**1번**

happy = pd.read\_csv("happy.csv")

**2번**

happy1.Economy = np.where(happy1.Economy.isna(),

np.mean(happy.Economy),

happy.Economy)

happy1.Family = np.where(happy1.Family.isna(),

np.mean(happy.Family),

happy.Family)

happy1.Health = np.where(happy1.Health.isna(),

np.mean(happy.Health),

happy.Health)

happy1.Freedom = np.where(happy1.Freedom.isna(),

np.mean(happy.Freedom),

happy.Freedom)

happy1.Trust = np.where(happy1.Trust.isna(),

np.mean(happy.Trust),

happy.Trust)  
Score 5.370728

Economy 0.982367

Family 0.996709

Health 0.667917

Freedom 0.405415

Trust 0.176356

Generosity 0.313749

Residual 2.131015

**3번**

7개

**4번**

8개

**5번**

from sklearn.neighbors import LocalOutlierFactor

happy2 = happy1.drop(columns = ["Rating", "Grade"])

from sklearn.neighbors import LocalOutlierFactor

lof1 = LocalOutlierFactor()

lof1.fit(happy2)

x =happy2.loc[lof1.negative\_outlier\_factor\_>-2,:]

y = happy1[["Rating", "Grade"]][lof1.negative\_outlier\_factor\_>-2]

from sklearn.model\_selection import train\_test\_split

tr\_x, te\_x, tr\_y, te\_y = train\_test\_split(x.drop(columns = "Score"), x.Score, test\_size = 0.3,

random\_state = 1234)

학습용 데이터 수 : 323개  
테스트용 데이터 수 : 139개

**6번**

from sklearn.preprocessing import MinMaxScaler

minmax1 = MinMaxScaler()

tr\_xs = tr\_x.copy()

te\_xs = te\_x.copy()

minmax1.fit(tr\_x)

tr\_xs = minmax1.transform(tr\_xs)

te\_xs = minmax1.transform(te\_xs)

**7번**

from sklearn.linear\_model import LinearRegression

lm\_model=LinearRegression()

lm\_model.fit(tr\_xs, tr\_y)

lm\_model.coef\_

lm\_model.intercept\_

y = 0.6134335622624434 + 0.06995891 \* Economy + 1.68938835 \* Family + 2.37707406 \* Health + 0.86148334 \* Freedom + 0.87613155 \* Trust + -0.15894256 \* Generosity + 3.21602393 \* Residual

**8번**

pred1=lm\_model.predict(te\_xs)

np.sum(pred1)

748.426006265056

**9번**

from sklearn.metrics import mean\_squared\_error

mean\_squared\_error(te\_y, pred1)

0.07492912307151056

**10번**

lm\_model.score(te\_xs, te\_y)

0.9482953143974799