

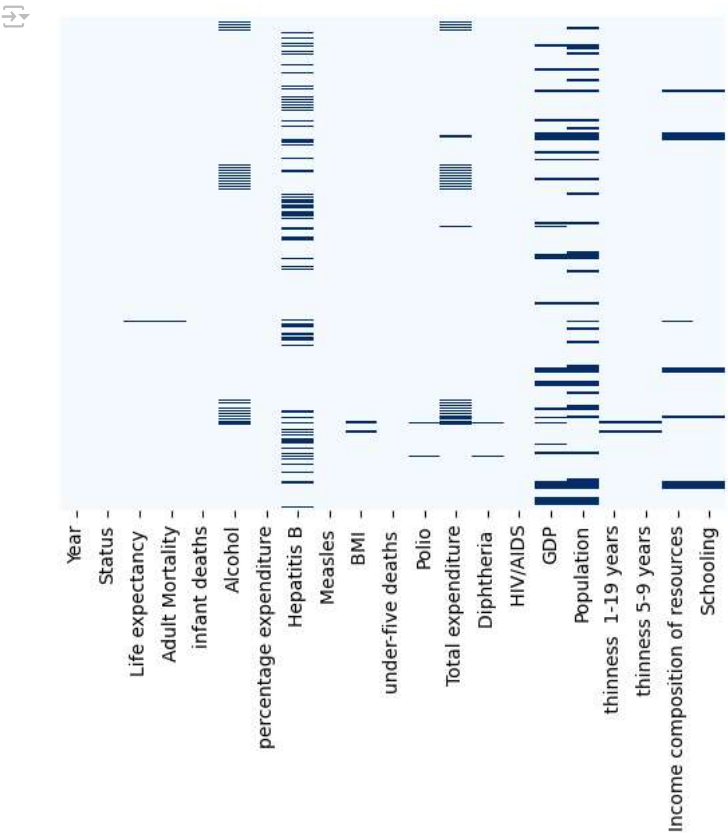
Proyecto Final Dia 3

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns

1 df_life_expectancy = pd.read_csv('Life_Expectancy_Data-230526-192351.csv')
2 df_life_expectancy
```

	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	...	Polio	Total expenditure	Diphtheria
0	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	19.1	...	6.0	8.16	65.0
1	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	18.6	...	58.0	8.18	62.0
2	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	18.1	...	62.0	8.13	64.0
3	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	17.6	...	67.0	8.52	67.0
4	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	17.2	...	68.0	7.87	68.0
...
2933	2004	Developing	44.3	723.0	27	4.36	0.000000	68.0	31	27.1	...	67.0	7.13	65.0
2934	2003	Developing	44.5	715.0	26	4.06	0.000000	7.0	998	26.7	...	7.0	6.52	68.0
2935	2002	Developing	44.8	73.0	25	4.43	0.000000	73.0	304	26.3	...	73.0	6.53	71.0
2936	2001	Developing	45.3	686.0	25	1.72	0.000000	76.0	529	25.9	...	76.0	6.16	75.0
2937	2000	Developing	46.0	665.0	24	1.68	0.000000	79.0	1483	25.5	...	78.0	7.10	78.0

```
1 sns.heatmap(df_life_expectancy.isnull(), yticklabels=False, cbar=False, cmap='Blues');
```



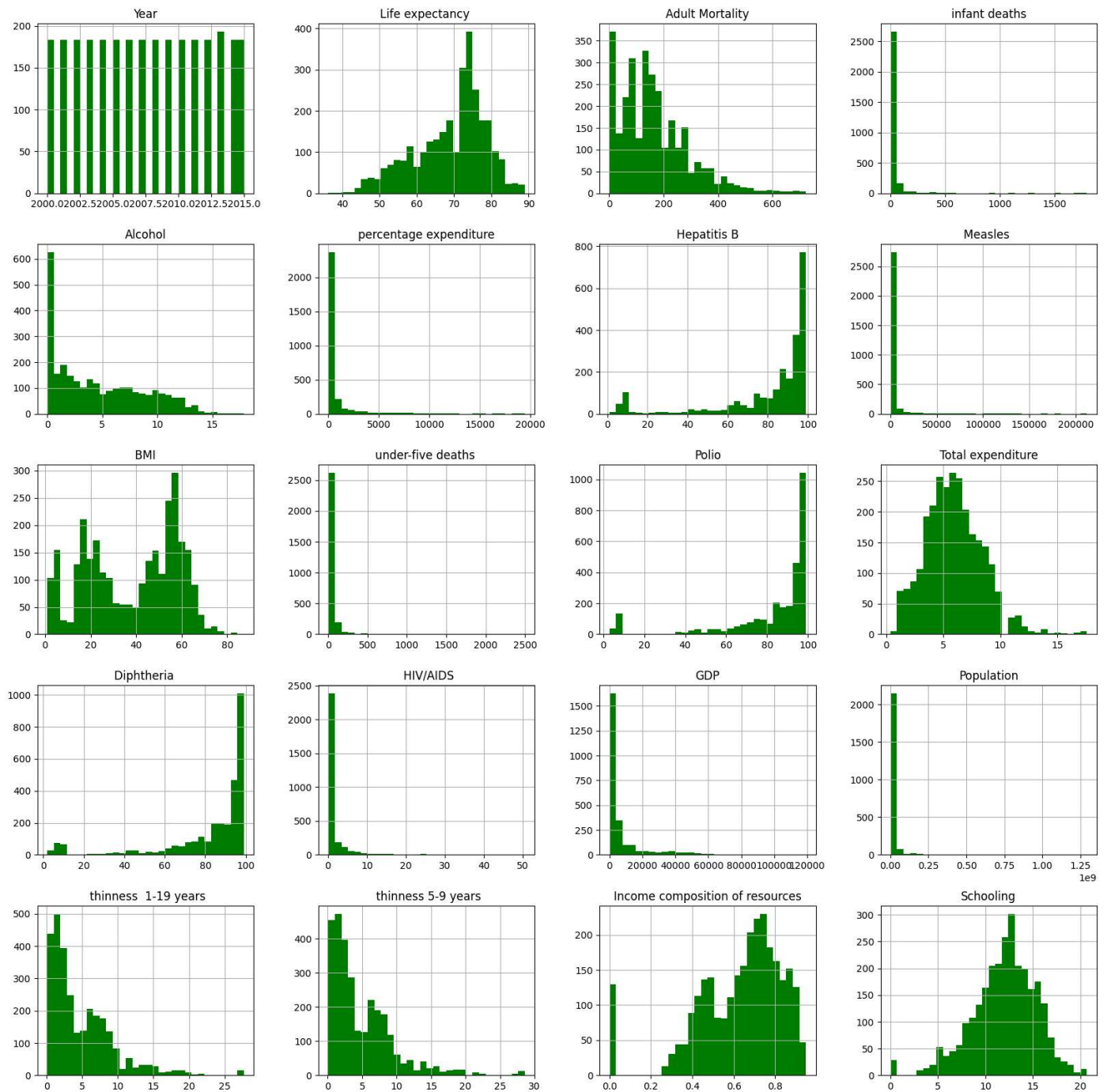
```
1 df_life_expectancy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 21 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Year                                2938 non-null   int64
1   Status                             2938 non-null   object
2   Life expectancy                    2928 non-null   float64
3   Adult Mortality                    2928 non-null   float64
4   infant deaths                      2938 non-null   int64
5   Alcohol                            2744 non-null   float64
6   percentage expenditure             2938 non-null   float64
7   Hepatitis B                        2385 non-null   float64
8   Measles                            2938 non-null   int64
9   BMI                                2904 non-null   float64
10  under-five deaths                  2938 non-null   int64
11  Polio                             2919 non-null   float64
12  Total expenditure                  2712 non-null   float64
13  Diphtheria                         2919 non-null   float64
14  HIV/AIDS                           2938 non-null   float64
15  GDP                                2490 non-null   float64
16  Population                         2286 non-null   float64
17  thinness 1-19 years                2904 non-null   float64
18  thinness 5-9 years                 2904 non-null   float64
19  Income composition of resources    2771 non-null   float64
20  Schooling                          2775 non-null   float64
dtypes: float64(16), int64(4), object(1)
memory usage: 482.1+ KB
```

```
1 df_life_expectancy.describe()
```

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000	2938.000000	2904.000000	2938.000000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.940461	2419.592240	38.321247	42.035739
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070016	11467.272489	20.044034	160.445548
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.000000	0.000000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.000000	0.000000	19.300000	0.000000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.000000	17.000000	43.500000	4.000000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.000000	360.250000	56.200000	28.000000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	87.300000	2500.000000

```
1 df_life_expectancy.hist(bins=30, figsize=(20,20), color='g');
```

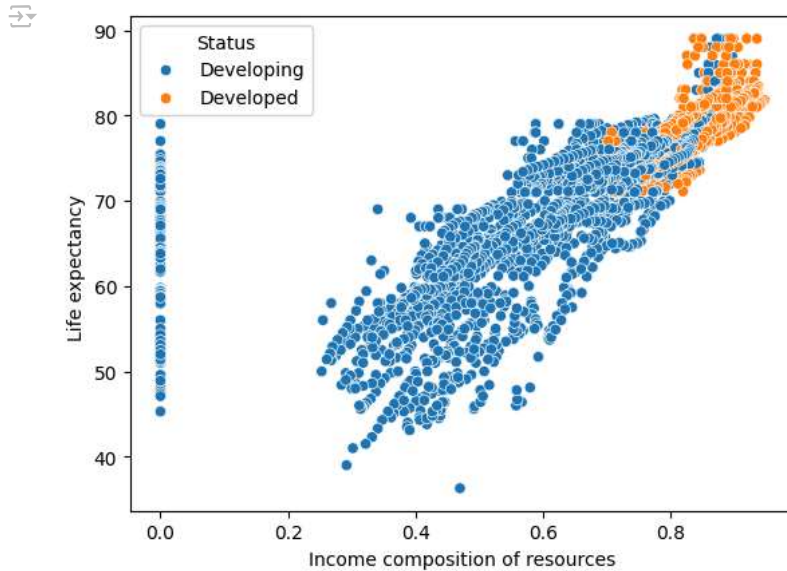


```

1 plt.figure(figsize=(20,20))
2 sns.pairplot(df_life_expectancy);

1 sns.scatterplot(x='Income composition of resources', y='Life expectancy ', hue='Status', data=df_life_expectancy);

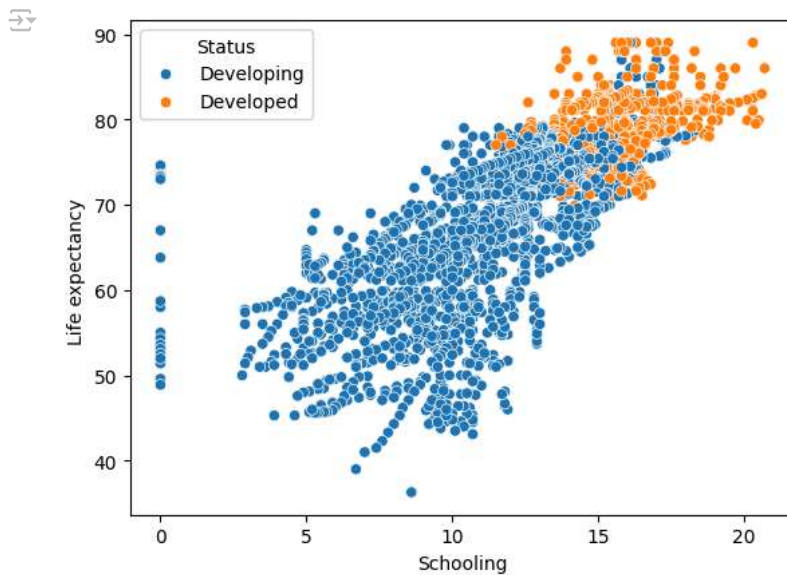
```



```

1 sns.scatterplot(x='Schooling', y='Life expectancy ', hue='Status', data=df_life_expectancy);

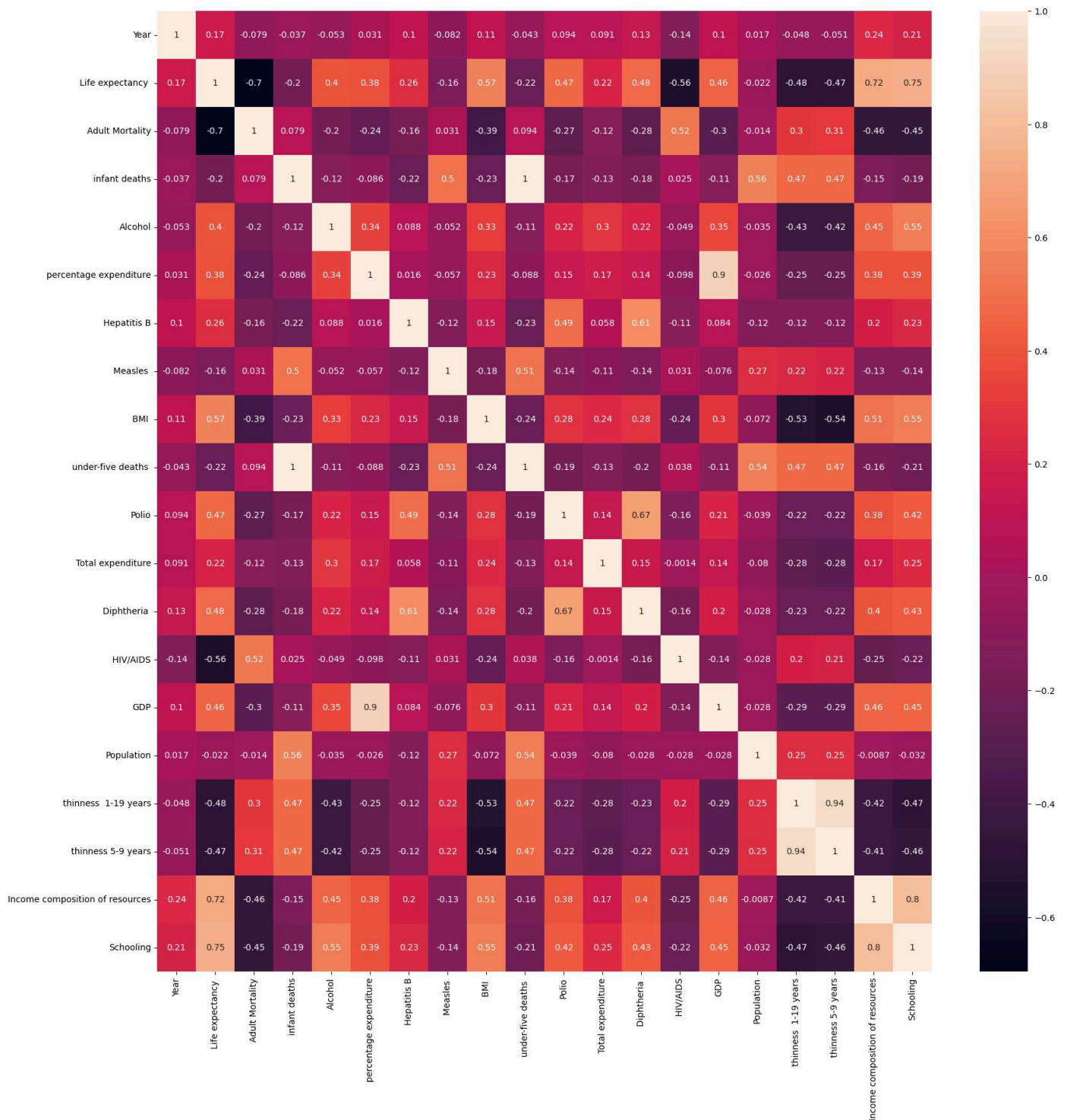
```



```

1 plt.figure(figsize=(20,20))
2 sns.heatmap(df_life_expectancy.corr(numeric_only=True), annot=True);

```



```
1 df_life_expectancy['Status'].value_counts(dropna=True)
```

```
Status
Developing    2426
Developed      512
Name: count, dtype: int64
```

```
1 df_life_expectancy = pd.get_dummies(df_life_expectancy, columns=['Status'], dtype=int)
```

```
1 df_life_expectancy
```

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	...	Diphtheria	HIV/AIDS	GDP
0	2015	65.0	263.0	62	0.01	71.279624	65.0	1154	19.1	83	...	65.0	0.1	584.259210
1	2014	59.9	271.0	64	0.01	73.523582	62.0	492	18.6	86	...	62.0	0.1	612.696514
2	2013	59.9	268.0	66	0.01	73.219243	64.0	430	18.1	89	...	64.0	0.1	631.744976
3	2012	59.5	272.0	69	0.01	78.184215	67.0	2787	17.6	93	...	67.0	0.1	669.959000
4	2011	59.2	275.0	71	0.01	7.097109	68.0	3013	17.2	97	...	68.0	0.1	63.537231
...
2933	2004	44.3	723.0	27	4.36	0.000000	68.0	31	27.1	42	...	65.0	33.6	454.366654
2934	2003	44.5	715.0	26	4.06	0.000000	7.0	998	26.7	41	...	68.0	36.7	453.351155
2935	2002	44.8	73.0	25	4.43	0.000000	73.0	304	26.3	40	...	71.0	39.8	57.348340
2936	2001	45.3	686.0	25	1.72	0.000000	76.0	529	25.9	39	...	75.0	42.1	548.587312
2937	2000	46.0	665.0	24	1.68	0.000000	79.0	1483	25.5	39	...	78.0	43.5	547.358879

```
1 null_counts = df_life_expectancy.isnull().sum()
2 null_counts[null_counts != 0]
```

```
Life expectancy      10
Adult Mortality      10
Alcohol             194
Hepatitis B         553
BMI                  34
Polio                19
Total expenditure    226
Diphtheria           19
GDP                  448
Population           652
thinness 1-19 years   34
thinness 5-9 years    34
Income composition of resources 167
Schooling            163
dtype: int64
```

```
1 df_life_expectancy.fillna(df_life_expectancy.mean(), inplace=True)
```

```
1 df_life_expectancy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                                     Non-Null Count  Dtype
---  -
 0   Year                                     2938 non-null   int64
 1   Life expectancy                         2938 non-null   float64
 2   Adult Mortality                        2938 non-null   float64
 3   infant deaths                          2938 non-null   int64
 4   Alcohol                                2938 non-null   float64
 5   percentage expenditure                 2938 non-null   float64
 6   Hepatitis B                           2938 non-null   float64
 7   Measles                                2938 non-null   int64
 8   BMI                                    2938 non-null   float64
 9   under-five deaths                     2938 non-null   int64
10   Polio                                 2938 non-null   float64
11   Total expenditure                     2938 non-null   float64
12   Diphtheria                           2938 non-null   float64
13   HIV/AIDS                             2938 non-null   float64
14   GDP                                   2938 non-null   float64
15   Population                            2938 non-null   float64
16   thinness 1-19 years                   2938 non-null   float64
17   thinness 5-9 years                   2938 non-null   float64
18   Income composition of resources       2938 non-null   float64
19   Schooling                             2938 non-null   float64
20   Status_Developed                      2938 non-null   int32
21   Status_Developing                     2938 non-null   int32
dtypes: float64(16), int32(2), int64(4)
memory usage: 482.1 KB
```

```
1 X = df_life_expectancy.drop(['Life expectancy'], axis=1)
2 y = df_life_expectancy[['Life expectancy']]
```

```
1 print('X:', X.shape)
2 print('y:', y.shape)
```

```
X: (2938, 21)
y: (2938, 1)
```

```
1 X = np.array(X).astype('float32')
2 y = np.array(y).astype('float32')
```

```
1 from sklearn.model_selection import train_test_split
2
3 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
1 import xgboost as xgb
2
3 model = xgb.XGBRegressor(objective='reg:squarederror', learning_rate=0.1, max_depth=30, n_estimators=100)
4 model.fit(X_train, y_train)
```

```
XGBRegressor
XGBRegressor(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=0.1, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=30, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=100, n_jobs=None,
              num_parallel_tree=None, random_state=None, ...)
```

```
1 model_accuracy = model.score(X_test, y_test)
2 model_accuracy
```

```
0.9557170271873474
```

```
1 y_predict = model.predict(X_test)
2 y_predict
```

```
array([72.34404 , 73.6362 , 76.22628 , 82.02471 , 68.35663 , 74.817924,  
80.3385 , 48.211582, 47.817608, 46.588158, 72.30148 , 63.123257,  
75.25701 , 67.22251 , 75.57087 , 71.41098 , 66.90883 , 64.26366 ,  
78.8844 , 72.219284, 71.44145 , 79.41251 , 63.84498 , 45.22156 ,  
68.33219 , 79.537865, 68.029884, 63.440365, 71.299255, 74.69634 ,  
76.94762 , 55.731453, 81.47984 , 73.69475 , 65.15339 , 72.399254,  
56.082493, 80.464775, 64.07231 , 52.437553, 71.14539 , 65.3044 ,  
68.11952 , 70.44313 , 57.633884, 49.296715, 52.15927 , 73.68431 ,  
73.50497 , 64.10023 , 82.21835 , 78.325356, 75.6751 , 73.946846,  
57.704445, 67.98306 , 74.13705 , 59.060413, 81.99794 , 72.565155,  
80.32465 , 80.18442 , 69.86382 , 78.93037 , 73.233345, 78.69052 ,  
48.54511 , 72.36069 , 82.37233 , 82.82037 , 81.73522 , 47.317467,  
74.78615 , 80.49802 , 75.38489 , 64.45316 , 72.53273 , 50.41102 ,  
74.773315, 82.14768 , 79.44498 , 51.94226 , 80.38384 , 78.02773 ,  
66.534546, 54.033016, 53.970276, 81.934616, 64.177574, 62.27521 ,  
76.12829 , 73.12171 , 54.060616, 77.66735 , 55.891727, 81.28307 ,  
79.47828 , 71.81165 , 56.968998, 83.37486 , 64.83048 , 78.18673 ,  
50.32807 , 66.816765, 82.19401 , 81.99499 , 66.86196 , 65.67265 ,  
57.720802, 47.481705, 64.58998 , 77.853546, 57.973427, 61.727688,  
63.603115, 72.17697 , 73.57714 , 83.42852 , 70.82494 , 68.98681 ,  
74.03657 , 70.82701 , 67.689835, 53.209488, 72.242 , 69.774864,  
78.25067 , 76.1236 , 77.70773 , 63.912518, 73.48328 , 73.65152 ,  
54.906322, 83.15115 , 75.69963 , 76.097755, 73.46032 , 67.18154 ,  
76.57 , 74.44935 , 82.73757 , 73.214905, 72.96752 , 82.68314 ,  
71.183624, 76.34852 , 70.525475, 73.40732 , 74.3284 , 73.577156,  
79.86605 , 74.14464 , 67.9605 , 58.26734 , 66.42389 , 59.588795,  
76.34755 , 82.735344, 72.113556, 80.62812 , 67.55864 , 60.83968 ,  
66.74065 , 74.55515 , 67.78601 , 73.079094, 69.23971 , 66.75159 ,  
71.98704 , 72.741516, 60.78055 , 61.082306, 68.590096, 73.037895,  
73.040634, 71.2712 , 63.651352, 75.58788 , 57.271393, 74.228775,  
81.27091 , 57.16971 , 51.40678 , 69.50811 , 48.301033, 73.174644,  
76.192116, 78.14747 , 68.81092 , 50.41485 , 65.379166, 71.641815,  
73.958046, 73.58436 , 60.09388 , 72.68704 , 69.448616, 51.77792 ,  
56.07046 , 68.94853 , 78.6975 , 59.196037, 79.15002 , 74.3501 ,  
63.202522, 76.66953 , 69.526344, 70.321495, 81.41262 , 54.934452,  
72.9603 , 59.51094 , 58.3834 , 76.19869 , 77.18455 , 73.46268 ,  
72.53199 , 73.79523 , 70.41163 , 76.54851 , 68.29229 , 75.00961 ,  
74.29511 , 74.79149 , 56.443607, 80.947075, 78.95948 , 81.41397 ,  
83.28882 , 72.654564, 74.31819 , 68.0008 , 76.261215, 72.74644 ,  
59.6883 , 74.58129 , 69.5978 , 70.2261 , 62.339787, 76.53025 ,  
74.66133 , 71.74152 , 69.24922 , 74.60299 , 73.16254 , 63.533085,  
76.16753 , 64.66317 , 62.33299 , 78.94081 , 68.362274, 64.85788 ,  
60.195007, 86.1148 , 74.783134, 73.154305, 76.65435 , 52.692623,  
80.62664 , 82.145546, 82.954575, 83.22169 , 73.385185, 56.11716 ,  
62.668705, 54.335354, 78.91987 , 74.367455, 72.59049 , 63.031864,  
75.29373 , 68.56331 , 73.80562 , 57.24064 , 70.169426, 54.10603 ,  
67.68885 , 73.1696 , 72.85659 , 58.82177 , 64.10812 , 70.02335 ,  
70.10513 , 75.23104 , 67.655624, 64.36277 , 51.31099 , 72.84503 ,  
64.32587 , 71.17091 , 62.808865, 73.1279 , 65.22461 , 76.1162 ,  
80.92714 , 82.034546, 61.63478 , 66.40296 , 72.38784 , 72.30123 ,  
73.210045, 74.074 , 71.65926 , 72.034706, 71.045044, 73.749146,  
75.44628 , 73.366516, 72.65401 , 71.68153 , 49.38367 , 82.56156 ,  
76.05921 , 74.39073 , 80.44011 , 68.8298 , 53.81882 , 52.90916 ,  
52.82172 , 70.25594 , 75.08934 , 47.704327, 52.551476, 72.040085,  
72.96732 , 52.698544, 57.959408, 47.600132, 52.59084 , 60.22685 ,  
58.24155 , 69.43961 , 75.9854 , 74.002075, 72.572235, 64.64194 ,  
78.50521 , 83.38497 , 72.52291 , 64.00236 , 82.81041 , 73.0712 ,  
74.67162 , 78.40395 , 59.941578, 79.33904 , 73.27477 , 73.219215,
```

```
1 plt.plot(y_test, y_predict, '^', color='g');
```