Logistic Regression

applying regression on the given dataset

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, accuracy_score
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df= pd.read_csv('Life Expectancy Data.csv')
df.head()
df.describe()
```

Out[2]:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.00000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.94046
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.07001
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.00000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.00000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.00000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.00000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.00000

In [3]:
```
categorical= df.select_dtypes(include= "O")
numerical= df.select_dtypes(exclude= "O")
categorical.describe()
```

Out[3]:

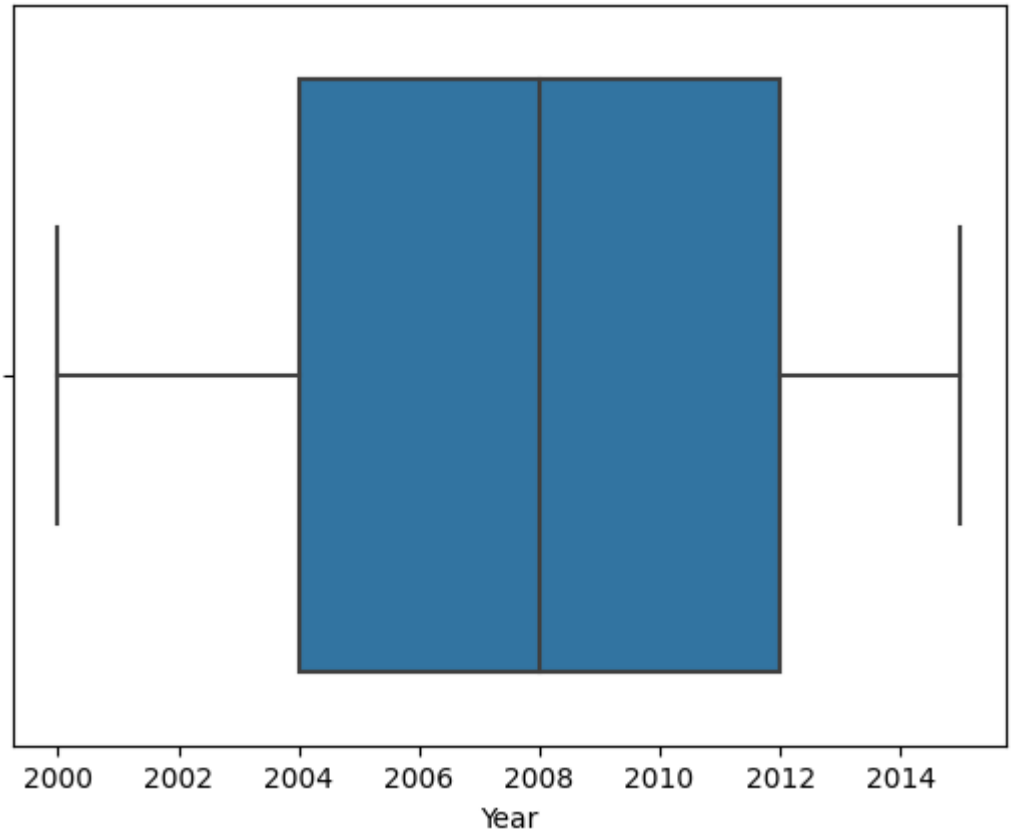
	Country	Status
count	2938	2938
unique	193	2
top	Afghanistan	Developing
freq	16	2426

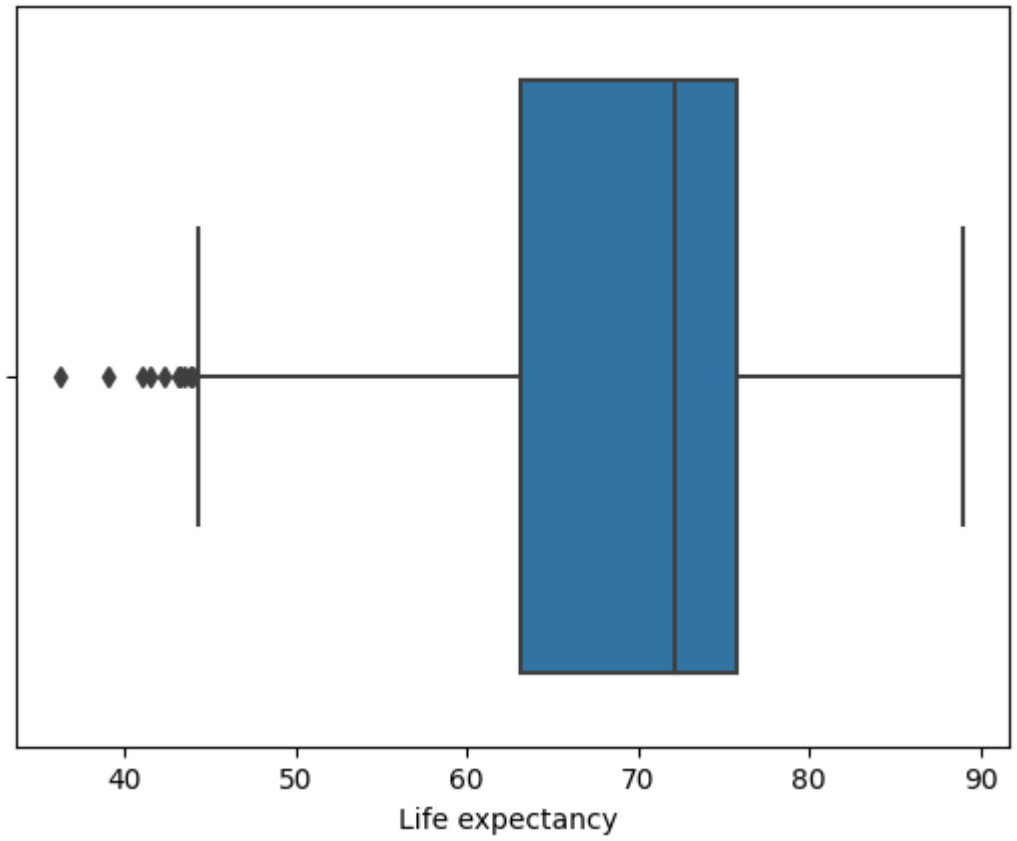
In [4]:
```
numerical.describe()
```

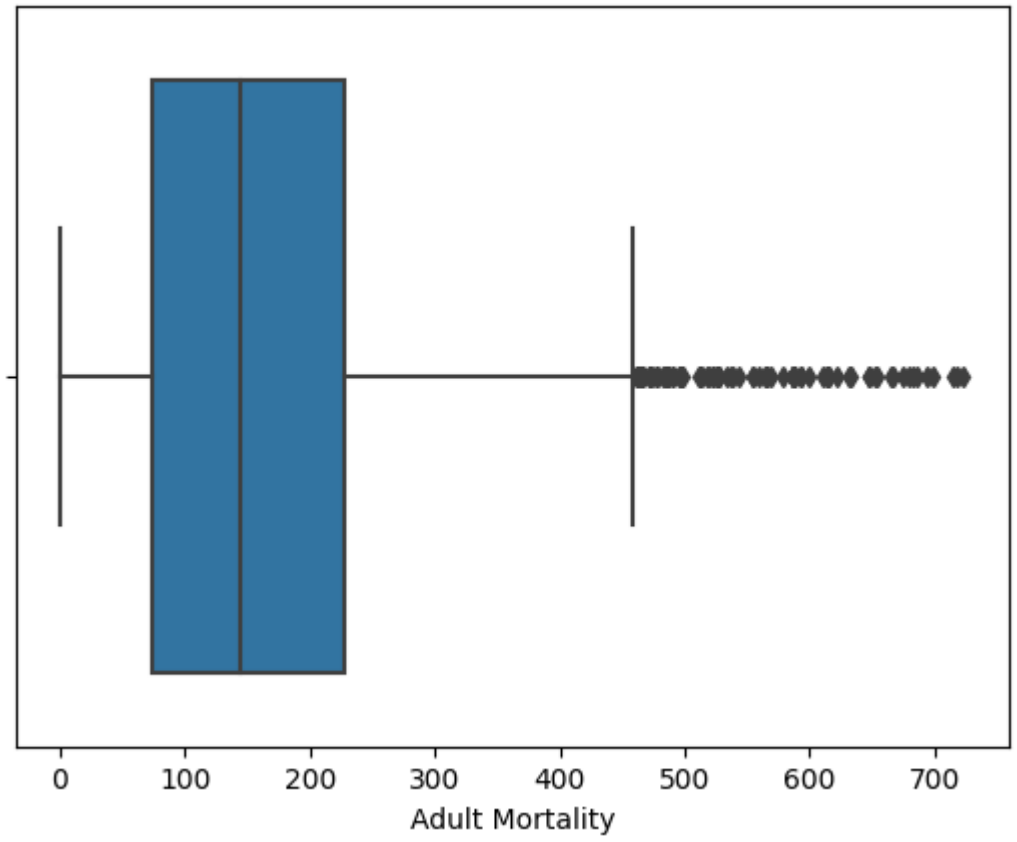
Out[4]:

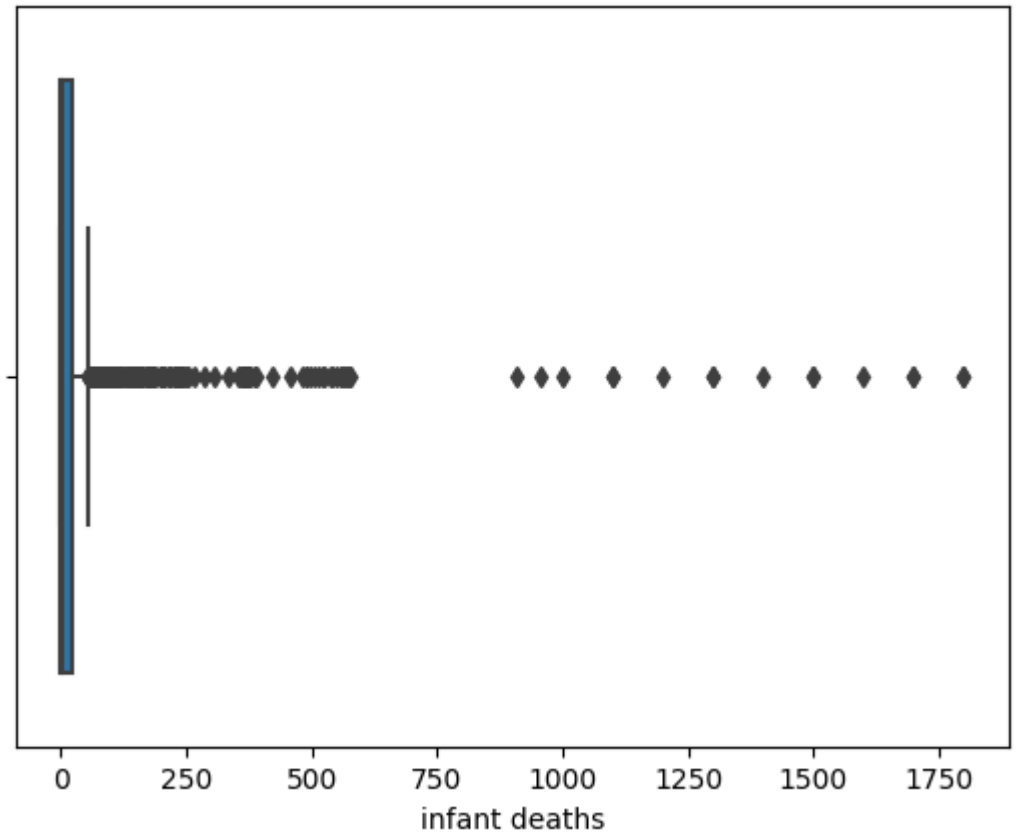
	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.00000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.94046
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.07001
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.00000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.00000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.00000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.00000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.00000

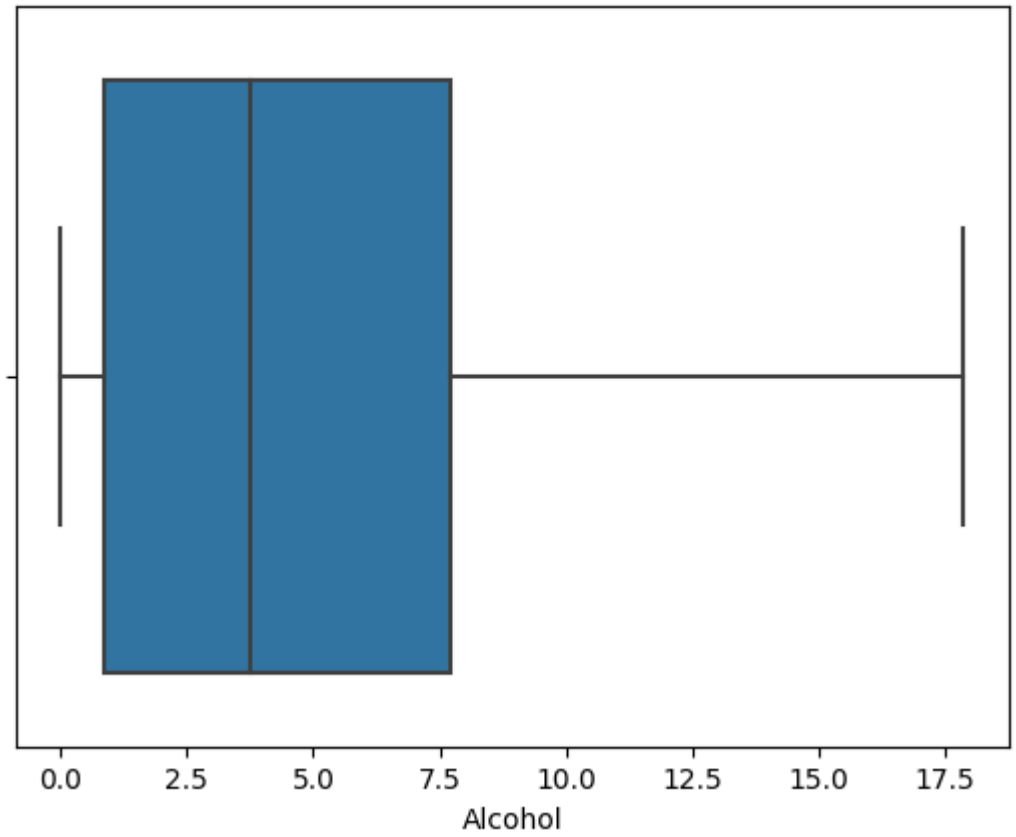
In [5]:
```
for feature in numerical.columns:
    sns.boxplot(x=numerical[feature])
    plt.show()
```

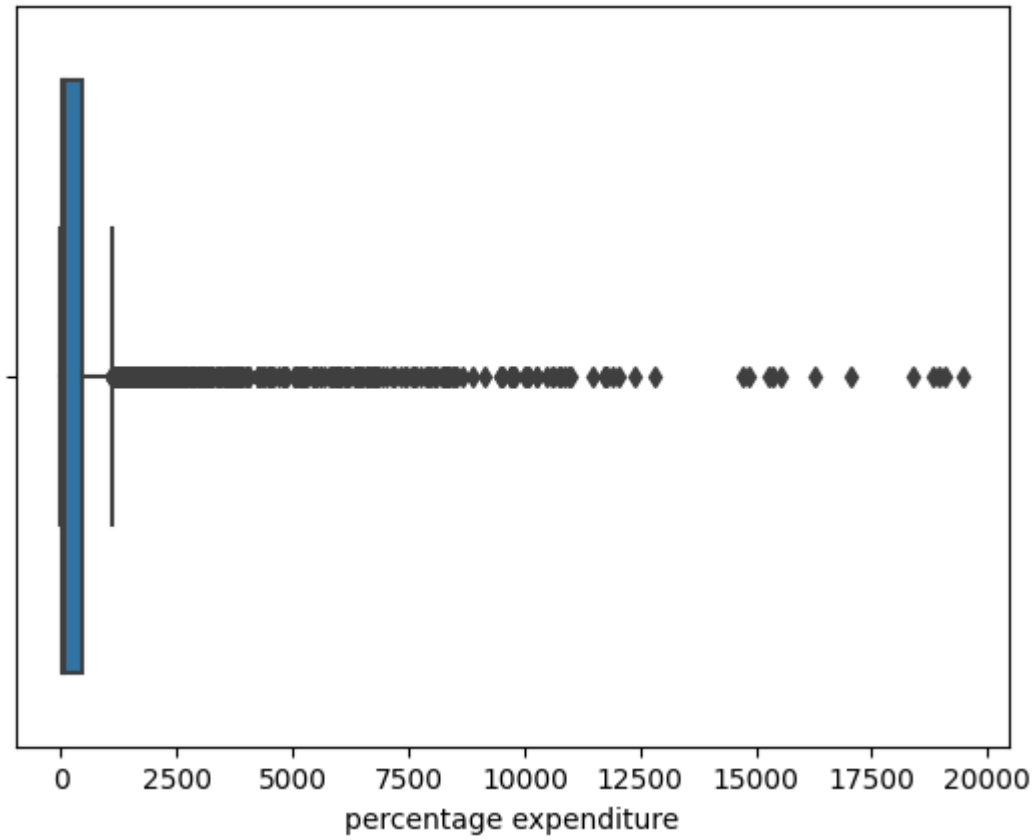


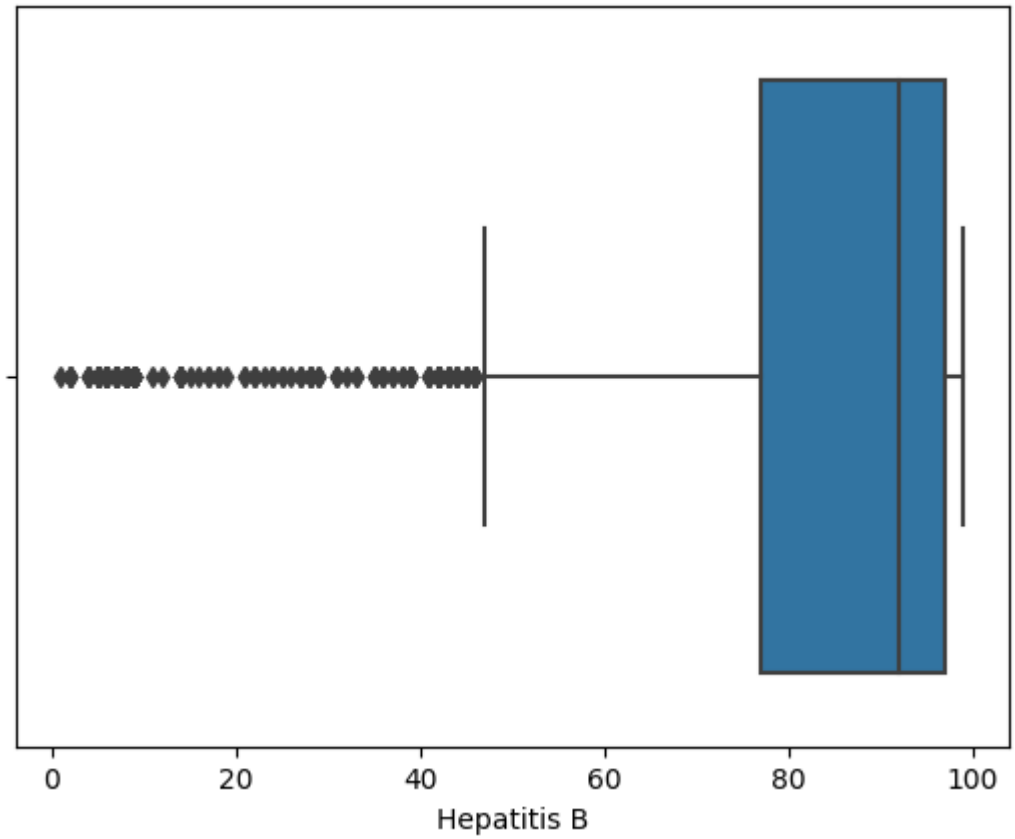

40
50
60
70
80
90
Life expectancy

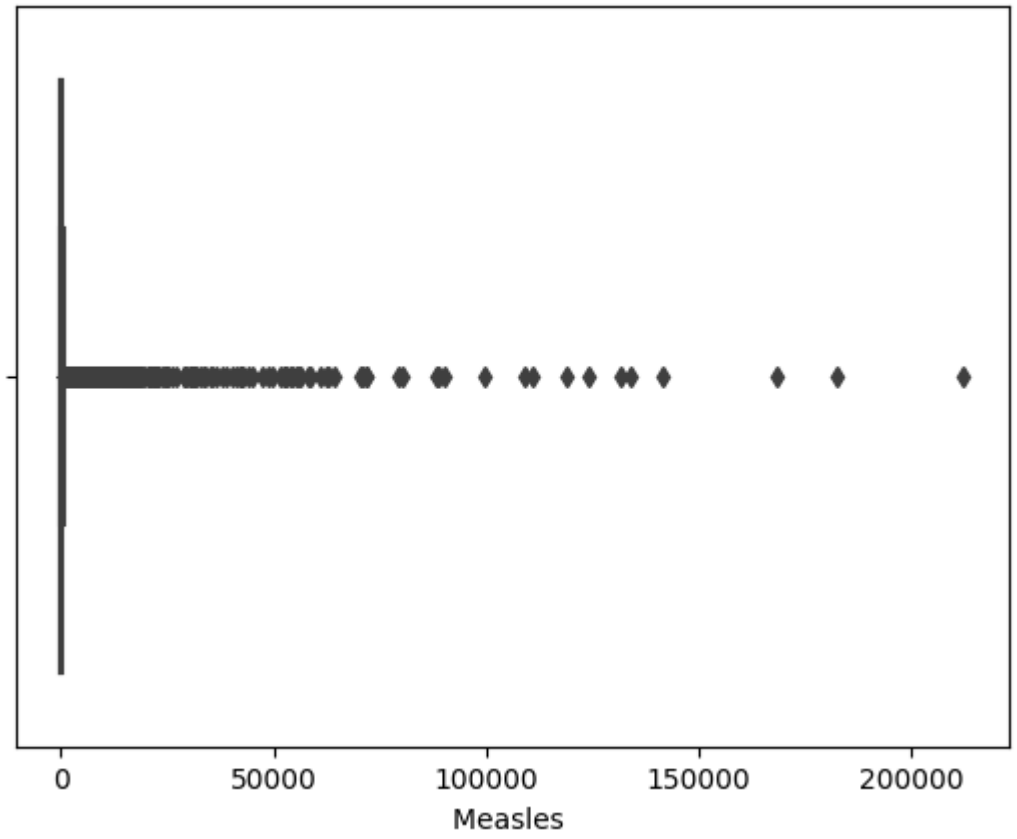

0
100
200
300
400
500
600
700
Adult Mortality

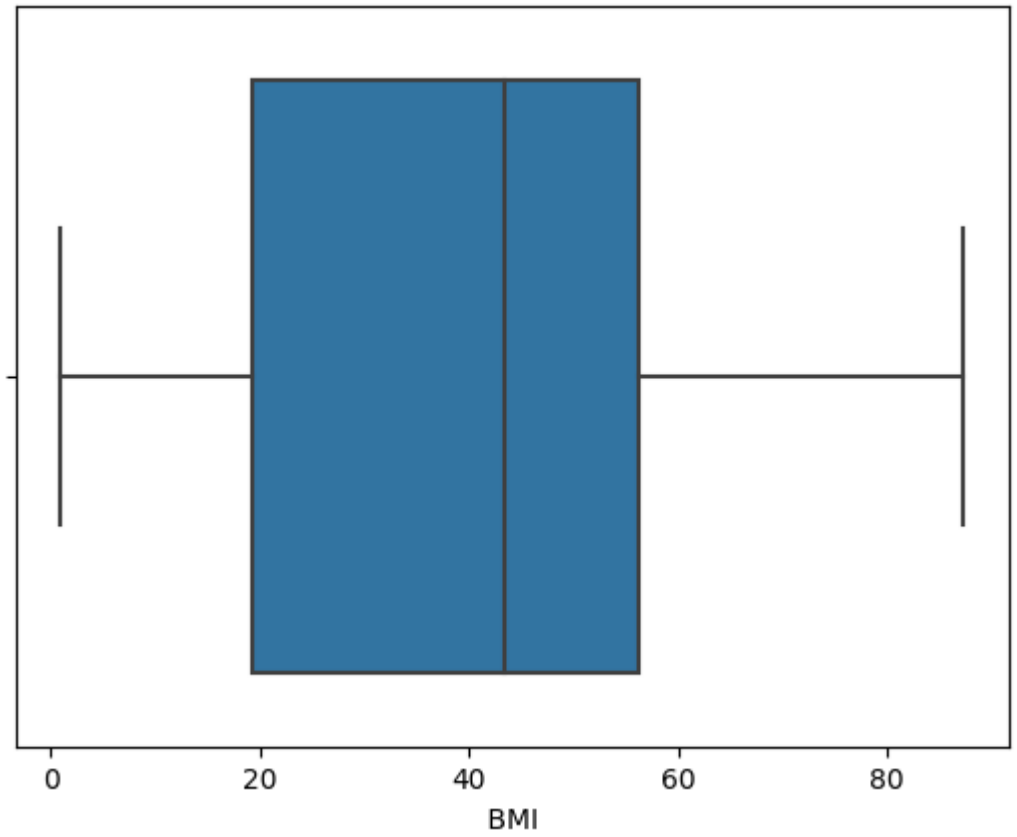

0
250
500
750
1000
1250
1500
1750
infant deaths

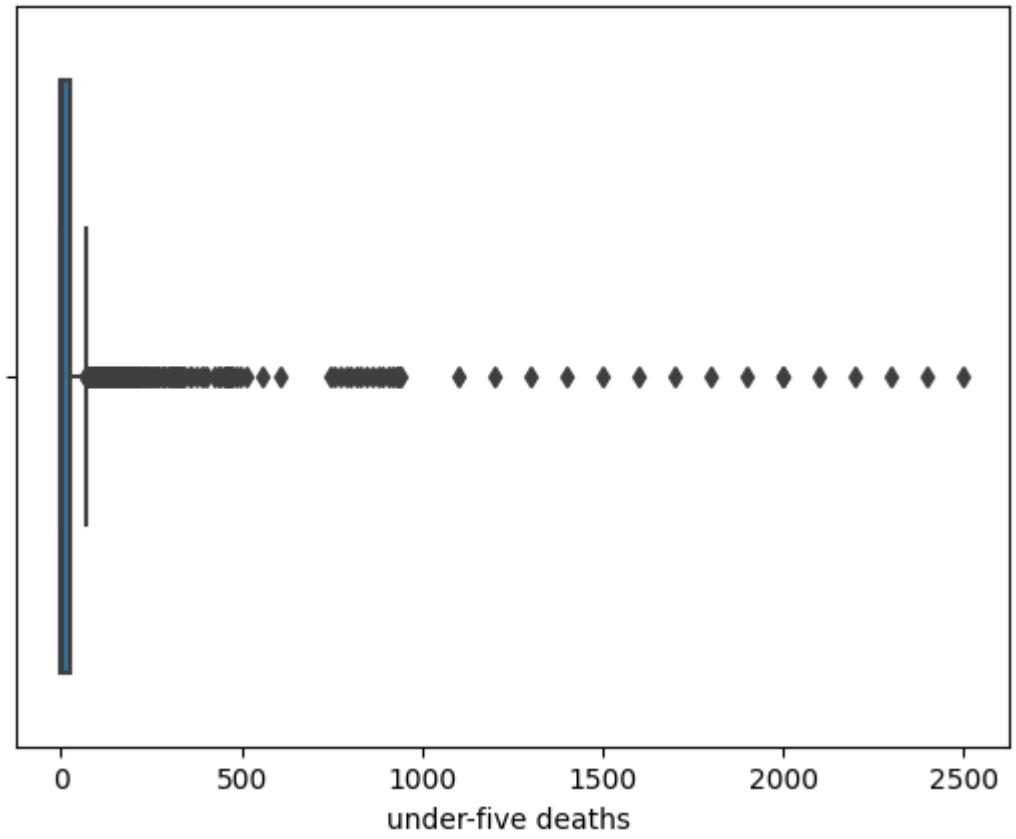

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Alcohol

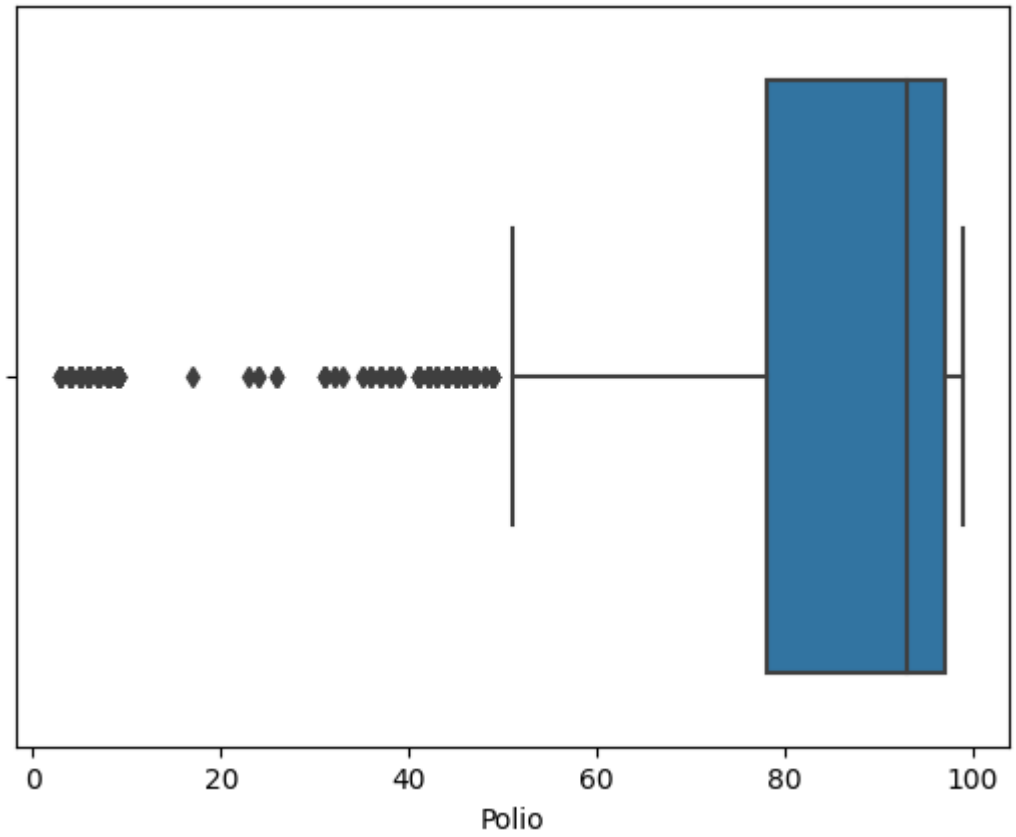

0
2500
5000
7500
10000
12500
15000
17500
20000
percentage expenditure

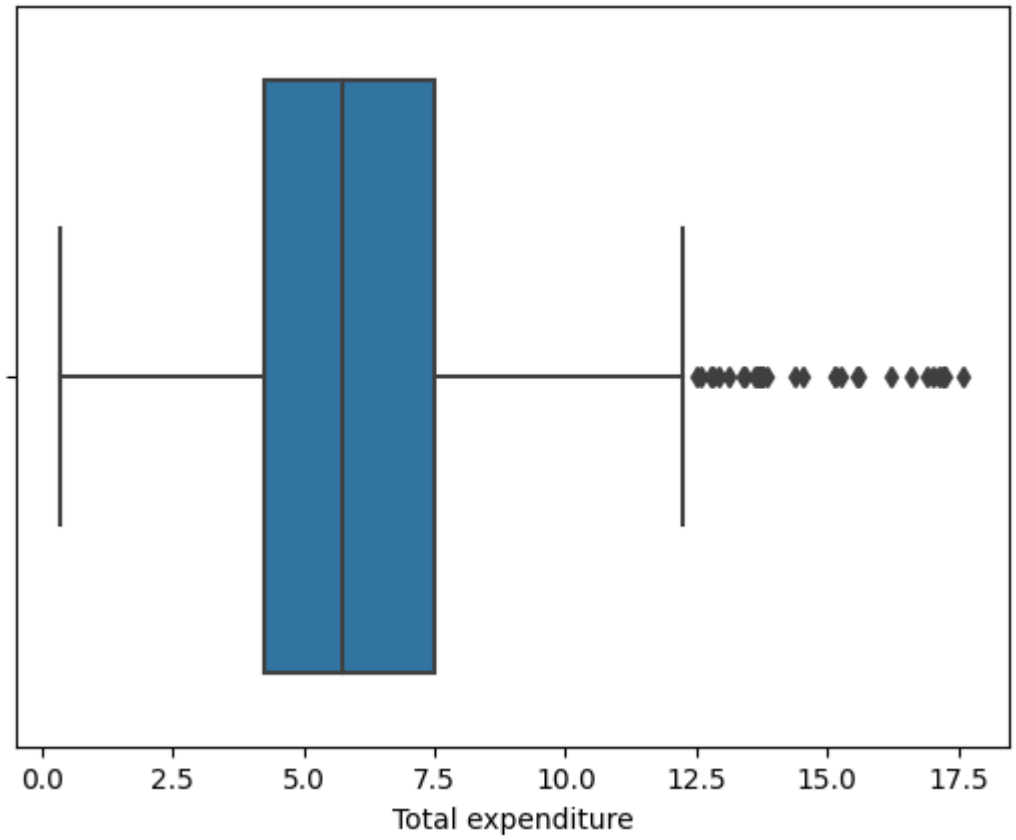

0
20
40
60
80
100
Hepatitis B

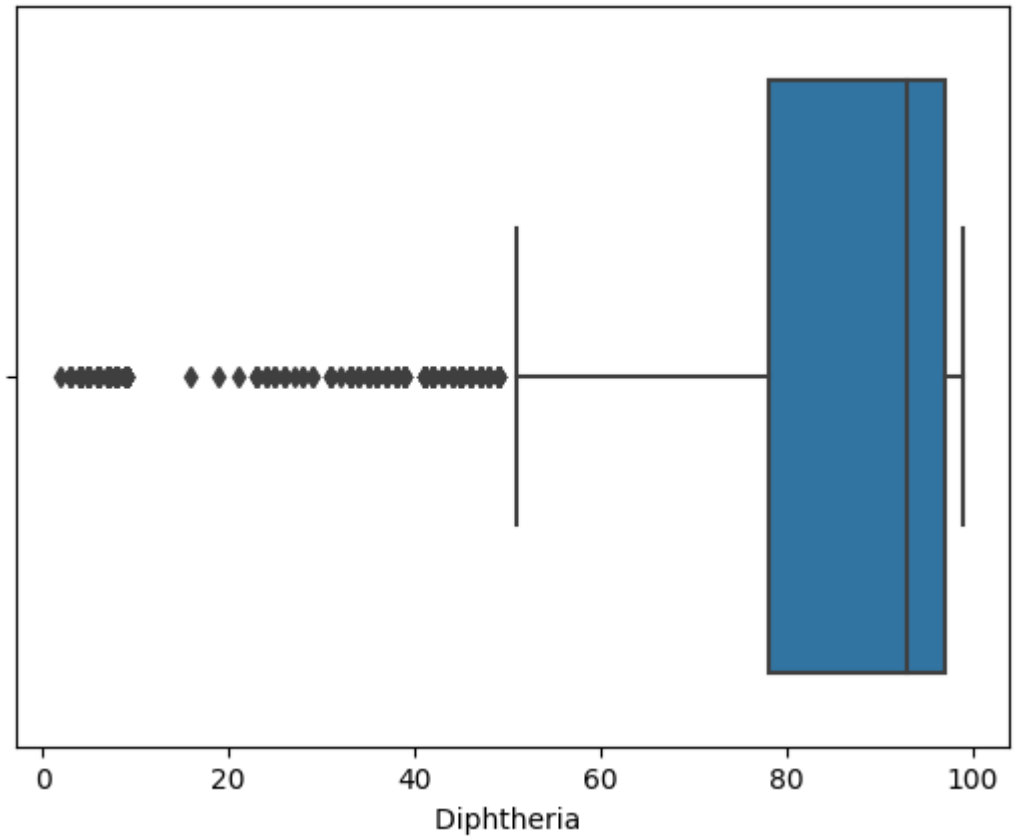

0
50000
100000
150000
200000
Measles

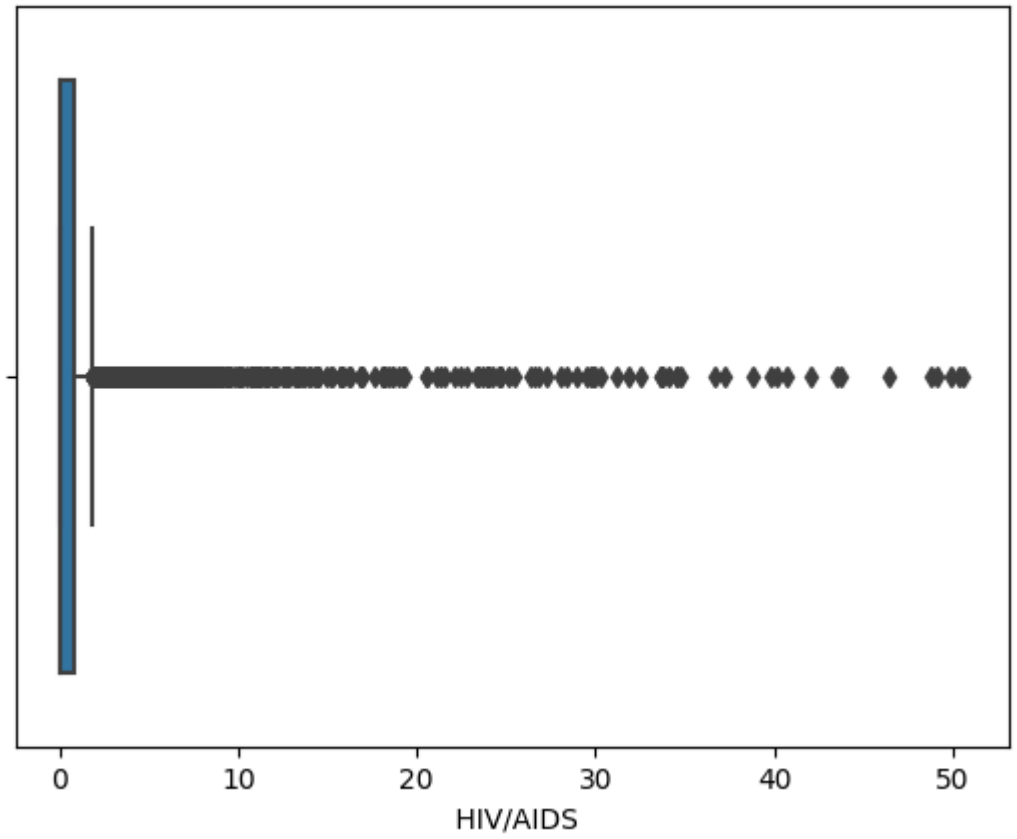

0
20
40
60
80
BMI

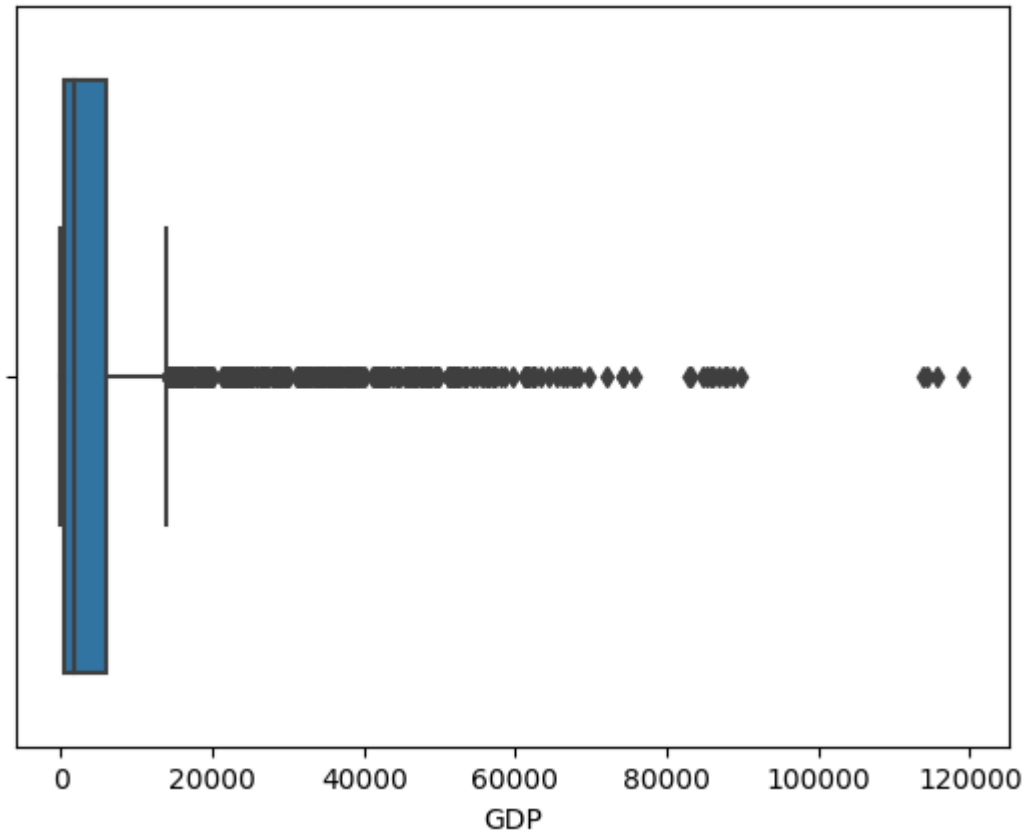

0
500
1000
1500
2000
2500
under-five deaths

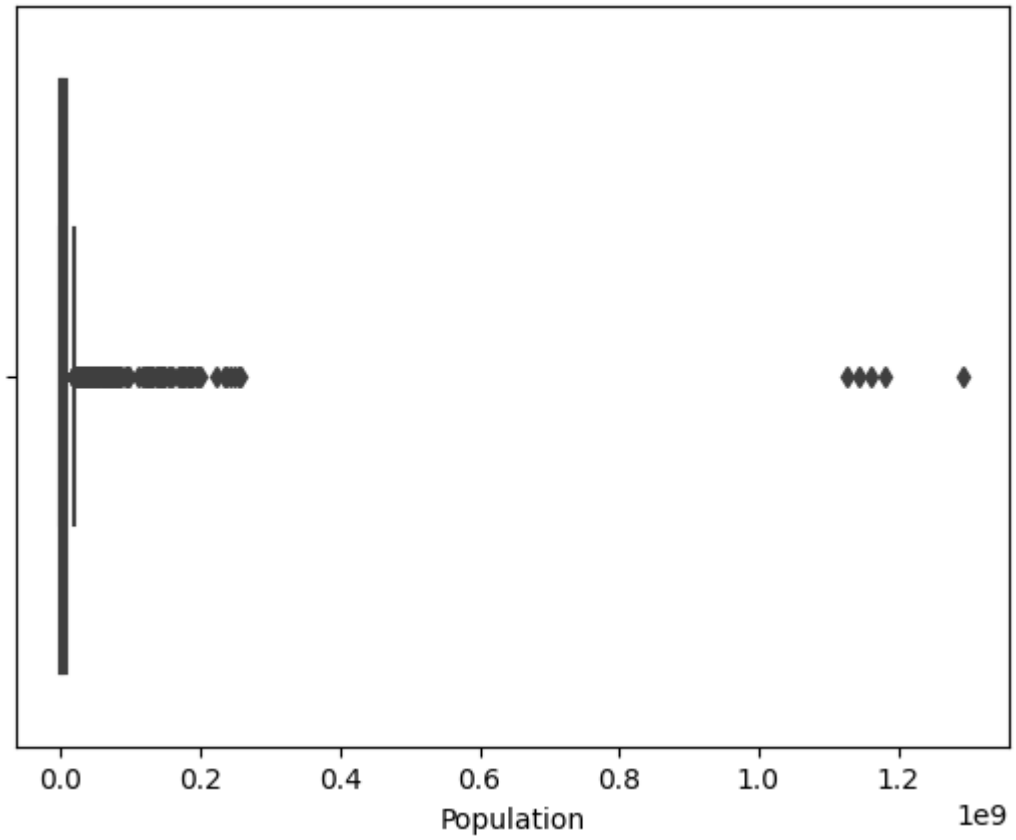

0
20
40
60
80
100
Polio

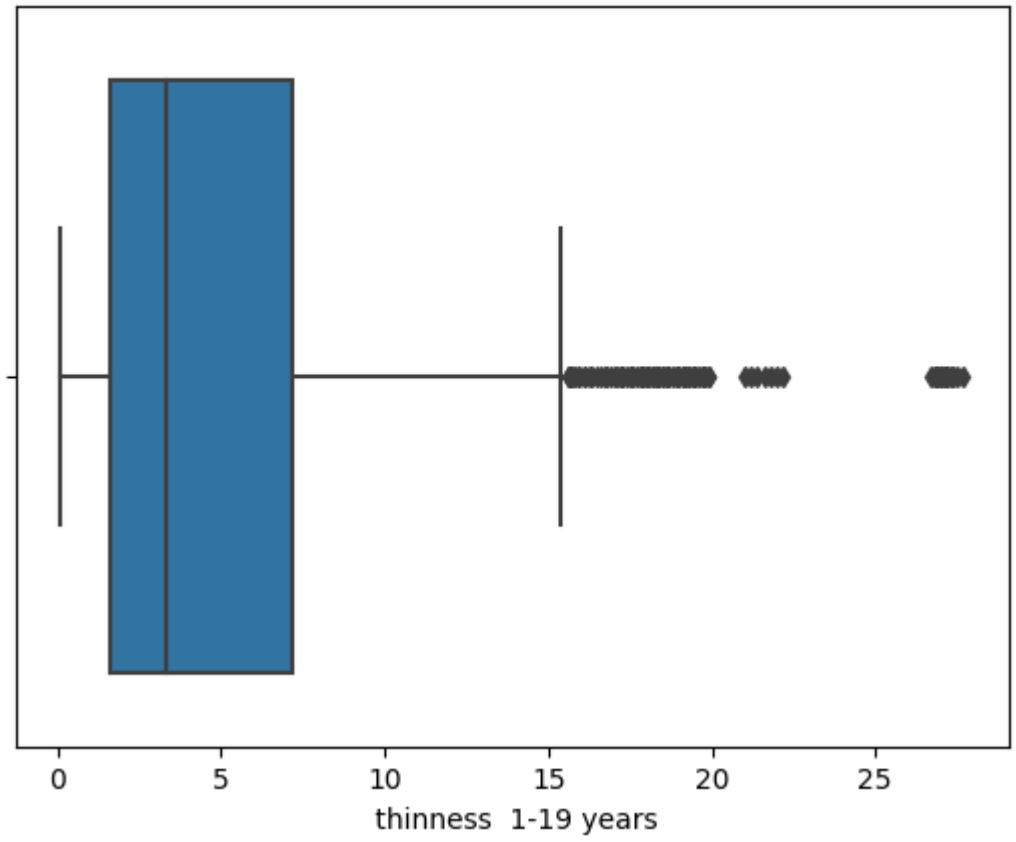

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Total expenditure

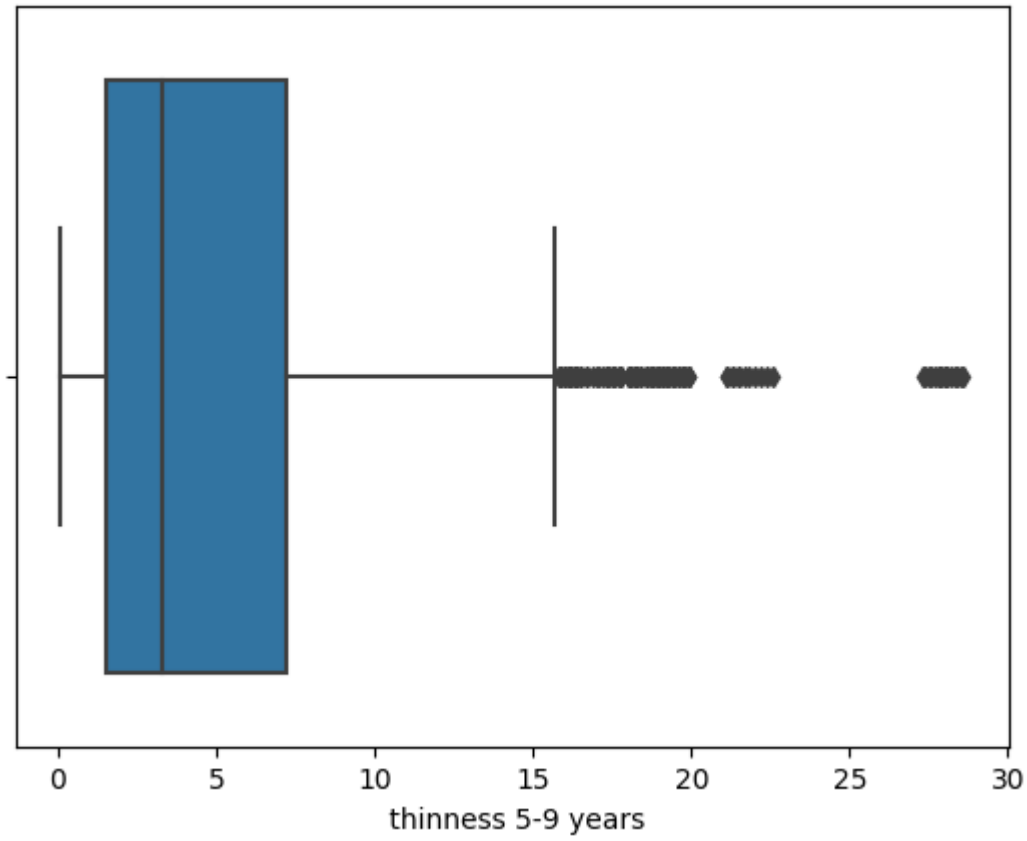

0
20
40
60
80
100
Diphtheria

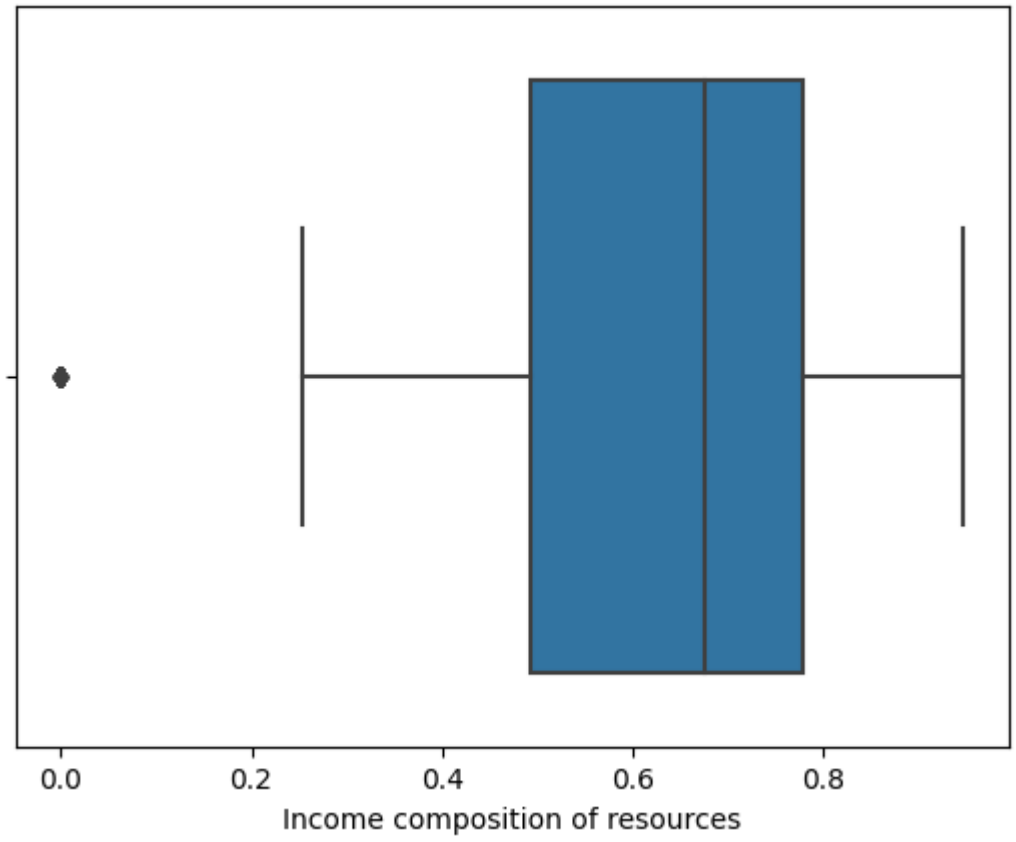

0
10
20
30
40
50
HIV/AIDS

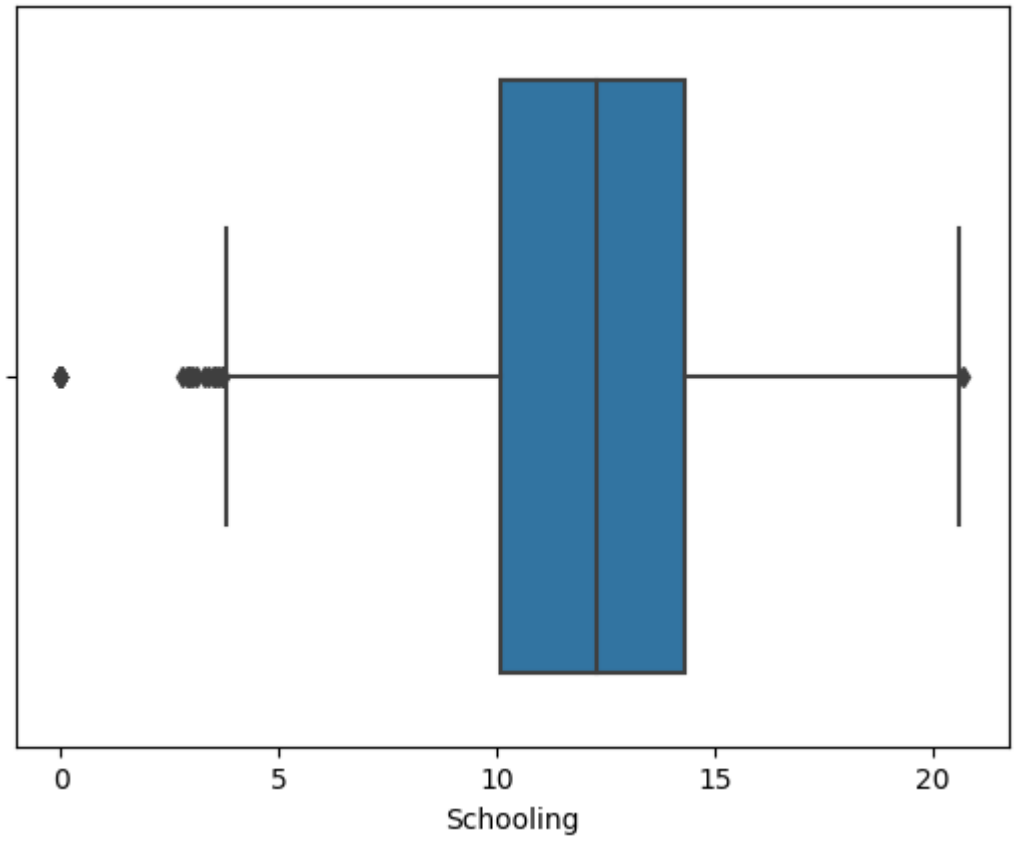

0
20000
40000
60000
80000
100000
120000
GDP


0.0
0.2
0.4
0.6
0.8
1.0
1.2
Population
1e9


0
5
10
15
20
25
thinness 1-19 years


0
5
10
15
20
25
30
thinness 5-9 years


0.0
0.2
0.4
0.6
0.8
Income composition of resources



In [6]:
```
# TOP 10 COUNTRIES WITH MOST LIFE EXPECTANCY
df.groupby("Country").agg({
    "Life expectancy ":"mean"
}).reset_index().sort_values("Life expectancy ", ascending = False).head(10)
```

Out[6]:

	Country	Life expectancy
84	Japan	82.53750
165	Sweden	82.51875
75	Iceland	82.44375
166	Switzerland	82.33125
60	France	82.21875
82	Italy	82.18750
160	Spain	82.06875
7	Australia	81.81250
125	Norway	81.79375
30	Canada	81.68750

In [7]:
```
# TOP 10 COUNTRIES WITH LEAST LIFE EXPECTANCY
df.groupby("Country").agg({
    "Life expectancy ":"mean"
}).reset_index().sort_values("Life expectancy ", ascending = True).head(10)
```

Out[7]:

	Country	Life expectancy
152	Sierra Leone	46.11250
31	Central African Republic	48.51250
94	Lesotho	48.78125
3	Angola	49.01875
100	Malawi	49.89375
32	Chad	50.38750
44	Côte d'Ivoire	50.38750
192	Zimbabwe	50.48750
164	Swaziland	51.32500
123	Nigeria	51.35625

In [8]:
```
y= df["Life expectancy "]
X=  df.drop(["Life expectancy "], axis=1)
y.fillna(y.median(), inplace=True)
X.fillna(X.mean(), inplace=True)

X.drop([ 'Status', 'Population'], axis=1, inplace= True)
X.Year = pd.to_datetime(X.Year).dt.year

import category_encoders as ce
bin_enc = ce.BinaryEncoder(drop_invariant=True)
X = bin_enc.fit_transform(X)

from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
X = sc.fit_transform(X)
```

In [9]:
```
X_train, X_test, y_train, y_test= train_test_split(X, y, test_size= 0.30, random_st
```

In [10]:
```
from sklearn import preprocessing
from sklearn import utils

lab = preprocessing.LabelEncoder()
y_train = lab.fit_transform(y_train)
```

In [11]:
```
clf.fit(X = X_train, y = y_train)
```

Out[11]:
```
▾ LogisticRegression
LogisticRegression()
```

In [12]:
```
y_test = lab.fit_transform(y_test)
```

In [13]:
```
pred = clf.predict(X_test)
pred
```

Out[13]:
```
array([216, 108, 246, 183, 258, 211, 190, 349, 282, 258, 293, 220, 290,
       193, 313,  68, 169, 179, 259, 237, 217, 303, 263, 244, 133, 244,
       323, 272, 343, 201, 323, 282,   9, 312, 226, 163, 283,  85, 269,
       272, 152, 252, 330, 350, 238, 131, 177, 302, 255, 259, 157, 220,
       302, 266, 328, 100, 261, 343, 203,  92, 261, 242, 278, 238, 126,
       310, 115, 271,   5, 135, 108,  93, 219, 272, 279, 235, 314,  47,
       126, 253, 205,  43, 241, 139, 281,  93, 243, 328, 135, 216, 185,
       257, 154, 264, 262, 158, 122, 258, 197,  90, 252, 328, 248, 269,
       109, 117, 328,  74, 137, 276, 349, 263, 159, 326,  92, 282, 261,
       189, 269, 258, 200, 318,  97, 116, 185, 212, 259, 258, 216, 219,
       183, 241, 151, 344, 339, 324, 185, 261, 324, 242, 155, 349, 298,
       282, 295, 281, 129, 252, 279, 252, 297, 302,  71, 291, 169, 182,
        90,  62,  10, 189, 294, 121, 269, 261, 135, 302,  90, 324, 268,
       100, 204, 238, 135, 242, 221, 263, 100, 100, 282, 210, 338, 219,
       327, 133, 124, 201, 326, 281,  50, 256, 135, 238, 282, 112, 255,
       121, 164, 151,  26, 254, 292, 181,  93, 238, 181, 131, 207, 258,
       255, 145, 253,  76, 208, 290, 258, 323, 220, 250, 230, 272, 259,
       237, 343, 211, 311, 261, 313, 272, 237, 151, 272, 169, 264,  21,
       350, 297, 263, 269, 237, 176, 269, 351, 106, 115,  62, 272,  56,
       263, 353, 137, 312, 283, 269,  74, 126, 218, 272, 292, 317, 267,
       246, 174, 168,  58, 146, 229, 263, 252, 251, 230, 242, 259, 264,
       258, 181, 324, 189,  88, 204, 243, 326,  93, 249, 263, 308, 302,
        58, 151, 252, 302,  62, 208, 133, 197, 211, 212, 302, 191, 264,
       351, 249, 208, 116, 302, 158, 257, 269, 230, 270, 258, 267, 288,
       281,  62, 298, 254, 321, 277, 276, 311, 246,  79, 257, 235, 202,
       264, 309, 263, 274, 309,  66, 124, 252, 302, 259, 261, 269, 164,
        66, 272, 312, 197, 228, 302, 324, 145, 219, 237,  66,  80, 259,
       161, 330, 258, 258, 267, 282, 219, 349, 250, 259, 238, 211,  60,
       176, 216, 310, 291, 211, 171,  59, 252, 174, 193, 241, 261, 181,
       218, 272, 258, 275, 324, 176, 291, 258, 277, 276, 242,  71, 181,
       159, 272, 240, 299, 253, 290, 228, 211, 258, 133, 253, 287, 275,
       323, 330, 175, 252, 258, 248, 243, 258, 276, 269, 272, 263, 158,
        85, 351, 152, 309, 287, 252, 324, 269, 311, 312, 279,   9, 263,
       272, 324,  15,  33, 237, 276, 310, 263,  60,  90,  72, 310,  80,
       259,  72, 252, 124,  74, 309,  94, 272, 216, 270, 193, 263, 303,
        79, 255, 252, 281, 271, 323, 269, 264, 259, 302,  66, 255, 153,
       204, 311, 137, 212, 244, 326, 302, 263, 269, 109, 258, 185, 302,
       269, 309, 279,  91,  31,  38, 351, 248, 263, 252, 354, 131, 265,
        48, 354, 156, 269, 116, 189, 309, 135, 157, 167, 279, 256, 263,
       249, 243, 166, 252, 318, 190, 267, 313, 237,  49, 271, 258, 272,
       326, 215, 309, 263, 176, 152, 145, 263, 269, 282, 160, 259, 252,
       279,  68, 211, 261, 211, 263,  51, 176, 133, 220, 262, 237,  93,
       223, 261, 242, 219,  73,  75, 263, 295, 303, 269, 189, 134, 219,
       115, 267, 293, 274, 343, 184, 130,  81, 261, 303, 139, 242, 126,
       115, 182, 266, 237, 322, 302, 120, 326, 197, 151, 211, 302, 252,
       261,  51, 311,  93, 243, 212,  90,  46, 258, 269, 321, 302, 241,
       193, 139, 304, 323, 100,  72, 243, 253,  36, 163,  33, 204, 191,
        38, 218,  91, 248, 151, 231, 108, 272, 350, 238,  62, 261, 119,
       303,  66, 324, 237,  73, 208, 100, 271, 135, 219, 165, 213, 197,
       271, 244, 309, 275, 177, 211, 299, 258, 267, 349, 252,  66, 171,
       260, 260, 311,  60, 287, 272, 182, 332,  93, 123, 258, 126, 248,
       252, 201, 275,  16, 271, 263, 151, 152, 289,  93, 250, 267, 228,
       295, 271, 193, 352, 209, 324, 282, 277, 271, 209, 354, 312, 159,
       300, 111,  71,  68, 130, 281, 145, 269, 282, 259, 117, 122, 123,
       190,  42, 269, 263, 191,   9, 311, 302, 217, 163, 158, 258, 237,
       131, 302, 258, 242, 108, 267,  62, 328, 259, 326, 137, 224, 248,
```

```
253,  56, 160, 219, 260, 331, 176, 236, 302, 152, 266, 133, 116,
228, 181, 324, 163, 126,  76, 274, 324, 230, 218, 120,  50, 255,
324, 211, 270, 235, 271, 242, 351, 155,  93, 311, 235, 237,  90,
295, 139, 324, 229, 100, 272, 243, 325, 199, 158, 137, 282, 203,
130, 103,  58, 328, 241, 311, 145, 302, 282, 181, 258, 189, 264,
269, 328, 276, 267, 176, 241, 229, 218, 100, 161, 299, 120, 237,
 60,  85, 241, 253, 308, 263, 242, 302,  48, 207,  15, 324, 199,
211,  93, 274, 249, 259, 287, 354,  43, 129, 201, 269, 237, 266,
 30, 324, 311, 349, 262,   7, 314, 323, 219, 145, 120, 305, 242,
131, 260,  66, 176, 323, 217, 302, 238, 224, 149, 292,   6, 252,
100, 258, 330, 254, 237, 309, 269, 263, 133, 168, 263, 277, 158,
267, 160, 258, 317, 111, 321, 176, 324, 242, 264,  91], dtype=int64)
```

In [14]:
```
np.array(y_test)
```

Out[14]:
```
array([156,  48, 178, 143, 221,  68, 146, 279, 189, 208, 277, 188, 251,
       149, 270,  47, 118, 141, 213, 144, 187, 268, 198, 204,  74, 151,
       297, 196, 263, 168, 300, 219,   2, 297, 196, 127, 211,  35, 237,
       204, 120, 227, 299, 275, 128,  93,  87, 242, 208, 243, 157, 174,
       300, 248, 252,  34, 203, 281, 161,  51, 197, 182, 228, 241, 137,
       285, 154, 248,   5, 119,  90,   9, 149, 224, 234, 195, 262,  17,
        67, 202, 181,  24, 199,  95, 243, 139, 210, 256, 155, 173, 214,
       204,  86, 256, 164, 118,  72, 213, 147,  59, 215, 292, 203, 276,
        60,  79, 297,  42,  71, 240, 285, 282, 216, 297,  69, 210, 188,
        99, 226, 222, 107, 285,  62,  96, 227, 192, 208, 221, 172, 166,
       151, 189, 103, 287, 285, 291, 224, 201, 280, 172, 197, 287, 208,
       206, 213, 238, 115, 150, 219, 232, 293, 262,  77, 240, 102, 209,
        76,  50,  11, 102, 262,  66, 230, 223, 119, 216,  71, 289, 127,
        89, 104, 194,  14, 211, 151, 228,  32,  92, 223, 116, 223, 179,
       283,  56,  84, 123, 278, 232,  42, 183, 123,  28, 203, 104, 218,
        59, 102, 145,  15, 250, 199,  97,  46, 190, 185, 104, 172, 205,
       210, 151, 213,  55, 200, 242, 216, 256, 179, 265, 232, 293, 229,
       214, 269, 117, 253, 221, 269, 183, 135, 117, 212, 156, 215,   6,
       272, 256, 269, 266, 224, 149, 228, 278,  64,  99,  66, 228,  35,
       231, 273, 158, 206, 238, 253,  18,  98, 134, 208, 194, 285, 193,
       209, 112,  37,  25, 161, 231, 207, 237, 211,  63, 212, 201, 147,
       207, 145, 293, 246, 170, 142, 205, 290,  38, 206, 208, 251, 252,
        22, 130, 188, 235,  39, 208, 106, 160, 123, 203, 256, 205, 209,
       296, 195, 198, 100, 263, 127, 210, 185, 204, 233, 204, 256, 154,
       261,  50, 195, 259, 299, 215, 174, 245, 227,  54, 209, 228, 164,
       224, 251, 214, 237, 254,  38,  86, 223, 267, 223, 217, 240,  91,
       158, 166, 188, 159, 162, 293, 282, 112, 168, 220, 222,  87, 239,
       150, 293, 205, 199, 233, 197, 172, 274, 180, 221, 198, 140,  19,
       161, 181, 264, 265, 138, 208,  60, 201, 163, 121, 237, 175, 181,
       180, 188, 207, 296, 242, 129, 277, 199, 207, 176, 265,  39, 135,
       127, 172, 195, 265, 256, 246, 213, 133, 208, 133, 106, 222, 142,
       244, 299, 156,  63, 184, 231, 216, 217, 225, 220, 196, 187, 119,
        38, 202, 120, 299, 245, 153, 281, 195, 260, 204, 214,   1, 233,
       211, 284,  28,  36, 114, 178, 260, 169,  87,  74,  30, 267,  82,
       225,  40, 190,  76,  13, 244,  71, 234, 171, 222, 154, 203, 256,
        49, 205, 215, 252, 223, 280, 258, 236, 219, 264, 218, 204,  78,
       159, 270, 110, 132, 256, 273, 298, 226, 239,  98, 217,  83, 238,
       232, 275, 288,  59,  27,  29, 276, 168, 238, 214, 257,  71, 228,
        41, 285, 139, 237, 108, 144, 244, 109,  16, 136, 239, 256, 207,
       261, 171, 121, 182, 268, 208, 208, 289, 219,  48, 245, 220, 188,
       295, 165, 269, 248, 125,   7, 124, 208, 219, 192, 157, 211, 230,
       236,  63,  81, 218, 170, 268,  35, 121, 171, 165, 236, 206,  41,
       198, 223, 177, 256,  62, 101, 216, 260, 262, 219, 186, 132, 213,
       120, 234, 297, 244, 270, 172,  88,  84, 209, 257,  93, 237, 145,
       105, 161, 237, 207, 272, 255,  88, 275, 152,  94, 179, 274, 228,
       208,  23, 262,  98, 217, 170,  49,  73, 226, 222, 276, 292, 226,
       156,  88, 256, 282,  43,  21, 166, 196,  15, 129,  20, 184, 192,
        34, 135,  67, 200,  92, 238,  62, 186, 276, 216,  51, 214,  47,
       234, 224, 263, 211,  31,  96,  28, 225, 115, 170, 225, 166, 181,
       222, 200, 258, 206,  96, 103, 261, 137, 194, 283, 227,  45, 177,
       111, 218, 268,  42, 203, 214, 197, 300,  92, 102, 208, 119, 192,
       239, 194, 223,   8, 221, 227, 113, 218, 227, 175, 181, 210, 214,
       254, 209, 148, 268, 169, 283, 225, 212, 209, 172, 259, 200, 177,
       285, 177,  86,  44, 122, 234, 123, 222, 190, 216,  75, 122,  99,
        77,  26, 215, 258, 142,   3, 257, 285, 168, 163, 124, 165, 217,
        93, 238, 204, 180,  57, 236,  53, 290, 216, 285,  38, 194, 210,
```

```
192,  33, 136, 175, 222, 277, 119, 191, 190, 141, 244,  85,  86,
240, 170, 296, 134,  81,  52, 232, 280,  72, 228,  12,  18, 213,
294,  76, 293, 192, 231, 192, 286, 165,  91, 267, 202, 214,  58,
253,  81, 274, 157,  80, 216, 208, 271, 158, 101,  78, 227, 163,
 70,  61,  15, 275, 228, 270, 123, 249, 239, 162, 206,  94, 226,
126, 276, 262, 178, 166, 209, 131, 143,  54, 139, 249,  29, 165,
 55,  87, 202, 270, 261, 195, 205, 277,  45, 167,  10, 299, 161,
106, 102, 185, 217, 216, 224, 298,  16, 142, 180, 229, 156, 238,
  4, 279, 266, 283, 228,  17, 267, 298, 157, 115,   0, 228, 200,
 65, 232,  33, 133, 258, 182, 206, 126, 136, 103, 196,  10, 203,
 87, 195, 285, 212, 205, 246, 247, 264,  93,  46, 220, 227, 127,
203, 154, 207, 276,  76, 297, 147, 240, 174, 199,  74], dtype=int64)
```

In [15]:
```
clf.score(X_train, y_train)
```

Out[15]:
```
0.23054474708171208
```

In [16]:
```
clf.score(X_test, y_test)
```

Out[16]:
```
0.009070294784580499
```