[i // 3, i % 3]
 data_child_outlier[column].plot(kind='box', ax=ax)
 ax.set_title(column)
 ax.set_xticklabels([]) # x축 레이블 제거

plt.suptitle('Child Data', fontsize=16)

레이아웃 조정
plt.title('child data')
plt.show()
```

### Child Data

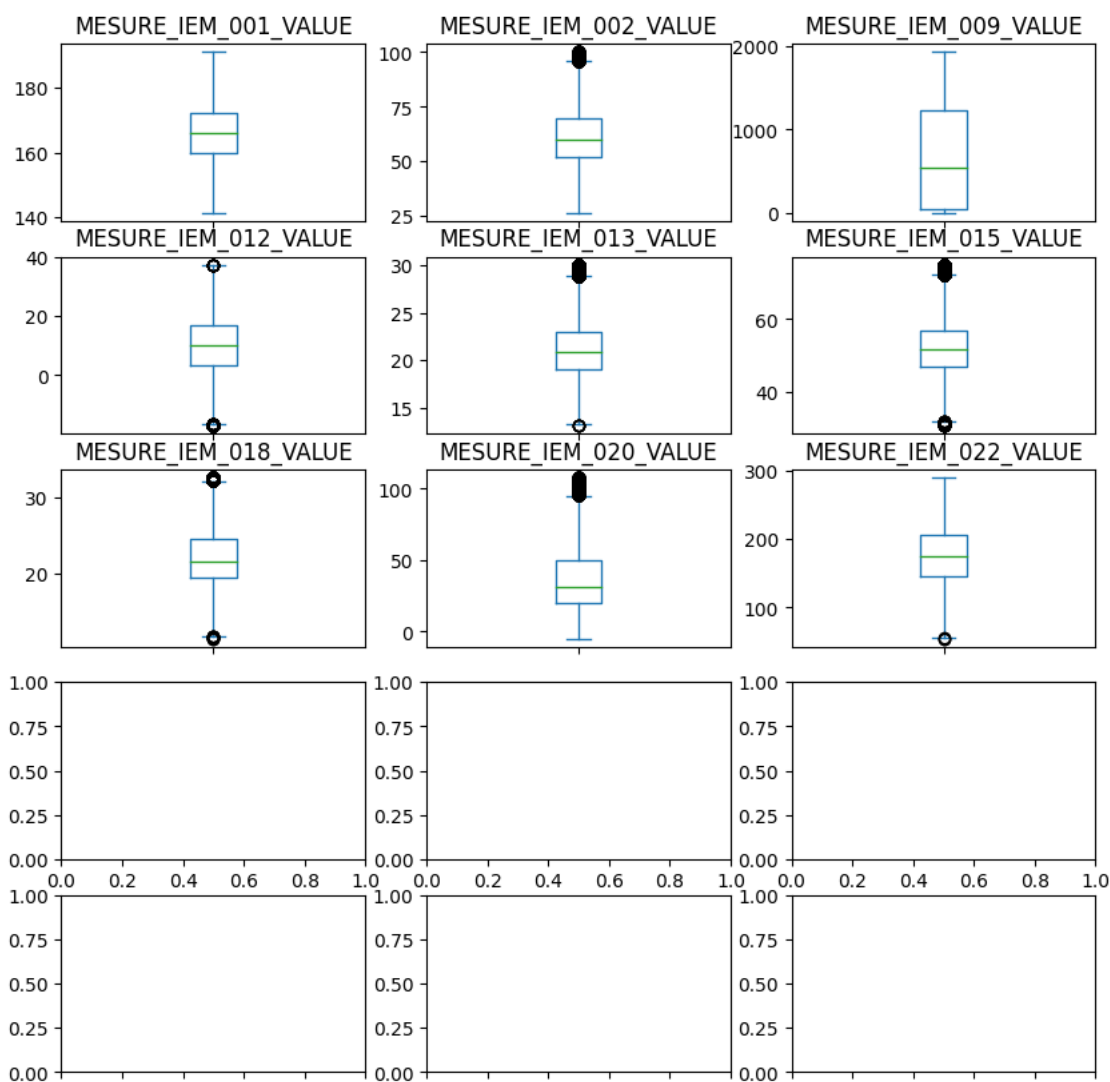


```
청소년(teen) 데이터 기준 박스플롯
그림 초기화
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(10, 10))

모든 열에 대해 박스플롯 그리기
for i, column in enumerate(data_teen_outlier.columns[6:-1]):
 ax = axes[i // 3, i % 3]
 data_teen_outlier[column].plot(kind='box', ax=ax)
 ax.set_title(column)
 ax.set_xticklabels([]) # x축 레이블 제거

plt.suptitle('Teen Data', fontsize=16)
plt.show()
```

## Teen Data



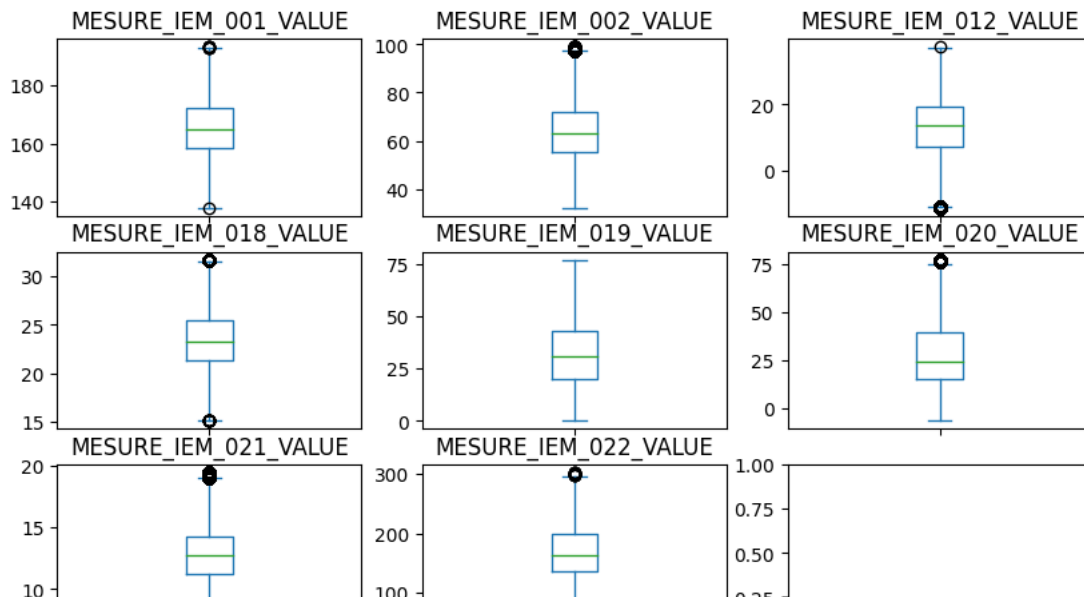
```
성인(adult) 데이터 기준 박스플롯
그림 초기화
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(10, 10))

모든 열에 대해 박스플롯 그리기
for i, column in enumerate(data_adult_outlier.columns[6:-1]):
 ax = axes[i // 3, i % 3]
 data_adult_outlier[column].plot(kind='box', ax=ax)
 ax.set_title(column)
 ax.set_xticklabels([]) # x축 레이블 제거

plt.suptitle('Adult Data', fontsize=16)
plt.show()
```



## Adult Data

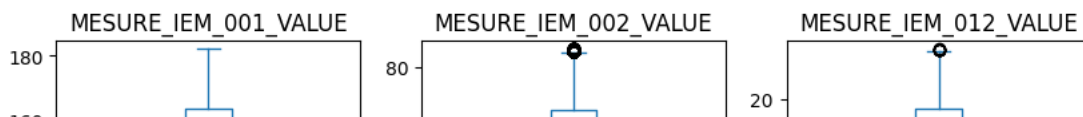


```
노인(elder) 데이터 기준 박스플롯
그림 초기화
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(10, 10))

모든 열에 대해 박스플롯 그리기
for i, column in enumerate(data_elder_outlier.columns[6:-1]):
 ax = axes[i // 3, i % 3]
 data_elder_outlier[column].plot(kind='box', ax=ax)
 ax.set_title(column)
 ax.set_xticklabels([]) # x축 레이블 제거

plt.suptitle('Elder Data', fontsize=16)
plt.show()
```

## Elder Data



```
import pandas as pd
pathname = r"C:\Users\Wyoju\Documents\Wmy_ws\project\data_project\data_child2.xlsx"
data_child_criteria = pd.read_excel(pathname)

등급표 인코딩하기
data_child_criteria['등급'] = data_child_criteria['등급'].map({'3등급':3, '2등급':2, '1등급':1})
print(data_child_criteria.head(10))
```

|   | 등급 | 나이 | 성별 | MESURE_IEM_020_VALUE | MESURE_IEM_022_VALUE | W |
|---|----|----|----|----------------------|----------------------|---|
| 0 | 1  | 11 | 남  | 77                   | 161.0                |   |
| 1 | 1  | 12 | 남  | 91                   | 160.3                |   |
| 2 | 2  | 11 | 남  | 59                   | 147.0                |   |
| 3 | 2  | 12 | 남  | 70                   | 146.0                |   |
| 4 | 3  | 11 | 남  | 46                   | 133.0                |   |
| 5 | 3  | 12 | 남  | 55                   | 133.0                |   |
| 6 | 1  | 11 | 여  | 62                   | 165.0                |   |
| 7 | 1  | 12 | 여  | 62                   | 164.0                |   |
| 8 | 2  | 11 | 여  | 51                   | 146.0                |   |
| 9 | 2  | 12 | 여  | 51                   | 145.0                |   |

|   | MESURE_IEM_022_VALUE.1 | MESURE_IEM_012_VALUE | MESURE_IEM_009_VALUE |
|---|------------------------|----------------------|----------------------|
| 0 | 161.0                  | 11.5                 | 36                   |
| 1 | 160.3                  | 12.0                 | 35                   |
| 2 | 147.0                  | 7.8                  | 26                   |
| 3 | 146.0                  | 7.9                  | 27                   |
| 4 | 133.0                  | 4.0                  | 19                   |
| 5 | 133.0                  | 4.0                  | 19                   |
| 6 | 165.0                  | 10.9                 | 36                   |
| 7 | 164.0                  | 11.0                 | 35                   |
| 8 | 146.0                  | 6.5                  | 26                   |
| 9 | 145.0                  | 6.5                  | 27                   |

```
import pandas as pd
pathname = r"C:\Users\Wyoju\Documents\Wmy_ws\project\data_project\data_teen2.xlsx"
data_teen_criteria = pd.read_excel(pathname)

등급표 인코딩하기
data_teen_criteria['등급'] = data_teen_criteria['등급'].map({'3등급':3, '2등급':2, '1등급':1})
print(data_teen_criteria.head(10))
```

|   | 등급 | 나이 | 성별 | MESURE_IEM_020_VALUE | MESURE_IEM_022_VALUE | W |
|---|----|----|----|----------------------|----------------------|---|
| 0 | 1  | 13 | 남  | 50                   | 185                  |   |
| 1 | 1  | 14 | 남  | 56                   | 192                  |   |
| 2 | 1  | 15 | 남  | 56                   | 199                  |   |
| 3 | 1  | 16 | 남  | 57                   | 206                  |   |
| 4 | 1  | 17 | 남  | 59                   | 213                  |   |
| 5 | 1  | 18 | 남  | 61                   | 219                  |   |
| 6 | 2  | 13 | 남  | 40                   | 168                  |   |
| 7 | 2  | 14 | 남  | 46                   | 175                  |   |
| 8 | 2  | 15 | 남  | 46                   | 182                  |   |
| 9 | 2  | 16 | 남  | 47                   | 189                  |   |

|   | MESURE_IEM_012_VALUE | MESURE_IEM_009_VALUE | MESURE_IEM_013_VALUE | W |
|---|----------------------|----------------------|----------------------|---|
| 0 | 9.6                  | 61                   | 19.1                 |   |
| 1 | 10.9                 | 60                   | 18.7                 |   |
| 2 | 11.9                 | 49                   | 19.0                 |   |
| 3 | 13.8                 | 49                   | 18.7                 |   |
| 4 | 14.0                 | 46                   | 18.3                 |   |
| 5 | 14.8                 | 53                   | 18.0                 |   |
| 6 | 5.3                  | 49                   | 20.3                 |   |
| 7 | 6.5                  | 48                   | 19.9                 |   |
| 8 | 6.9                  | 39                   | 20.3                 |   |
| 9 | 8.9                  | 38                   | 20.1                 |   |

|   | MESURE_IEM_015_VALUE |
|---|----------------------|
| 0 | 53.9                 |
| 1 | 51.0                 |
| 2 | 50.2                 |
| 3 | 47.8                 |
| 4 | 47.3                 |
| 5 | 47.0                 |
| 6 | 58.2                 |
| 7 | 55.1                 |
| 8 | 53.9                 |
| 9 | 51.7                 |

```
import pandas as pd
pathname = r"C:\Users\Wysoju\Documents\my_ws\project\data_project\data_adult.xlsx"
data_adult_criteria = pd.read_excel(pathname)
등급표 인코딩하기
data_adult_criteria['등급'] = data_adult_criteria['등급'].map({'3등급':3, '2등급':2, '1등급':1})
print(data_adult_criteria.tail(10))
```

|    | 등급 | 나이    | 성별 | MESURE_IEM_020_VALUE | MESURE_IEM_022_VALUE | W |
|----|----|-------|----|----------------------|----------------------|---|
| 44 | 3  | 40~44 | 남  | 27                   | 191                  |   |
| 45 | 3  | 40~44 | 여  | 15                   | 129                  |   |
| 46 | 3  | 45~49 | 남  | 24                   | 186                  |   |
| 47 | 3  | 45~49 | 여  | 14                   | 131                  |   |
| 48 | 3  | 50~54 | 남  | 21                   | 182                  |   |
| 49 | 3  | 50~54 | 여  | 12                   | 129                  |   |
| 50 | 3  | 55~59 | 남  | 18                   | 175                  |   |
| 51 | 3  | 55~59 | 여  | 11                   | 123                  |   |
| 52 | 3  | 60~64 | 남  | 15                   | 164                  |   |
| 53 | 3  | 60~64 | 여  | 9                    | 113                  |   |

|    | MESURE_IEM_019_VALUE | MESURE_IEM_021_VALUE | MESURE_IEM_022_VALUE.1 | W   |
|----|----------------------|----------------------|------------------------|-----|
| 44 | 32                   | 12.6                 |                        | 191 |
| 45 | 19                   | 14.9                 |                        | 129 |
| 46 | 30                   | 12.9                 |                        | 186 |
| 47 | 16                   | 14.9                 |                        | 131 |
| 48 | 26                   | 13.1                 |                        | 182 |
| 49 | 13                   | 15.0                 |                        | 129 |
| 50 | 23                   | 13.5                 |                        | 175 |
| 51 | 9                    | 15.4                 |                        | 123 |
| 52 | 19                   | 14.0                 |                        | 164 |
| 53 | 7                    | 15.9                 |                        | 113 |

|    | MESURE_IEM_012_VALUE |
|----|----------------------|
| 44 | 4.8                  |
| 45 | 10.4                 |
| 46 | 4.6                  |
| 47 | 10.7                 |
| 48 | 4.7                  |
| 49 | 11.7                 |
| 50 | 3.9                  |
| 51 | 11.9                 |
| 52 | 2.3                  |
| 53 | 11.8                 |

```
import pandas as pd
pathname = r"C:\Users\Wysoju\Documents\my_ws\project\data_project\data_elder2.xlsx"
data_elder_criteria = pd.read_excel(pathname)
등급표 인코딩하기
data_elder_criteria['등급'] = data_elder_criteria['등급'].map({'3등급':3, '2등급':2, '1등급':1})
print(data_elder_criteria.head(10))
```

|   | 등급 | 나이    | 성별 | MESURE_IEM_025_VALUE | MESURE_IEM_027_VALUE | W |
|---|----|-------|----|----------------------|----------------------|---|
| 0 | 1  | 65~69 | 남  | 122                  | 21.0                 |   |
| 1 | 1  | 70~74 | 남  | 116                  | 22.5                 |   |
| 2 | 1  | 75~79 | 남  | 112                  | 24.0                 |   |
| 3 | 1  | 80~84 | 남  | 105                  | 26.3                 |   |
| 4 | 1  | 85~   | 남  | 94                   | 28.5                 |   |
| 5 | 2  | 65~69 | 남  | 109                  | 23.9                 |   |
| 6 | 2  | 70~74 | 남  | 103                  | 25.6                 |   |
| 7 | 2  | 75~79 | 남  | 97                   | 27.8                 |   |
| 8 | 2  | 80~84 | 남  | 89                   | 30.8                 |   |
| 9 | 2  | 85~   | 남  | 77                   | 33.6                 |   |

|   | MESURE_IEM_023_VALUE | MESURE_IEM_027_VALUE.1 | MESURE_IEM_012_VALUE | W    |
|---|----------------------|------------------------|----------------------|------|
| 0 | 25                   | 21.0                   |                      | 11.4 |
| 1 | 22                   | 22.5                   |                      | 9.2  |
| 2 | 20                   | 24.0                   |                      | 7.5  |
| 3 | 18                   | 26.3                   |                      | 5.3  |
| 4 | 16                   | 28.5                   |                      | 3.7  |
| 5 | 21                   | 23.9                   |                      | 6.6  |
| 6 | 19                   | 25.6                   |                      | 4.4  |
| 7 | 17                   | 27.8                   |                      | 2.7  |
| 8 | 15                   | 30.8                   |                      | 0.4  |
| 9 | 13                   | 33.6                   |                      | -1.2 |

|   | MESURE_IEM_026_VALUE |
|---|----------------------|
| 0 | 4.9                  |
| 1 | 5.1                  |
| 2 | 5.4                  |
| 3 | 5.7                  |
| 4 | 6.0                  |
| 5 | 5.4                  |
| 6 | 5.6                  |
| 7 | 5.9                  |
| 8 | 6.2                  |
| 9 | 6.5                  |

```
import pandas as pd
pathname = r"C:\Users\Wysoju\Documents\my_ws\project\data_project\recommended_exercise_list.xlsx"
recommended_exercise = pd.read_excel(pathname)
recommended_exercise
```

|     | 나이    | 체력항목  | 추천 운동 동영상                                                               |
|-----|-------|-------|-------------------------------------------------------------------------|
| 0   | 11~12 | 근력    | <a href="https://youtu.be/KLp2eCPRc6Q">https://youtu.be/KLp2eCPRc6Q</a> |
| 1   | 11~12 | 근력    | <a href="https://youtu.be/LXvXhGDda-c">https://youtu.be/LXvXhGDda-c</a> |
| 2   | 11~12 | 근력    | <a href="https://youtu.be/3ofToKYGZX4">https://youtu.be/3ofToKYGZX4</a> |
| 3   | 11~12 | 심폐지구력 | <a href="https://youtu.be/7Rqc2MhfqKU">https://youtu.be/7Rqc2MhfqKU</a> |
| 4   | 11~12 | 심폐지구력 | <a href="https://youtu.be/fo1FIS41oig">https://youtu.be/fo1FIS41oig</a> |
| ... | ...   | ...   | ...                                                                     |
| 67  | 65~   | 순발력   | <a href="https://youtu.be/1fA4qBGyqhM">https://youtu.be/1fA4qBGyqhM</a> |
| 68  | 65~   | 순발력   | <a href="https://youtu.be/N7lBjoT4oGo">https://youtu.be/N7lBjoT4oGo</a> |
| 69  | 65~   | 평형성   | <a href="https://youtu.be/Csa1WTJc5ys">https://youtu.be/Csa1WTJc5ys</a> |
| 70  | 65~   | 평형성   | <a href="https://youtu.be/1CTWSroJe3g">https://youtu.be/1CTWSroJe3g</a> |
| 71  | 65~   | 평형성   | <a href="https://youtu.be/1CTWSroJe3g">https://youtu.be/1CTWSroJe3g</a> |

72 rows × 3 columns

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
기입했을 때 레이더 차트 그려지도록

값이 높을수록 좋은 등급인 측정 항목 (~ 이상): 9, 12, 19, 20, 22, 23, 25
값이 낮을수록 좋은 등급인 측정 항목 (~ 이하): 13, 15, 21, 26, 27
positive_rating = ('MESURE_IEM_009_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_019_VALUE', 'MESURE_IEM_020_VALUE', 'MESURE_IEM_022_VALUE', 'MESURE_IEM_022',
negative_rating = ('MESURE_IEM_013_VALUE', 'MESURE_IEM_015_VALUE', 'MESURE_IEM_021_VALUE', 'MESURE_IEM_026_VALUE', 'MESURE_IEM_027_VALUE', 'MESURE_IEM_027

def child_health_estimate(age, gender, value1, value2, value3, value4, *value5):
 # 기종표 성별 범위, 나이 범위 좁히기
 data_list = []
 if age<=12:
 data_child_gender = data_child_criteria[data_child_criteria['성별']==gender]
 data_age = data_child_gender[data_child_gender['나이']==age]
 print(data_age.head())
 columns = data_age.columns.tolist()
 if value5:
 data_list = [value1, value2, value3, value4]
 for value in value5:
 data_list.append(value)
 else:
 data_list = [value1, value2, value3, value4]

 #행 하나씩 내려가게
 grade_li = []

 for i in range(0,len(data_list)): #열에 적용
 grade_cnt = 0
 grade=1
 for index, row in data_age.iterrows(): # 행에 적용
 if columns[i+3] in positive_rating:
 if row[columns[i+3]] <= data_list[i]:
 if grade <= 5-int(row['등급']):
 grade = 5-int(row['등급'])
 grade_li.append(grade)
 break
 else:
 grade_cnt = grade_cnt + 1
 if grade_cnt==3:
 grade = 1
 grade_li.append(grade)

 elif columns[i+3] in negative_rating:
 if row[columns[i+3]] >= data_list[i]:
 if grade <= 5-int(row['등급']):
 grade=5-int(row['등급'])
 grade_li.append(grade)
 break
 else:
 grade_cnt = grade_cnt + 1
 if grade_cnt ==3:
 grade = 1
 grade_li.append(grade)
 making_radar_chart(grade_li)

def making_radar_chart(grade_li):
 # 데이터 준비
 values=[]
 num_vars = 0
 categories = []
 if len(grade_li) ==4:
 categories = ['심폐지구력', '순발력', '평형성', '유연성']
 num_vars = len(categories)
 for i in range(len(grade_li)):
 values.append(grade_li[i])

 elif len(grade_li) ==5:
 categories = ['심폐지구력', '순발력', '평형성', '유연성', '근지구력']
 num_vars = len(categories)
 for i in range(len(grade_li)):
 values.append(grade_li[i])
 # 각 카테고리의 각도 계산
 angles = np.linspace(0, 2 * np.pi, num_vars, endpoint=False).tolist()
 # 첫 번째 요소를 뒤로 복사하여 폐곡선을 만듭니다.
 values += values[:1]
 angles += angles[:1]

 # 레이더 차트 그리기
 plt.rc('font', family='Malgun Gothic')

```

```
fig, ax = plt.subplots(figsize=(6, 4), subplot_kw=dict(polar=True))
ax.fill(angles, values, color='green', alpha=0.25)
```

```
각 카테고리에 레이블 추가
ax.set_xticks(angles[:-1])
ax.set_xticklabels(categories)
```

```
y축 눈금을 1 단위로 설정
ax.set_yticks(np.arange(1, 5, 1))
ax.set_yticklabels(['노력필요', '낮음', '보통', ' 좋음'])
```

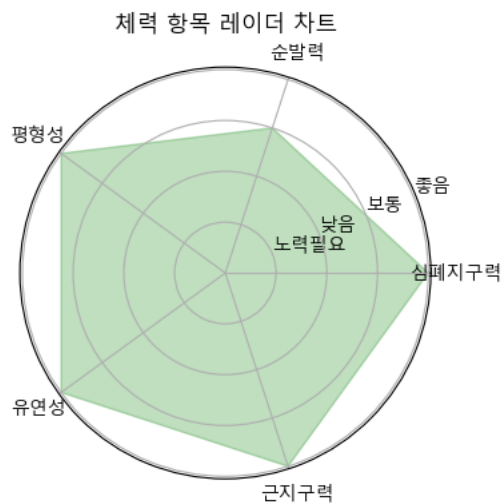
```
차트 제목
plt.rc('font', family='Malgun Gothic')
plt.title('체력 항목 레이더 차트', pad=20)
```

```
차트 표시
plt.show()
```

```
child_health_estimate(11, '여', 62, 164, 165, 10.9, 36)
```

| 등급 | 나이 | 성별 | MESURE_IEM_020_VALUE | MESURE_IEM_022_VALUE | W     |
|----|----|----|----------------------|----------------------|-------|
| 6  | 1  | 11 | 여                    | 62                   | 165.0 |
| 8  | 2  | 11 | 여                    | 51                   | 146.0 |
| 10 | 3  | 11 | 여                    | 40                   | 125.0 |

|    | MESURE_IEM_022_VALUE.1 | MESURE_IEM_012_VALUE | MESURE_IEM_009_VALUE |
|----|------------------------|----------------------|----------------------|
| 6  | 165.0                  | 10.9                 | 36                   |
| 8  | 146.0                  | 6.5                  | 26                   |
| 10 | 125.0                  | 3.0                  | 18                   |



```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
기입했을 때 레이더 차트 그려지도록

값이 높을수록 좋은 등급인 측정 항목 (~ 이상): 9, 12, 19, 20, 22, 23, 25
값이 낮을수록 좋은 등급인 측정 항목 (~ 이하): 13, 15, 21, 26, 27
positive_rating = ('MESURE_IEM_009_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_019_VALUE', 'MESURE_IEM_020_VALUE', 'MESURE_IEM_022_VALUE', 'MESURE_IEM_022')
negative_rating = ('MESURE_IEM_013_VALUE', 'MESURE_IEM_015_VALUE', 'MESURE_IEM_021_VALUE', 'MESURE_IEM_026_VALUE', 'MESURE_IEM_027_VALUE', 'MESURE_IEM_027')
def teen_health_estimate(age, gender, value1, value2, value3, *value4):
 # 기술평 성별 범위, 나이 범위 좁히기
 data_list=[]
 if age>=13:
 data_teen_gender = data_teen_criteria[data_teen_criteria['성별']==gender]
 data_age = data_teen_gender[data_teen_gender['나이']==age]
 columns = data_age.columns.tolist()
 if value4:
 data_list = [value1, value2, value3]
 for arg in value4:
 data_list.append(arg)

 else:
 data_list = [value1, value2, value3]

#행 하나씩 내려가게
grade_li = []

for i in range(0, len(data_list)):
 grade_cnt = 0
 grade=1
 for index, row in data_age.iterrows():

 if columns[i+3] in positive_rating:
 if row[columns[i+3]] <= data_list[i]:
 if grade <= 5-int(row['등급']):
 grade = 5-int(row['등급'])
 #print(grade)
 grade_li.append(grade)
 break
 else:
 grade_cnt = grade_cnt + 1
 if grade_cnt==3:
 grade = 1
 grade_li.append(grade)

 elif columns[i+3] in negative_rating:
 if row[columns[i+3]] >= data_list[i]:
 if grade <= 5-int(row['등급']):
 grade=5-int(row['등급'])
 #print(grade)
 grade_li.append(grade)
 break
 else:
 grade_cnt = grade_cnt + 1
 if grade_cnt ==3:
 grade = 1
 grade_li.append(grade)

 making_radar_chart(grade_li)

def making_radar_chart(grade_li):
 # 데이터 준비
 values=[]
 num_vars = 0
 categories = []
 if len(grade_li) ==3:
 categories = ['심폐지구력', '순발력', '유연성']
 num_vars = len(categories)
 for i in range(len(grade_li)):
 values.append(grade_li[i])

 elif len(grade_li) ==6:
 categories = ['심폐지구력', '순발력', '유연성', '근지구력', '민첩성', '평형성',]
 num_vars = len(categories)
 for i in range(len(grade_li)):
 values.append(grade_li[i])
 # 각 카테고리의 각도 계산
 angles = np.linspace(0, 2 * np.pi, num_vars, endpoint=False).tolist()
 # 첫 번째 요소를 뒤로 복사하여 폐곡선을 만듭니다.
 values += values[:1]
 angles += angles[:1]

```

```

레이더 차트 그리기
plt.rc('font', family='Malgun Gothic')
fig, ax = plt.subplots(figsize=(6, 4), subplot_kw=dict(polar=True))
ax.fill(angles, values, color='green', alpha=0.25)

각 카테고리에 레이블 추가
ax.set_xticks(angles[:-1])
ax.set_xticklabels(categories)

y축 눈금을 1 단위로 설정
ax.set_yticks(np.arange(1, 5, 1))
ax.set_yticklabels(['노력필요', '낮음', '보통', ' 좋음'])

차트 제목
plt.rc('font', family='Malgun Gothic')
plt.title('체력 항목 레이더 차트', pad=20)

차트 표시
plt.show()

teen_health_estimate(16, '남', 47, 189, 8.9)

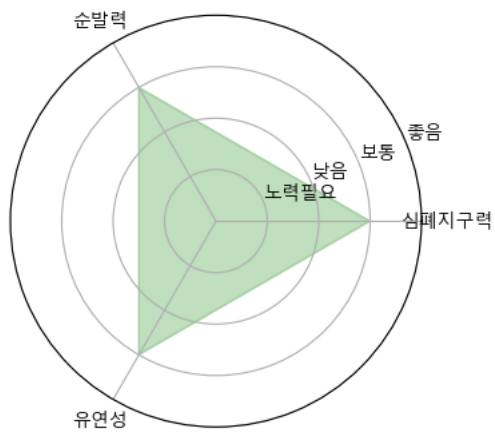
```

```

[47, 189, 8.9]
[3, 3, 3]

```

체력 항목 레이더 차트





```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
기입했을 때 레이더 차트 그려지도록

값이 높을수록 좋은 등급인 측정 항목 (~ 이상): 9, 12, 19, 20, 22, 23, 25
값이 낮을수록 좋은 등급인 측정 항목 (~ 이하): 13, 15, 21, 26, 27
positive_rating = ('MESURE_IEM_009_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_019_VALUE', 'MESURE_IEM_020_VALUE', 'MESURE_IEM_022_VALUE', 'MESURE_IEM_023_VALUE', 'MESURE_IEM_025_VALUE')
negative_rating = ('MESURE_IEM_013_VALUE', 'MESURE_IEM_015_VALUE', 'MESURE_IEM_021_VALUE', 'MESURE_IEM_026_VALUE', 'MESURE_IEM_027_VALUE', 'MESURE_IEM_028_VALUE')

def adult_health_estimate(age, gender, value1, value2, value3, value4, value5, value6):
 data_list = []
 # 기준표 성별, 나이 범위 좁히기
 if age <= 64 and age >= 19:
 data_adult_gender = data_adult_criteria[data_adult_criteria['성별'] == gender]
 ranges = [(i, i + 4) for i in range(25, 65, 5)]
 ranges.append((19, 24))
 for start, end in ranges:
 if start <= age <= end:
 data_age = data_adult_gender[data_adult_gender['나이'] == f'{start}~{end}']

 columns = data_age.columns.tolist()

 data_list = [value1, value2, value3, value4, value5, value6]

 # 행 하나씩 내려가게
 grade_li = []
 for i in range(0, len(data_list)):
 grade_cnt = 0
 grade = 1
 for index, row in data_age.iterrows():
 # print(data_age.columns[i])
 # print(row)
 if columns[i+3] in positive_rating:
 if row[data_age.columns.tolist()[i+3]] <= data_list[i]:
 # print(row[data_age.columns.tolist()[i]], "입력값>=", data_list[i-3])
 if grade <= 5 - int(row['등급']):
 grade = 5 - int(row['등급'])
 # print(grade, grade_cnt)
 grade_li.append(grade)
 # print(grade_li)
 break
 else:
 grade_cnt = grade_cnt + 1
 if grade_cnt == 3:
 grade = 1
 grade_li.append(grade)
 # print(grade_li)

 elif columns[i+3] in negative_rating:
 if row[columns[i+3]] >= data_list[i]:
 # print(row[data_age.columns.tolist()[i]], "입력값>=", data_list[i-3])
 if grade <= 5 - int(row['등급']):
 grade = 5 - int(row['등급'])
 grade_li.append(grade)
 # print(grade_li)
 break
 else:
 grade_cnt = grade_cnt + 1
 if grade_cnt == 3:
 grade = 1
 grade_li.append(grade)
 # print(grade_li)

 making_radar_chart(grade_li)

def making_radar_chart(grade_li):
 # 데이터 준비
 values = []
 num_vars = 0
 categories = []
 if len(grade_li) == 5:
 categories = ['심폐지구력', '순발력', '근지구력', '민첩성', '유연성']
 num_vars = len(categories)
 for i in range(len(grade_li)):
 values.append(grade_li[i])

 elif len(grade_li) == 6:
 categories = ['심폐지구력', '순발력', '근지구력', '민첩성', '유연성', '평형성']

```

```

 num_vars = len(categories)
 for i in range(len(grade_li)):
 values.append(grade_li[i])
각 카테고리의 각도 계산
angles = np.linspace(0, 2 * np.pi, num_vars, endpoint=False).tolist()
첫 번째 요소를 뒤로 복사하여 폐곡선을 만듭니다.
values += values[:1]
angles += angles[:1]

레이더 차트 그리기
plt.rc('font', family='Malgun Gothic')
fig, ax = plt.subplots(figsize=(6, 4), subplot_kw=dict(polar=True))
ax.fill(angles, values, color='green', alpha=0.25)

""" 각 카테고리별 계산된 총합 """

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
기입했을 때 레이더 차트 그려지도록

값이 높을수록 좋은 등급인 측정 항목 (~ 이상): 9, 12, 19, 20, 22, 23, 25
값이 낮을수록 좋은 등급인 측정 항목 (~ 이하): 13, 15, 21, 26, 27
positive_rating = ('MESURE_IEM_009_VALUE', 'MESURE_IEM_012_VALUE', 'MESURE_IEM_019_VALUE', 'MESURE_IEM_020_VALUE', 'MESURE_IEM_022_VALUE', 'MESURE_IEM_022',
negative_rating = ('MESURE_IEM_013_VALUE', 'MESURE_IEM_015_VALUE', 'MESURE_IEM_021_VALUE', 'MESURE_IEM_026_VALUE', 'MESURE_IEM_027_VALUE', 'MESURE_IEM_027

def elder_health_estimate(age,gender,value1, value2,value3,value4, value5,*value6):
 data_list = []
 # 기술표 성별,나이 범위 좁히기
 if age>=65:
 data_elder_gender = data_elder_criteria[data_elder_criteria['성별']==gender]
 if age>=85:
 data_age = data_elder_gender[data_elder_gender['나이']=='85~']
 else:
 ranges = [(i, i + 4) for i in range(65, 85, 5)]
 for start, end in ranges:
 #print(start,end)
 if start <= age <= end:
 #print(start,end)
 data_age = data_elder_gender[data_elder_gender['나이']==f'{start}~{end}']

 columns = data_age.columns.tolist()
 if value6:
 data_list = [value1, value2, value3, value4,value5]
 for value in value6:
 data_list.append(value)
 else:
 data_list = [value1, value2, value3, value4, value5]

#행 하나씩 내려가게
grade_li = []
for i in range(0, len(data_list)):
 .
 .
 .

```