In [1]:

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

In [2]:

```
df_17 = pd.read_csv('../PBL/dataset/NHIS_2017_2018_100m/NHIS_OPEN_GJ_2017_100
df_18 = pd.read_csv('../PBL/dataset/NHIS_2017_2018_100m/NHIS_OPEN_GJ_2018_100
```

In [3]:

```
valid_17 = df_17.loc[df_17['식전혈당(공복혈당)'] <= 300]
valid_18 = df_18.loc[df_18['식전혈당(공복혈당)'] <= 300]
```

In [4]:

```
valid_df = valid_17.append(valid_18)
valid_df.dropna(subset=['식전혈당(공복혈당)'], inplace=True)
```

In [5]:

```
def get_col_num(df):
    return df.columns.size
```

In [6]:

```
def get_row_num(df):
    return df.size / get_col_num(df)
```

In [7]:

```
def get_entropy(px):
    if px == 0:
        return 0;
    return px * np.log2(px)
```

In [8]:

```
def getParentEntropy(dataFrame) :
    row_num = get_row_num(dataFrame)
    col_num = get_col_num(dataFrame)
    idx = pd.RangeIndex(start=0, stop=row_num)
    dataFrame.index=idx
    confirmed_person = dataFrame.loc[dataFrame["식전혈당(공복혈당)"] >= 126]
    confirmed_mask = dataFrame["식전혈당(공복혈당)"] >= 126
    confirmed_size = confirmed_person.size / col_num
    unconfirmed_person = dataFrame.loc[dataFrame["식전혈당(공복혈당)"] < 126]
    unconfirmed_size = row_num - confirmed_size
    confirmed_px_root = confirmed_size / row_num
    unconfirmed_px_root = 1 - confirmed_px_root
    entropy_root = -1 * (get_entropy(confirmed_px_root) + get_entropy(unconfi
    return entropy_root
```

In [9]:

```
def getIG(dataFrame, col_name, value_list):
    sum_entropy = 0
    col_num = get_col_num(dataFrame)
    row_num = get_row_num(dataFrame)
    for val in value_list:
        df = dataFrame.loc[dataFrame[col_name] == val]
        df_size = df.size / col_num

        Y_df = df.loc[df["식전혈당(공복혈당)"] >= 126]

        Y_df_size = Y_df.size / col_num

        Y_px = Y_df_size / df_size
        N_px = 1 - Y_px
        rtn = -1 * (get_entropy(Y_px) + get_entropy(N_px))
        sum_entropy += rtn * (df_size / row_num)
    print("H(%s):"%(col_name), sum_entropy)
    return getParentEntropy(dataFrame) - sum_entropy
```

In [10]:

```
def getRangeIG(dataFrame, col_name, range_list):
    sum_entropy = 0
    prev_ran = 0
    col_num = get_col_num(dataFrame)
    row_num = get_row_num(dataFrame)
    for ran in range_list:
        df = dataFrame.loc[dataFrame[col_name] > prev_ran]
        df = df.loc[df[col_name] <= ran]
        df_size = df.size / col_num

        Y_df = df.loc[df["식전혈당(공복혈당)"] >= 126]

        Y_df_size = Y_df.size / col_num

        Y_px = Y_df_size / df_size
        N_px = 1 - Y_px

        rtn = -1 * (get_entropy(Y_px) + get_entropy(N_px))
        sum_entropy += rtn * (df_size / row_num)
        prev_ran = ran
    print("H(%s): "%(col_name), sum_entropy)
    return getParentEntropy(dataFrame) - sum_entropy
```

In [11]:

```
# print(valid_df["성별코드"].unique())
print("정보획득량: " , getIG(valid_df, "성별코드", valid_df["성별코드"].unique()))
```

```
H(성별코드): 0.3867554444550354
정보획득량:  0.003918456695817585
```

In [12]:

```
# print(valid_df["연령대코드(5세단위)"].unique())
print("정보획득량: ", getIG(valid_df, "연령대코드(5세단위)", valid_df["연령대코드(5세단위
```

```
H(연령대코드(5세단위)): 0.3672852841699641
정보획득량:  0.0233886169808889
```

In [13]:

```
# print(valid_df["신장(5Cm단위)"].unique())
print("정보획득량: ", getIG(valid_df, "신장(5Cm단위)", valid_df["신장(5Cm단위)"].unic
```

```
H(신장(5Cm단위)): 0.3893746207368564
정보획득량:  0.0012992804l3996573
```

In [14]:

```
print("정보획득량: ", getIG(valid_df, "체중(5Kg단위)", valid_df["체중(5Kg단위)"].unic
```

```
H(체중(5Kg단위)): 0.38498086687893274
정보획득량:  0.00569303427192025
```

In [15]:

```
bmi_ref = valid_df.loc[:,["체중(5Kg단위)", "신장(5Cm단위)"]]
bmi_df = (bmi_ref["체중(5Kg단위)"] / ((bmi_ref["신장(5Cm단위)"]/100)**2))
bmi_df = round(bmi_df)

copy_df_bmi = valid_df
copy_df_bmi.insert(get_col_num(copy_df_bmi), 'BMI', bmi_df)

bmi_list = [24.9, 29.9, 34.9]
```

```
print("정보획득량: ", getRangeIG(copy_df_bmi, "BMI", bmi_list))
```

```
H(BMI):  0.3787294284942003
정보획득량:  0.01194447265665266
```

In [16]:

```
copy_df_waist = valid_df
copy_df_waist.dropna(subset=["허리둘레"], inplace=True)
copy_df_waist = copy_df_waist.loc[copy_df_waist["허리둘레"] >= 35]
copy_df_waist = copy_df_waist.loc[copy_df_waist["허리둘레"] <= 111.76]

waist_list = [66.04, 71.12, 78.74, 83.82, 104.14, 106.68, 111.76]

print("정보획득량: ", getRangeIG(copy_df_waist, "허리둘레", waist_list))
```

```
H(허리둘레):  0.3718847314419549
정보획득량:  0.017810867741973213
```