In [10]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly as px

from scipy.stats import ttest_ind
```

In [3]:
```
!pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\hp\anaconda3\lib\site-packages (5.
14.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\hp\anaconda3\lib\site-pac
kages (from plotly) (8.2.2)
Requirement already satisfied: packaging in c:\users\hp\anaconda3\lib\site-packages
(from plotly) (21.0)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\hp\anaconda3\lib\site-pa
ckages (from packaging->plotly) (3.0.4)
```

In [4]:
```
df = pd.read_csv("Life Expectancy Data.csv")
```

In [5]:
```
df.head()
```

Out[5]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	M
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	

5 rows × 22 columns

In [6]:
```
df.shape
```

Out[6]:
```
(2938, 22)
```

In [7]:
```
df.dtypes.to_frame('Dataframes of attributes')
```

Out[7]:

	Dataframes of attributes
Country	object
Year	int64
Status	object
Life expectancy	float64
Adult Mortality	float64

Dataframes of attributes	
infant deaths	int64
Alcohol	float64
percentage expenditure	float64
Hepatitis B	float64
Measles	int64
BMI	float64
under-five deaths	int64
Polio	float64
Total expenditure	float64
Diphtheria	float64
HIV/AIDS	float64
GDP	float64
Population	float64
thinness 1-19 years	float64
thinness 5-9 years	float64
Income composition of resources	float64
Schooling	float64

In [8]:
```
df.columns
```

Out[8]:
```
Index(['Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortality',
       'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
       'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total expenditure',
       'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
       ' thinness  1-19 years', ' thinness 5-9 years',
       'Income composition of resources', 'Schooling'],
      dtype='object')
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   Country                          2938 non-null   object 
 1   Year                             2938 non-null   int64  
 2   Status                           2938 non-null   object 
 3   Life expectancy                  2928 non-null   float64
 4   Adult Mortality                  2928 non-null   float64
 5   infant deaths                    2938 non-null   int64  
 6   Alcohol                          2744 non-null   float64
 7   percentage expenditure           2938 non-null   float64
 8   Hepatitis B                      2385 non-null   float64
 9   Measles                          2938 non-null   int64  
 10   BMI                             2904 non-null   float64
 11  under-five deaths                2938 non-null   int64  
 12  Polio                            2919 non-null   float64
```

```
 13  Total expenditure                2712 non-null   float64
 14  Diphtheria                       2919 non-null   float64
 15   HIV/AIDS                        2938 non-null   float64
 16  GDP                              2490 non-null   float64
 17  Population                       2286 non-null   float64
 18   thinness  1-19 years            2904 non-null   float64
 19   thinness 5-9 years             2904 non-null   float64
 20  Income composition of resources  2771 non-null   float64
 21  Schooling                        2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Country                            0
Year                               0
Status                             0
Life expectancy                   10
Adult Mortality                   10
infant deaths                      0
Alcohol                          194
percentage expenditure             0
Hepatitis B                      553
Measles                            0
 BMI                              34
under-five deaths                  0
Polio                             19
Total expenditure                226
Diphtheria                        19
 HIV/AIDS                          0
GDP                              448
Population                       652
 thinness  1-19 years             34
 thinness 5-9 years               34
Income composition of resources  167
Schooling                        163
dtype: int64
```

In [14]:
```
from sklearn.impute import SimpleImputer
imputer=SimpleImputer(missing_values=np.nan,strategy='mean',fill_value=None)
df['Life expectancy ']=imputer.fit_transform(df[['Life expectancy ']])
df['Adult Mortality']=imputer.fit_transform(df[['Adult Mortality']])
df['Alcohol']=imputer.fit_transform(df[['Alcohol']])
df['Hepatitis B']=imputer.fit_transform(df[['Hepatitis B']])
df[' BMI ']=imputer.fit_transform(df[[' BMI ']])
df['Polio']=imputer.fit_transform(df[['Polio']])
df['Total expenditure']=imputer.fit_transform(df[['Total expenditure']])
df['Diphtheria ']=imputer.fit_transform(df[['Diphtheria ']])
df['GDP']=imputer.fit_transform(df[['GDP']])
df['Population']=imputer.fit_transform(df[['Population']])
df[' thinness  1-19 years']=imputer.fit_transform(df[[' thinness  1-19 years']])
df[' thinness 5-9 years']=imputer.fit_transform(df[[' thinness 5-9 years']])
df['Income composition of resources']=imputer.fit_transform(df[['Income composition
df['Schooling']=imputer.fit_transform(df[['Schooling']])
```

In [15]:
```
df.isnull().sum()
```

Out[15]:
```
Country                          0
Year                             0
Status                           0
Life expectancy                  0
```

```
Adult Mortality                  0
infant deaths                    0
Alcohol                          0
percentage expenditure           0
Hepatitis B                      0
Measles                          0
 BMI                             0
under-five deaths                0
Polio                            0
Total expenditure                0
Diphtheria                       0
 HIV/AIDS                        0
GDP                              0
Population                       0
 thinness  1-19 years            0
 thinness 5-9 years              0
Income composition of resources  0
Schooling                        0
dtype: int64
```

In [14]:
```
df.describe()
```

Out[14]:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.940461
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070016
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.000000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.000000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.000000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000

In [8]:
```
categorical= df.select_dtypes(include= "O")
numerical= df.select_dtypes(exclude= "O")
```

In [9]:
```
categorical.describe()
```

Out[9]:

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

In [10]:
```
numerical.describe()
```

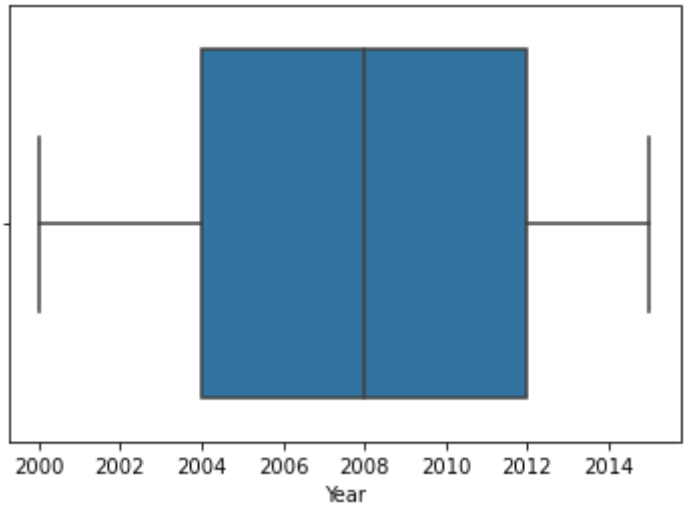
Out[10]:

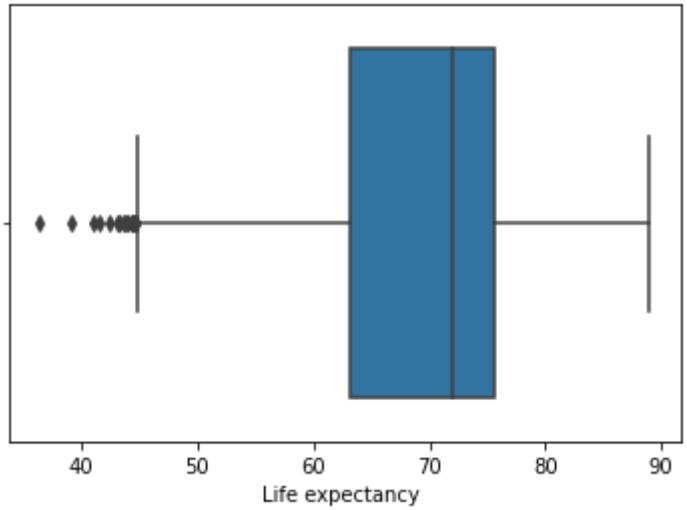
	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.940461
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070016
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.000000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.000000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.000000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000

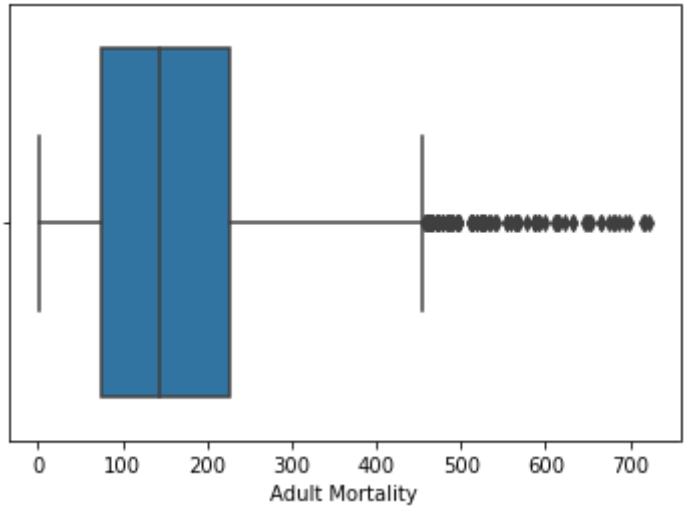
Univariate Analysis

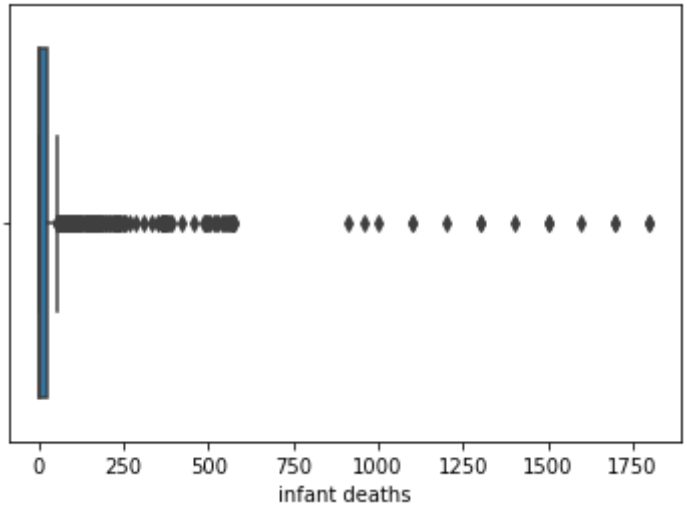
In [24]:

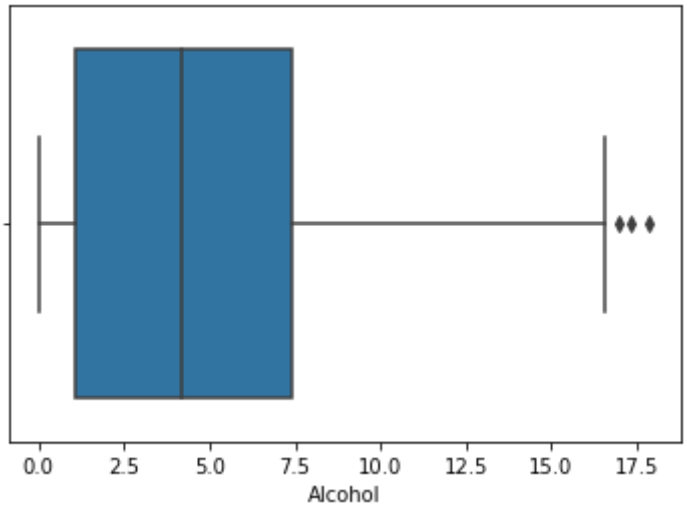
```
for feature in numerical.columns:
    sns.boxplot(x=numerical[feature])
    plt.show()
```

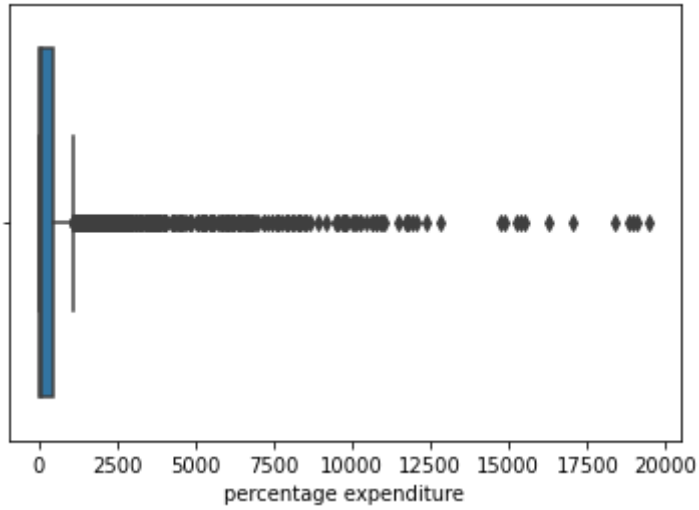


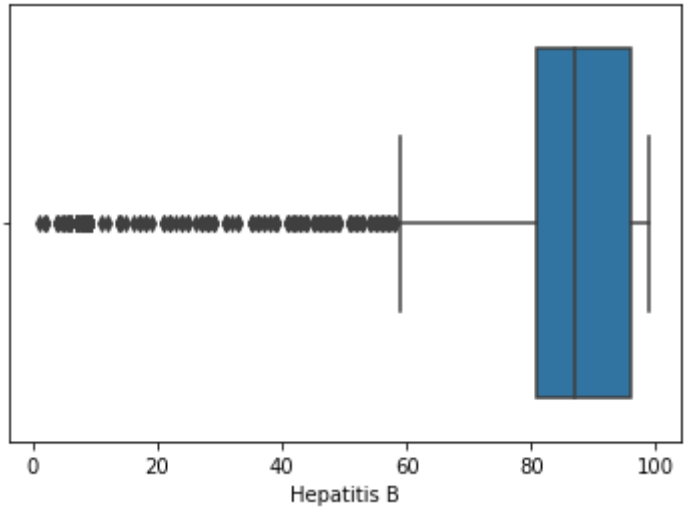


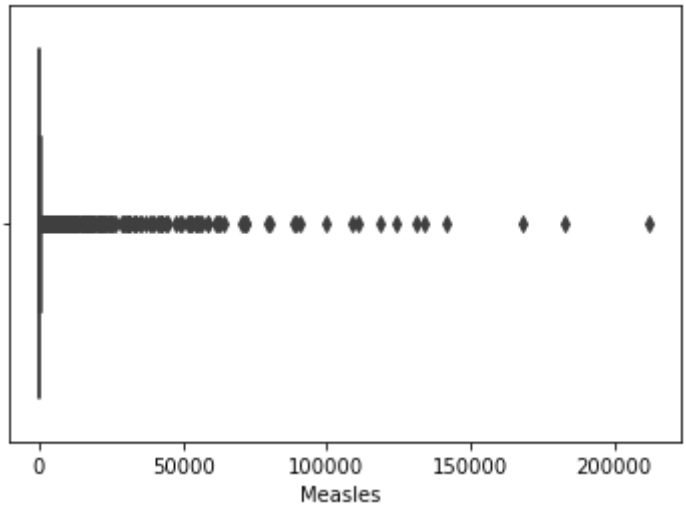

0
100
200
300
400
500
600
700
Adult Mortality

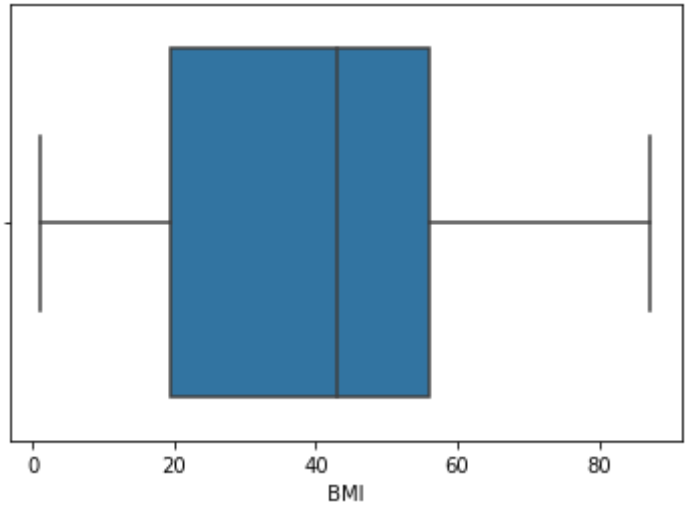

0
250
500
750
1000
1250
1500
1750
infant deaths

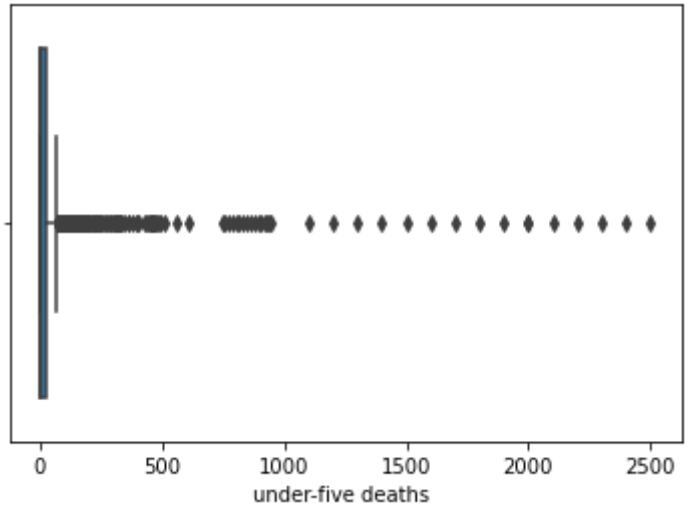

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Alcohol

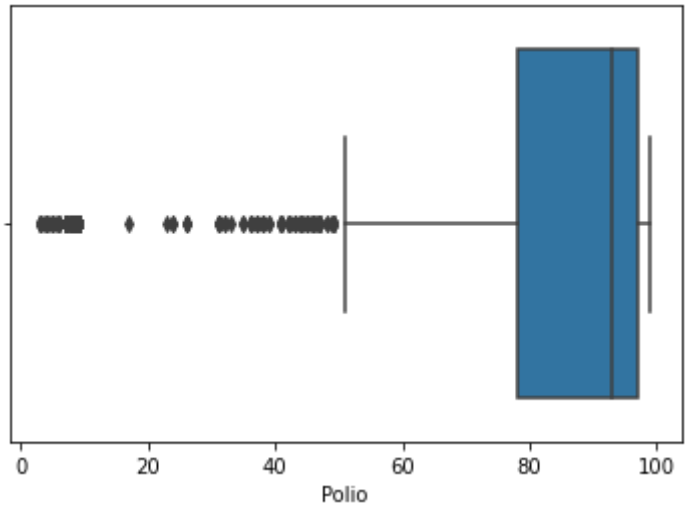

0
2500
5000
7500
10000
12500
15000
17500
20000
percentage expenditure

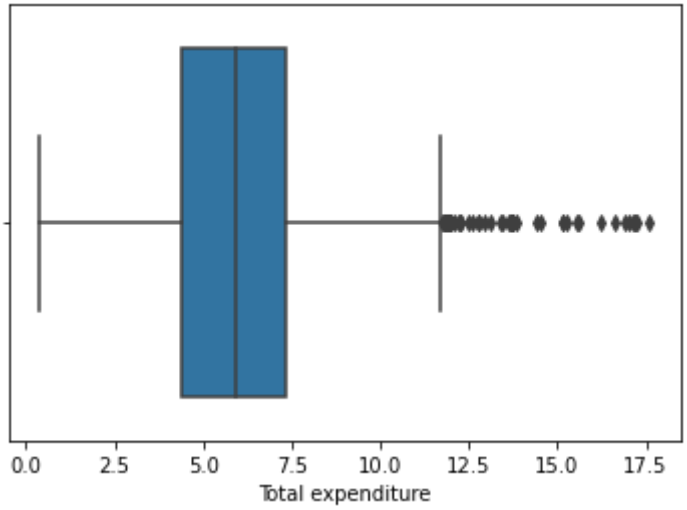

0
20
40
60
80
100
Hepatitis B

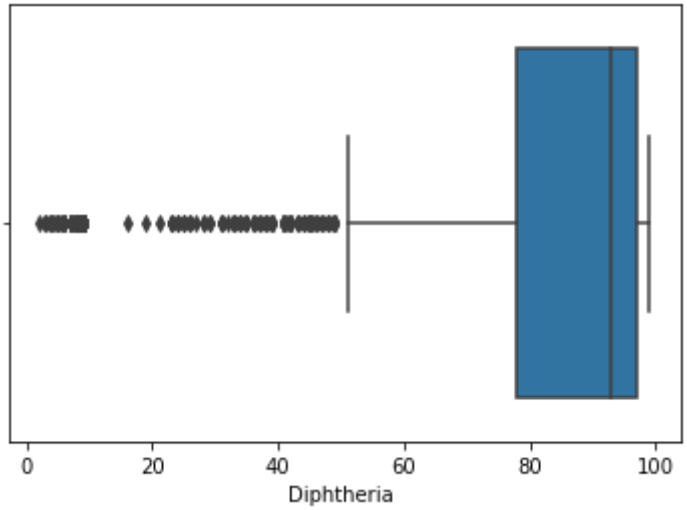

0
50000
100000
150000
200000
Measles

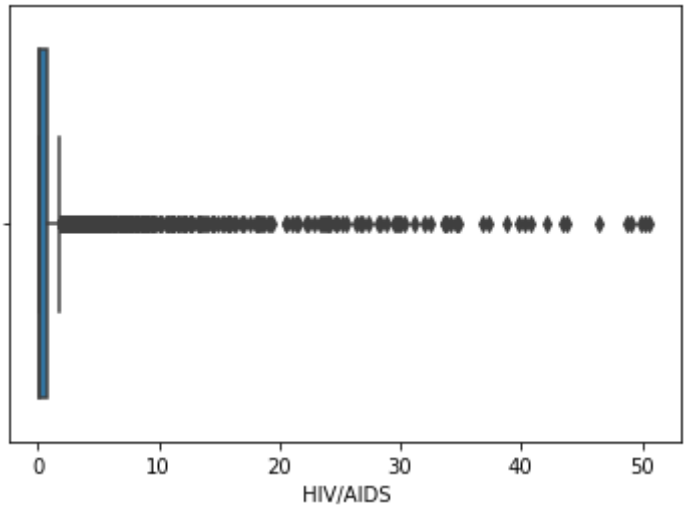

0
20
40
60
80
BMI

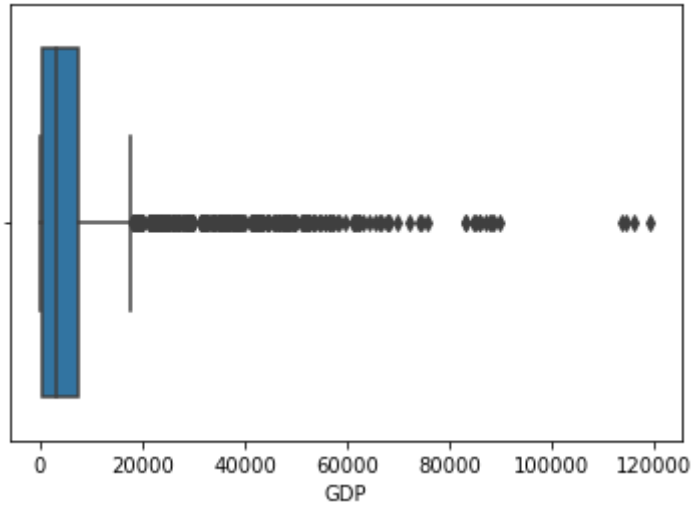

0
500
1000
1500
2000
2500
under-five deaths

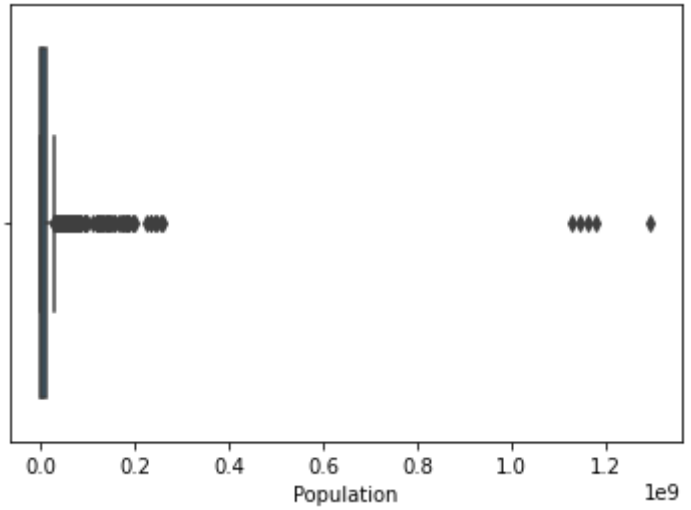

0
20
40
60
80
100
Polio

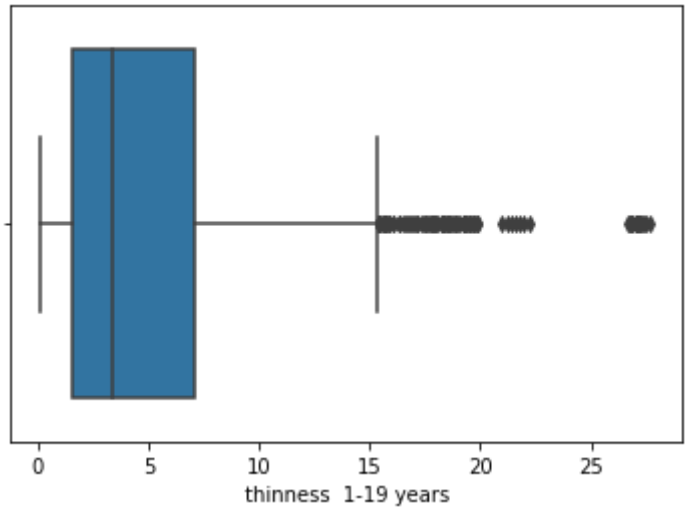

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Total expenditure

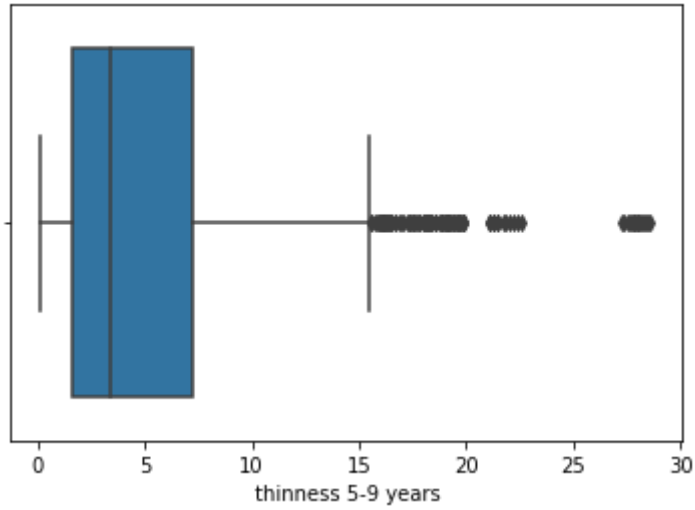

0
20
40
60
80
100
Diphtheria

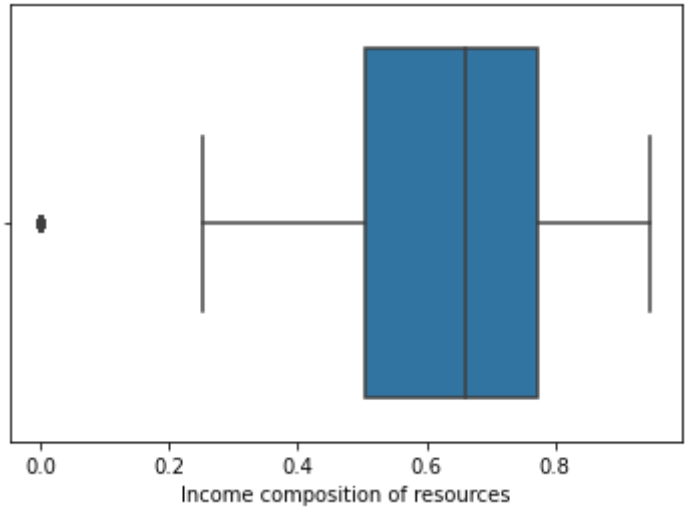

0
10
20
30
40
50
HIV/AIDS

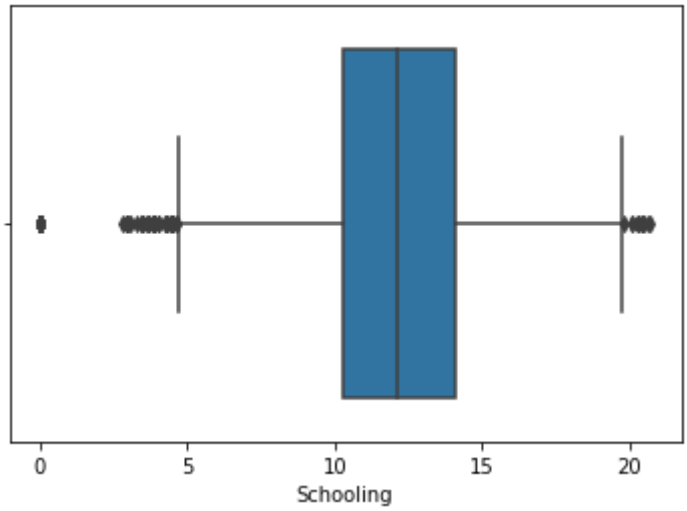

0
20000
40000
60000
80000
100000
120000
GDP


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



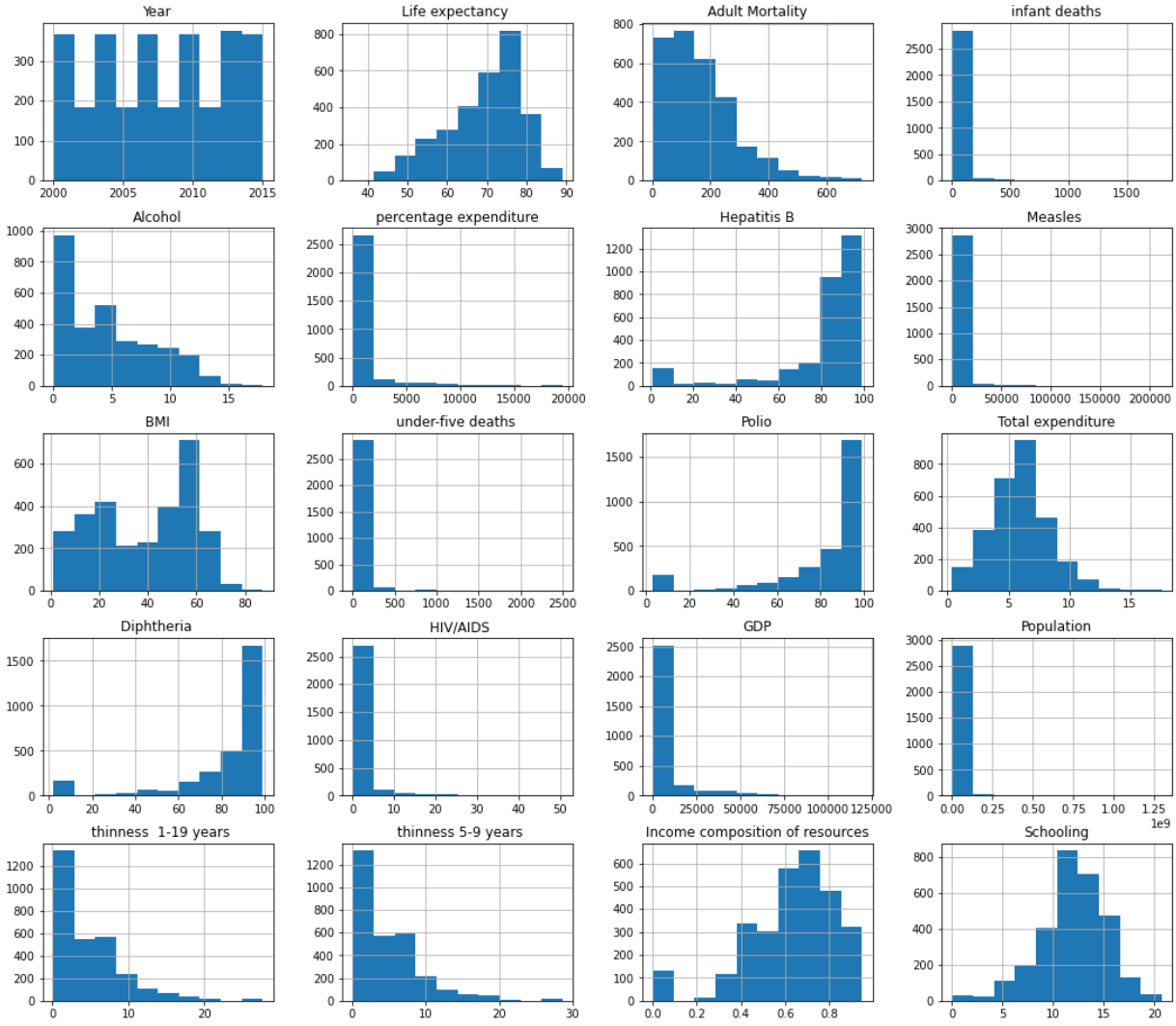




In [33]:
```
df.hist(figsize=(18,16))
```

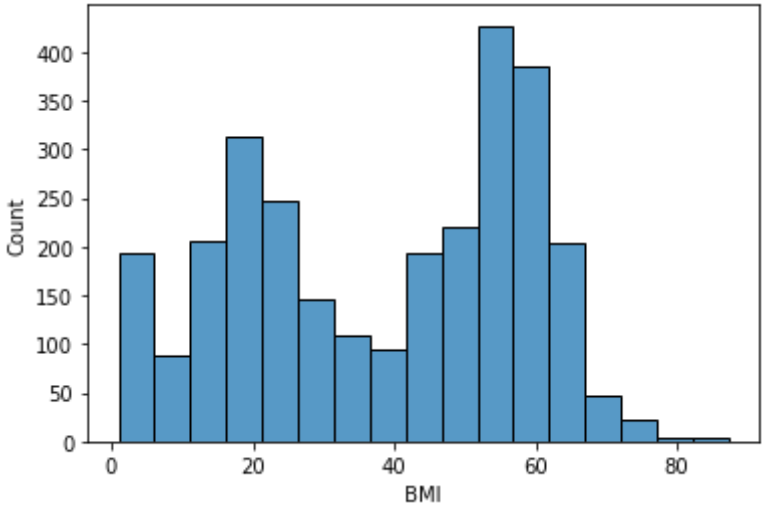
Out[33]:
```
array([[<AxesSubplot:title={'center':'Year'}>,
        <AxesSubplot:title={'center':'Life expectancy '}>,
        <AxesSubplot:title={'center':'Adult Mortality'}>,
        <AxesSubplot:title={'center':'infant deaths'}>],
       [<AxesSubplot:title={'center':'Alcohol'}>,
        <AxesSubplot:title={'center':'percentage expenditure'}>,
        <AxesSubplot:title={'center':'Hepatitis B'}>,
        <AxesSubplot:title={'center':'Measles '}>],
       [<AxesSubplot:title={'center':' BMI '}>,
        <AxesSubplot:title={'center':'under-five deaths '}>,
        <AxesSubplot:title={'center':'Polio'}>,
```

```
 <AxesSubplot:title={'center':'Total expenditure'}>],
[<AxesSubplot:title={'center':'Diphtheria '}>,
 <AxesSubplot:title={'center':' HIV/AIDS'}>,
 <AxesSubplot:title={'center':'GDP'}>,
 <AxesSubplot:title={'center':'Population'}>],
[<AxesSubplot:title={'center':' thinness  1-19 years'}>,
 <AxesSubplot:title={'center':' thinness 5-9 years'}>,
 <AxesSubplot:title={'center':'Income composition of resources'}>,
 <AxesSubplot:title={'center':'Schooling'}>]], dtype=object)
```



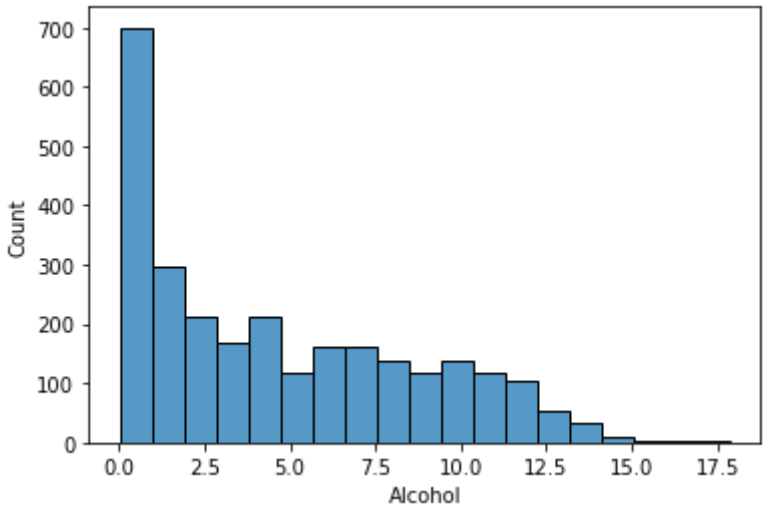
In [16]:
```
sns.histplot(df[' BMI '])
```

Out[16]:
```
<AxesSubplot:xlabel=' BMI ', ylabel='Count'>
```



In [18]:
```
sns.histplot(df['Alcohol'])
```

Out[18]:
```
<AxesSubplot:xlabel='Alcohol', ylabel='Count'>
```



In [35]:
```
df.head(1)
```

Out[35]:

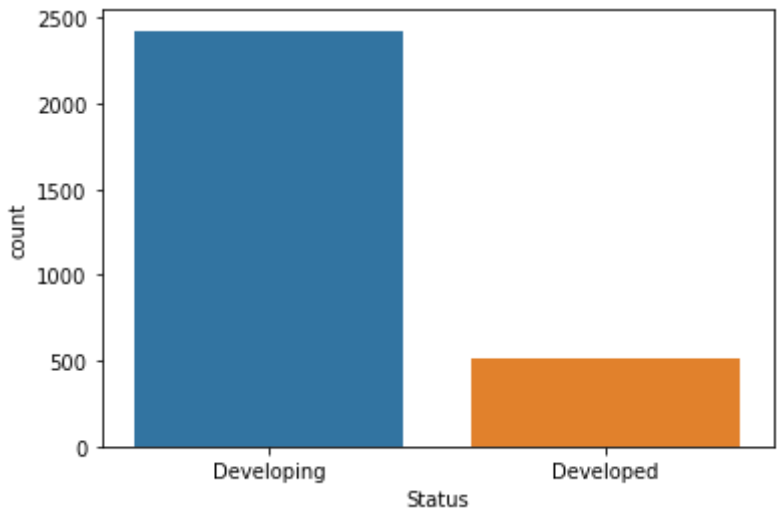
	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	M
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	

1 rows × 22 columns

In [36]:
```
sns.countplot(df['Status'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variable as a keyword arg: x. From version 0.12, the only valid pos
itional argument will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```

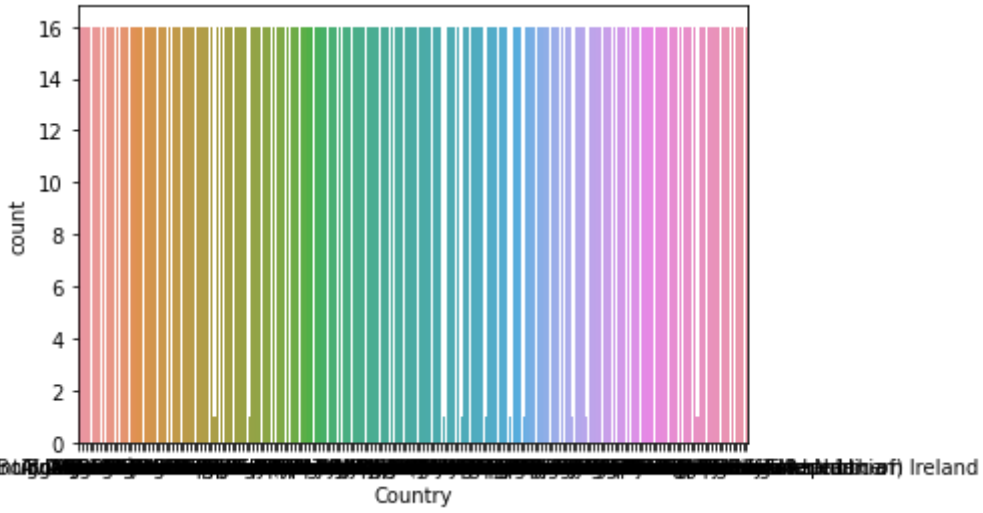
Out[36]:
```
<AxesSubplot:xlabel='Status', ylabel='count'>
```



In [19]:
```
sns.countplot(df['Country'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variable as a keyword arg: x. From version 0.12, the only valid pos
itional argument will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```

Out[19]:
```
<AxesSubplot:xlabel='Country', ylabel='count'>
```



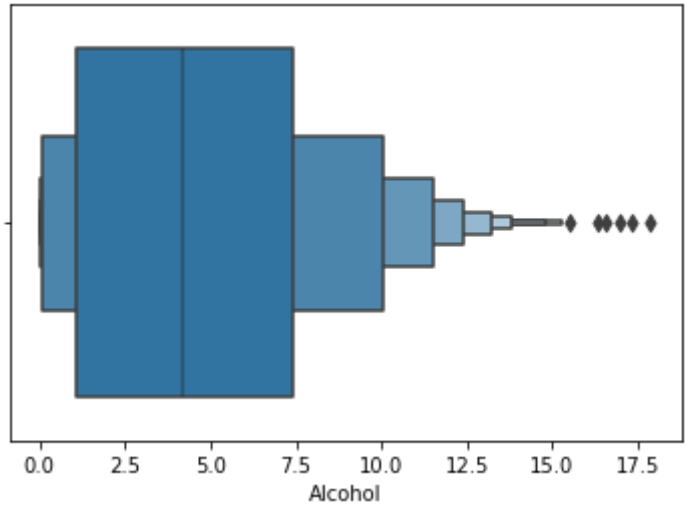
In [38]:
```
df['Country'].value_counts()
```

Out[38]:
```
Afghanistan              16
Peru                     16
Nicaragua                16
Niger                    16
Nigeria                  16
                         ..
Niue                      1
San Marino                1
Nauru                     1
Saint Kitts and Nevis     1
Dominica                  1
Name: Country, Length: 193, dtype: int64
```

In [39]:
```
sns.boxenplot(df['Alcohol'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variable as a keyword arg: x. From version 0.12, the only valid pos
itional argument will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```

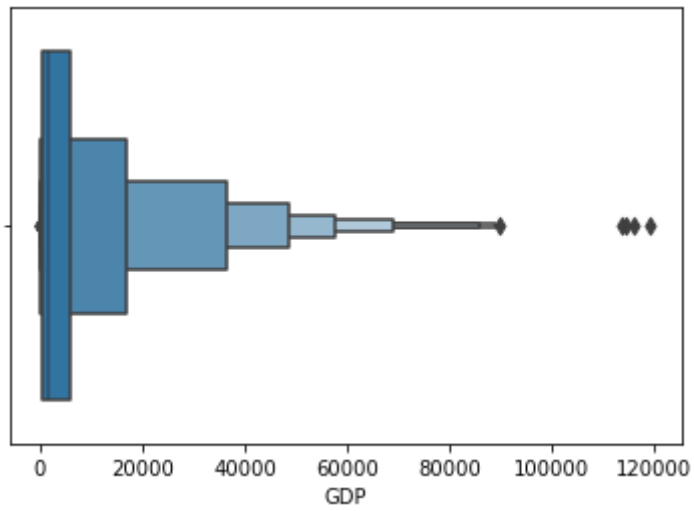
Out[39]:
```
<AxesSubplot:xlabel='Alcohol'>
```



In [11]:
```
sns.boxenplot(df['GDP'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variable as a keyword arg: x. From version 0.12, the only valid pos
itional argument will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```

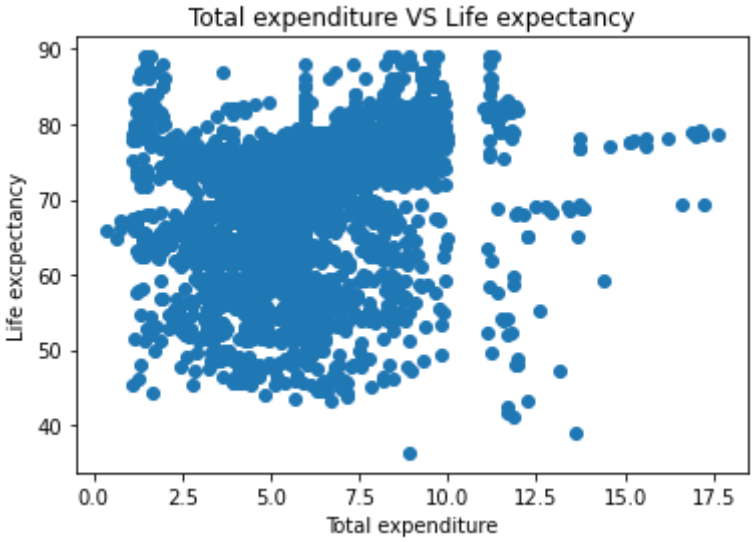
Out[11]:
```
<AxesSubplot:xlabel='GDP'>
```



Bivariant

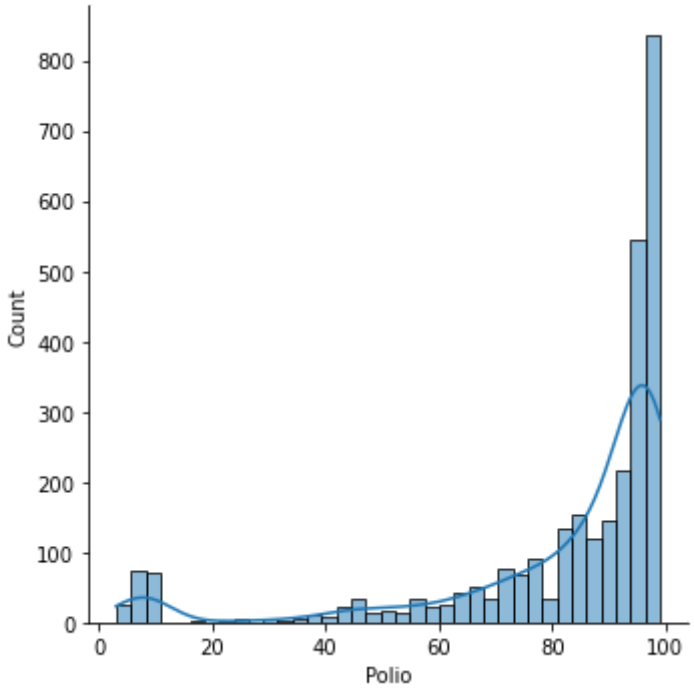
In [25]:
```
plt.scatter(x=df["Total expenditure"], y=df["Life expectancy "])

plt.title("Total expenditure VS Life expectancy");
plt.ylabel("Life excpectancy");
plt.xlabel("Total expenditure");
```



In [26]:
```
sns.displot(x=df["Polio"], kde="True")
```

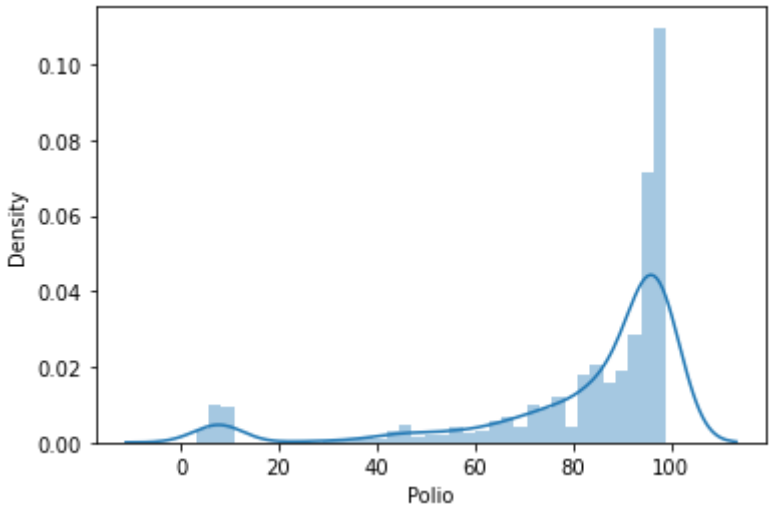
Out[26]:
```
<seaborn.axisgrid.FacetGrid at 0x2ca28ef1130>
```



In [40]:
```
sns.distplot(df['Polio'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarnin
g: `distplot` is a deprecated function and will be removed in a future version. Plea
se adapt your code to use either `displot` (a figure-level function with similar fle
xibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[40]:
```
<AxesSubplot:xlabel='Polio', ylabel='Density'>
```



In [41]:
```
df.head(2)
```

Out[41]:

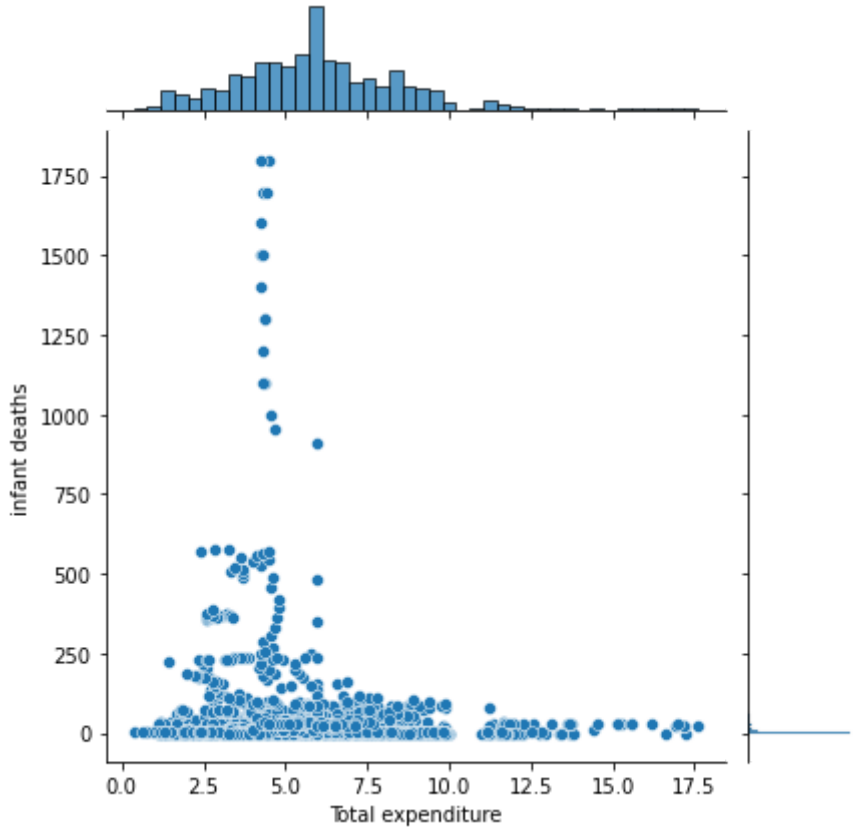
	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	M
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	

2 rows × 22 columns

In [43]:
```
sns.jointplot(df['Total expenditure'],df['infant deaths'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

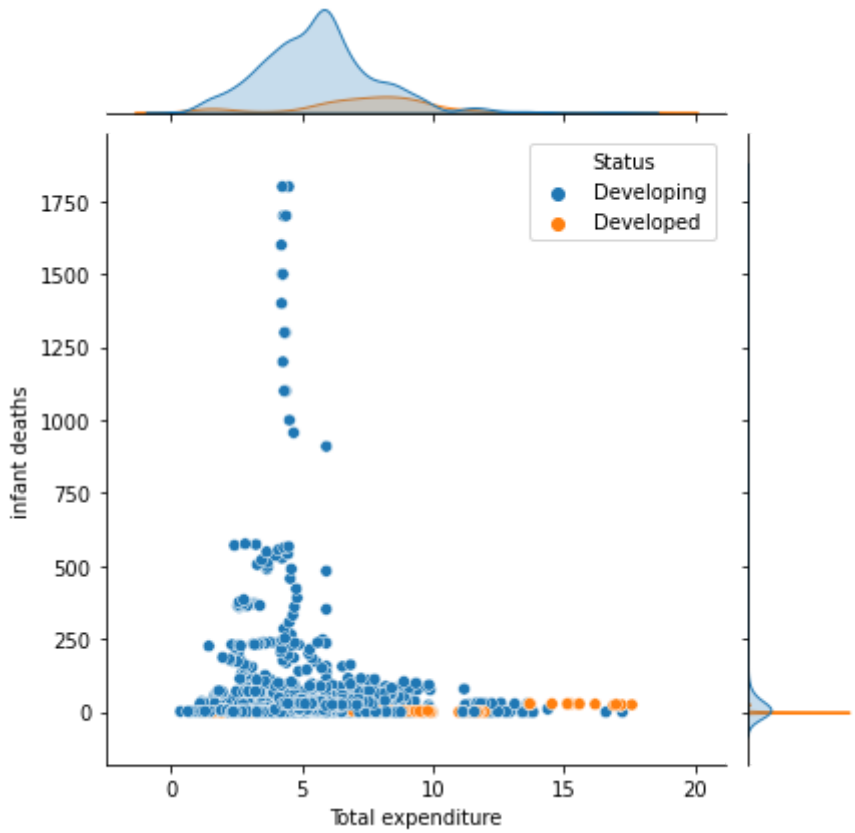
Out[43]:
```
<seaborn.axisgrid.JointGrid at 0x2ca309347f0>
```



In [45]:
```
sns.jointplot(df['Total expenditure'],df['infant deaths'],hue=df['Status'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

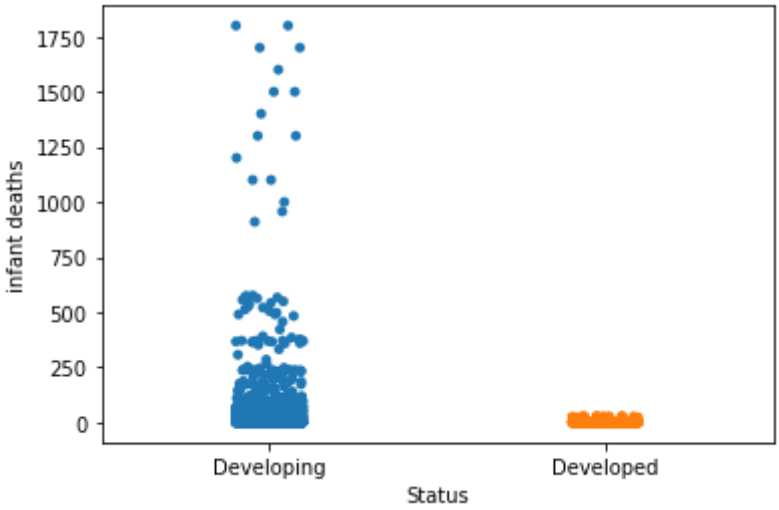
Out[45]:
```
<seaborn.axisgrid.JointGrid at 0x2ca3295e280>
```



In [46]:
```
sns.stripplot(df['Status'],df['infant deaths'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

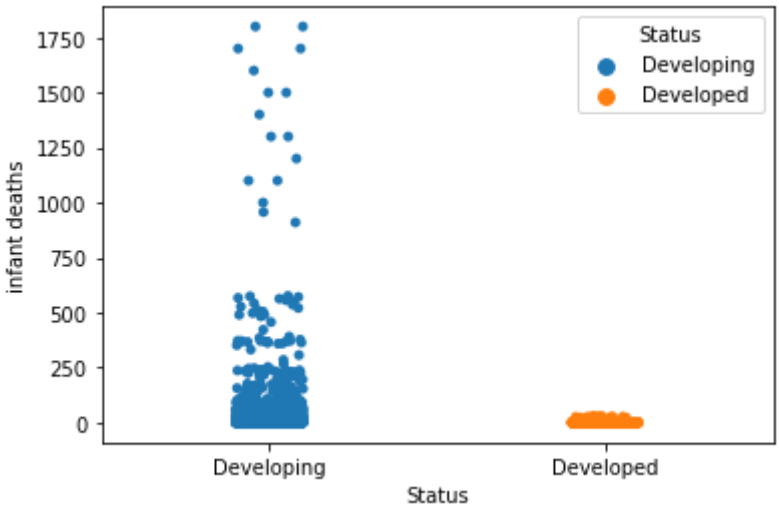
Out[46]:
```
<AxesSubplot:xlabel='Status', ylabel='infant deaths'>
```



In [47]:
```
sns.stripplot(df['Status'],df['infant deaths'],hue=df['Status'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[47]:
```
<AxesSubplot:xlabel='Status', ylabel='infant deaths'>
```

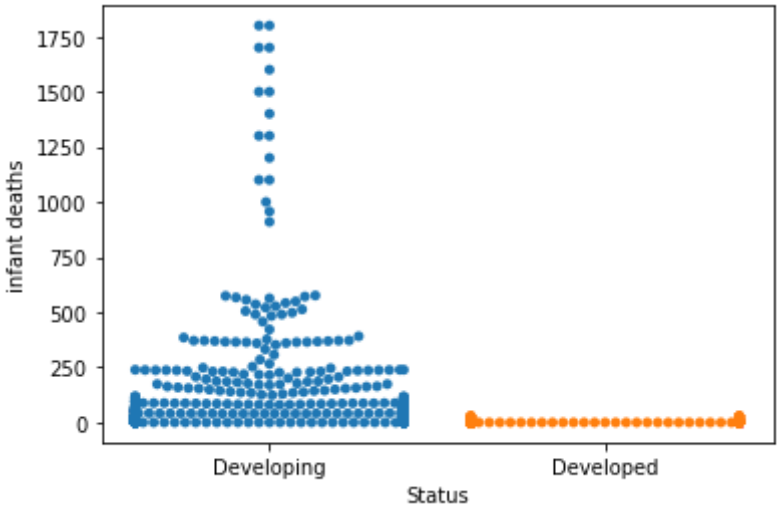


In [48]:
```
sns.swarmplot(df['Status'],df['infant deaths'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\HP\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 9
```

```
2.0% of the points cannot be placed; you may want to decrease the size of the marker
s or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\HP\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 9
5.1% of the points cannot be placed; you may want to decrease the size of the marker
s or use stripplot.
  warnings.warn(msg, UserWarning)
```

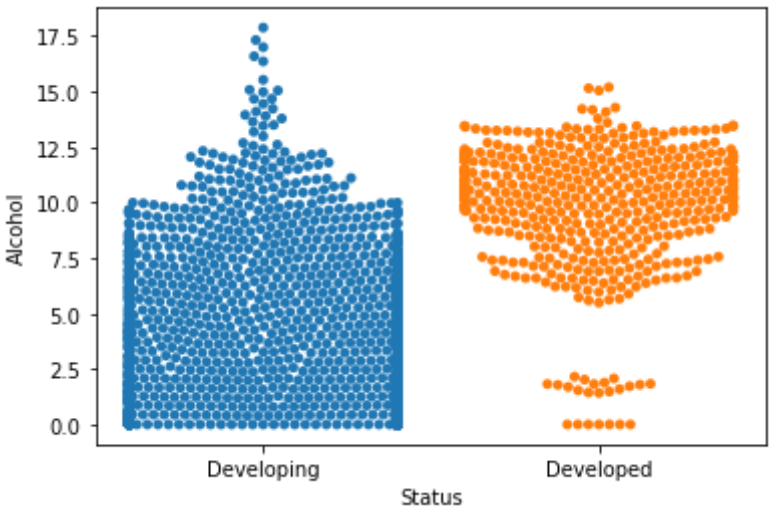
Out[48]:
```
<AxesSubplot:xlabel='Status', ylabel='infant deaths'>
```



In [22]:
```
sns.swarmplot(df['Status'],df['Alcohol'])
```

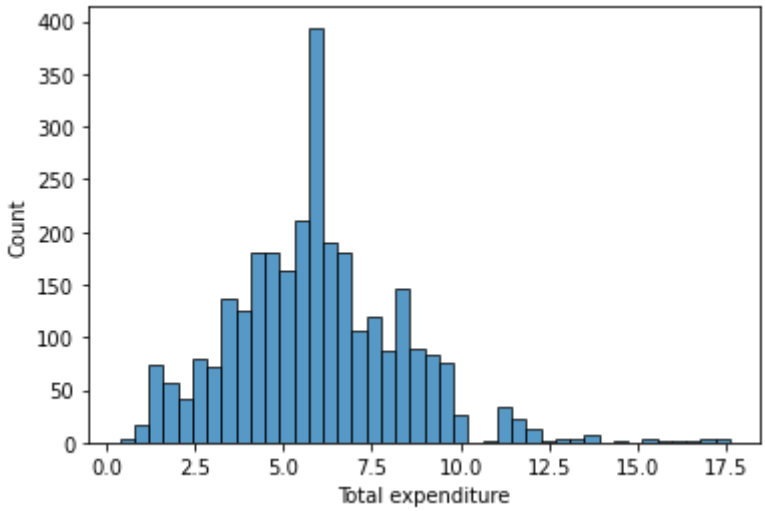
```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pa
ss the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\HP\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 6
3.6% of the points cannot be placed; you may want to decrease the size of the marker
s or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\HP\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 1
6.4% of the points cannot be placed; you may want to decrease the size of the marker
s or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[22]:
```
<AxesSubplot:xlabel='Status', ylabel='Alcohol'>
```



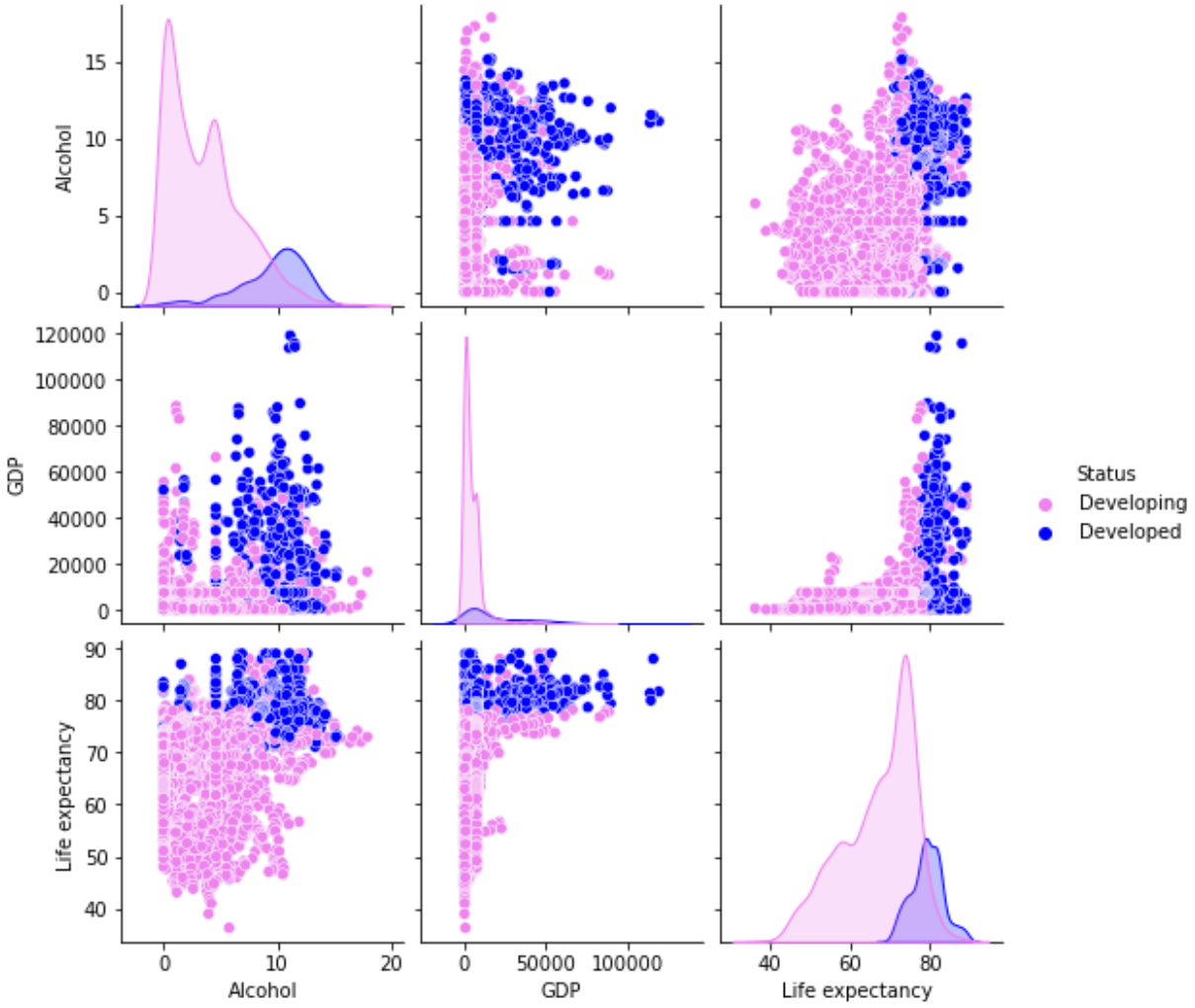
In [44]:
```
sns.histplot(df['Total expenditure'
```

Out[44]:
```
<AxesSubplot:xlabel='Total expenditure', ylabel='Count'>
```



Multi Variate

In [27]:
```
sns.pairplot(
    data = df[[ "Alcohol", "GDP", "Status", "Life expectancy "]],
    hue = 'Status' , palette = ['Violet', 'Blue']);
```



In [21]:
```
px.histogram(df,df["Life expectancy "], title="Life expectancy distribution")
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_22040/1656893340.py in <module>
----> 1 px.histogram(df,df["Life expectancy "], title="Life expectancy distributio
n")

~\anaconda3\lib\site-packages\_plotly_utils\importers.py in __getattr__(import_name)
     37             return getattr(class_module, class_name)
     38
---> 39         raise AttributeError(
     40             "module {__name__!r} has no attribute {name!r}".format(
     41                 name=import_name, __name__=parent_name

AttributeError: module 'plotly' has no attribute 'histogram'
```

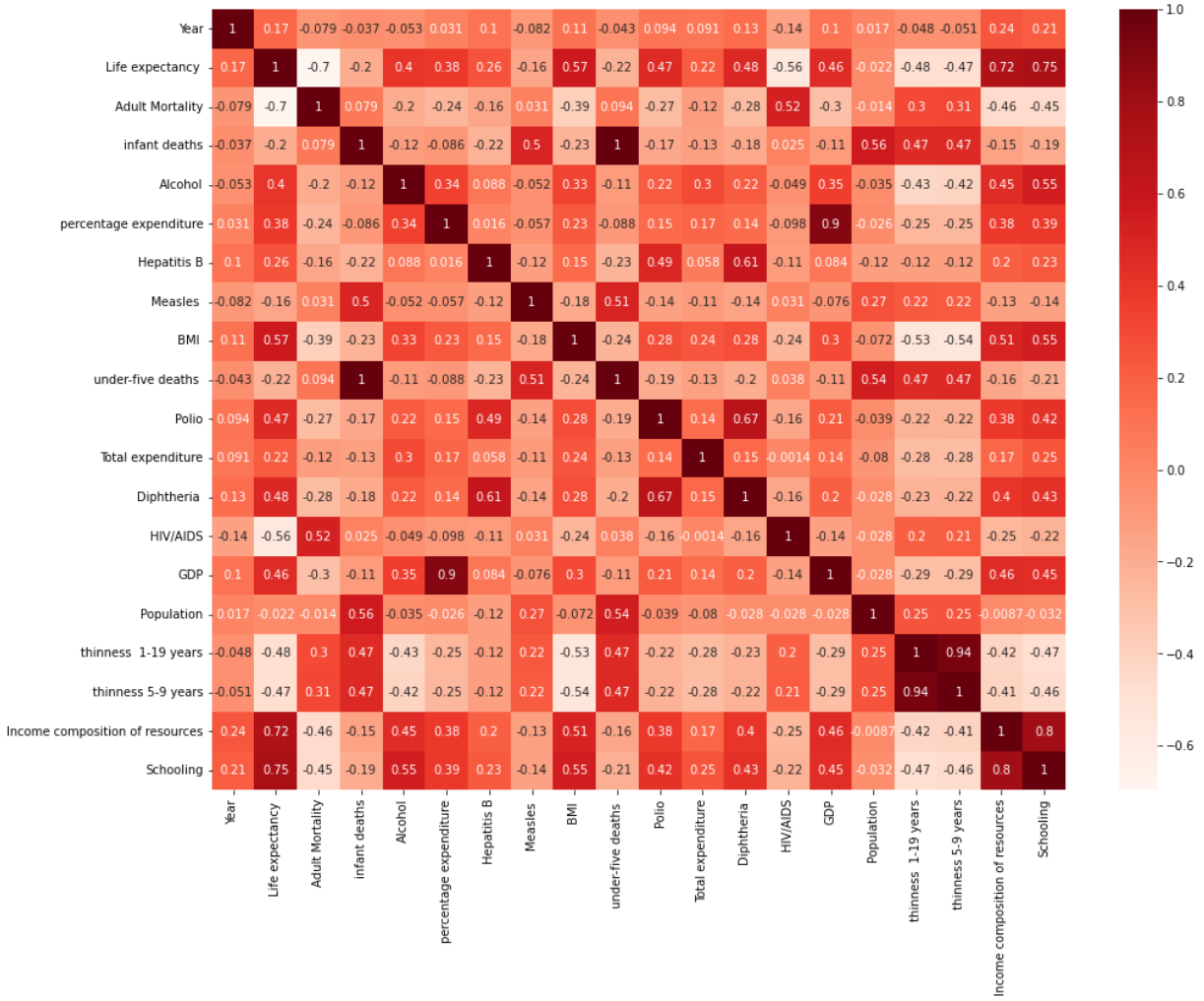
In [24]:
```
df.corr()
```

Out[24]:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measl
Year	1.000000	0.170033	-0.079052	-0.037415	-0.052990	0.031400	0.104333	-0.0824
Life expectancy	0.170033	1.000000	-0.696359	-0.196557	0.404877	0.381864	0.256762	-0.1575
Adult Mortality	-0.079052	-0.696359	1.000000	0.078756	-0.195848	-0.242860	-0.162476	0.0311
infant deaths	-0.037415	-0.196557	0.078756	1.000000	-0.115638	-0.085612	-0.223566	0.5011
Alcohol	-0.052990	0.404877	-0.195848	-0.115638	1.000000	0.341285	0.087549	-0.0518
percentage expenditure	0.031400	0.381864	-0.242860	-0.085612	0.341285	1.000000	0.016274	-0.0565
Hepatitis B	0.104333	0.256762	-0.162476	-0.223566	0.087549	0.016274	1.000000	-0.1205
Measles	-0.082493	-0.157586	0.031176	0.501128	-0.051827	-0.056596	-0.120529	1.0000
BMI	0.108974	0.567694	-0.387017	-0.227279	0.330408	0.228700	0.150380	-0.1759
under-five deaths	-0.042937	-0.222529	0.094146	0.996629	-0.112370	-0.087852	-0.233126	0.5078
Polio	0.094158	0.465556	-0.274823	-0.170689	0.221734	0.147259	0.486171	-0.1361
Total expenditure	0.090740	0.218086	-0.115281	-0.128616	0.296942	0.174420	0.058280	-0.1062
Diphtheria	0.134337	0.479495	-0.275131	-0.175171	0.222020	0.143624	0.611495	-0.1418
HIV/AIDS	-0.139741	-0.556556	0.523821	0.025231	-0.048845	-0.097857	-0.112675	0.0308
GDP	0.101620	0.461455	-0.296049	-0.108427	0.354712	0.899373	0.083903	-0.0764
Population	0.016969	-0.021538	-0.013647	0.556801	-0.035252	-0.025662	-0.123321	0.2659
thinness 1-19 years	-0.047876	-0.477183	0.302904	0.465711	-0.428795	-0.251369	-0.120429	0.2248
thinness 5-9 years	-0.050929	-0.471584	0.308457	0.471350	-0.417414	-0.252905	-0.124960	0.2210
Income composition of resources	0.243468	0.724776	-0.457626	-0.145139	0.450040	0.381952	0.199549	-0.1295

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measle
Schooling	0.209400	0.751975	-0.454612	-0.193720	0.547378	0.389687	0.231117	-0.13722

In [30]:
```
plt.figure(figsize=(16,12))
sns.heatmap(df.corr(),annot=True,cmap='Reds')
plt.show()
```



Countries Vs Exceptancy

In [15]:
```
print("Top 10 Countries with Most Life Expectancy")
df.groupby("Country").agg({
    "Life expectancy ":"mean"
}).reset_index().sort_values("Life expectancy ", ascending = False).head(10)
```

```
Top 10 Countries with Most Life Expectancy
```

Out[15]:

	Country	Life expectancy
84	Japan	82.53750
165	Sweden	82.51875
75	Iceland	82.44375
166	Switzerland	82.33125

	Country	Life expectancy
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 [16]:
```
print("Top 10 Countries with Least Life Expectancy")
df.groupby("Country").agg({
    "Life expectancy ":"mean"
}).reset_index().sort_values("Life expectancy ", ascending = True).head(10)
```

```
Top 10 Countries with Least Life Expectancy
```

Out[16]:

	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

Status of the Countries Vs Life exceptancy

In [17]:
```
df.groupby("Status").agg({
    "Life expectancy ":"mean"
}).reset_index().sort_values("Life expectancy ", ascending = False)
```

Out[17]:

	Status	Life expectancy
0	Developed	79.197852
1	Developing	67.111465

Countries Vs GDP

In [18]:
```
print("Top 10 Countries with Highest GDP")
df.groupby("Country").agg({
    "GDP":"mean"
}).reset_index().sort_values("GDP", ascending = False).head(10)
```

Top 10 Countries with Highest GDP

Out[18]:

	Country	GDP
166	Switzerland	57362.874601
98	Luxembourg	53257.012741
136	Qatar	40748.444104
119	Netherlands	34964.719797
7	Australia	34637.565047
80	Ireland	33835.272005
8	Austria	33827.476309
47	Denmark	33067.407916
153	Singapore	32790.105907
89	Kuwait	31914.378339

In [19]:

```
print("Top 10 Countries with Lowest GDP")
df.groupby("Country").agg({
    "GDP":"mean"
}).reset_index().sort_values("GDP", ascending = True).head(10)
```

Top 10 Countries with Lowest GDP

Out[19]:

	Country	GDP
117	Nauru	136.183210
26	Burundi	137.815321
100	Malawi	237.504042
95	Liberia	246.281748
55	Eritrea	259.395356
122	Niger	259.782441
57	Ethiopia	264.970950
152	Sierra Leone	271.505561
149	Senegal	274.611166
69	Guinea	279.464798

Top 10 countries with death due to HIV

In [20]:

```
print("Top 10 countries with deaths duo to HIV")
print("the number in the table represent deaths per 1000 live births")
df.groupby("Country").agg({
    " HIV/AIDS":"mean"
}).reset_index().sort_values(" HIV/AIDS", ascending = False).head(10)
```

```
Top 10 countries with deaths duo to HIV
the number in the table represent deaths per 1000 live births
```

Out[20]:

	Country	HIV/AIDS
164	Swaziland	32.94375
192	Zimbabwe	23.26250
94	Lesotho	22.96875
158	South Africa	18.49375
100	Malawi	16.68125
21	Botswana	16.52500
116	Namibia	13.64375
191	Zambia	11.93125
114	Mozambique	11.38750
31	Central African Republic	8.98125

Top 10 countries with high average Body Mass Index of entire population

In [21]:

```
print("Top 10 countries with with high average Body Mass Index of entire population")
df.groupby("Country").agg({
    " BMI ":"mean"
}).reset_index().sort_values(" BMI ", ascending = False).head(10)
```

```
Top 10 countries with with high average Body Mass Index of entire population
```

Out[21]:

	Country	BMI
117	Nauru	87.30000
128	Palau	83.30000
38	Cook Islands	82.80000
105	Marshall Islands	81.60000
178	Tuvalu	79.30000
124	Niue	77.30000
88	Kiribati	69.43125
104	Malta	66.18125
136	Qatar	65.65000
109	Micronesia (Federated States of)	65.15000

Top 10 countries with low average Body Mass Index of entire population

In [22]:

```
print("Top 10 countries with with low average Body Mass Index of entire population")
df.groupby("Country").agg({
    " BMI ":"mean"
}).reset_index().sort_values(" BMI ", ascending = True).head(10)
```

Top 10 countries with with low average Body Mass Index of entire population

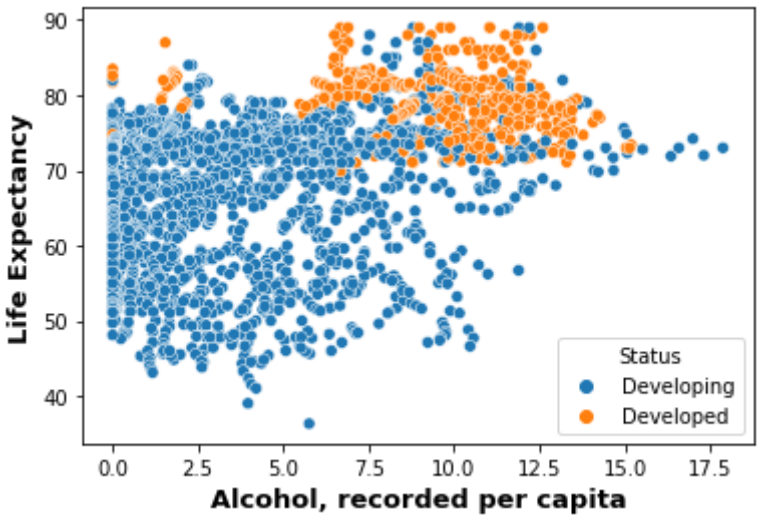
Out[22]:

	Country	BMI
142	Saint Kitts and Nevis	5.20000
189	Viet Nam	11.18750
12	Bangladesh	12.87500
91	Lao People's Democratic Republic	14.36250
171	Timor-Leste	14.55000
141	Rwanda	14.75000
99	Madagascar	14.76875
76	India	14.79375
57	Ethiopia	14.80000
55	Eritrea	15.15625

Alcohol, recorded per capita vs Life Expectancy based on status

In [23]:

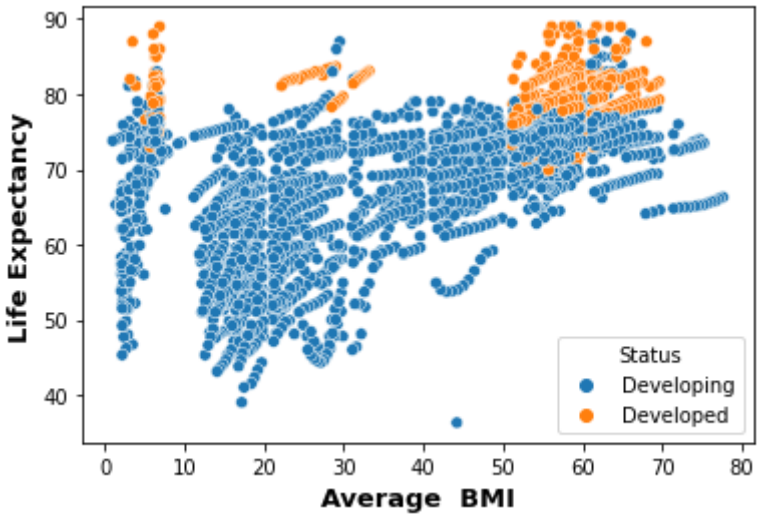
```
sns.scatterplot(x= df["Alcohol"], y= df["Life expectancy "] , hue= df["Status"])
plt.ylabel("Life Expectancy", fontsize= 13, fontweight="bold")
plt.xlabel("Alcohol, recorded per capita", fontsize=13, fontweight="bold")
plt.show()
```



Average BMI Vs Life Expectancy based on status

In [24]:

```
sns.scatterplot(x= df[" BMI "], y= df["Life expectancy "] , hue= df["Status"])
plt.ylabel("Life Expectancy", fontsize= 13, fontweight="bold")
plt.xlabel("Average  BMI", fontsize=13, fontweight="bold")
plt.show()
```



GDP Vs Life Expectancy based on status

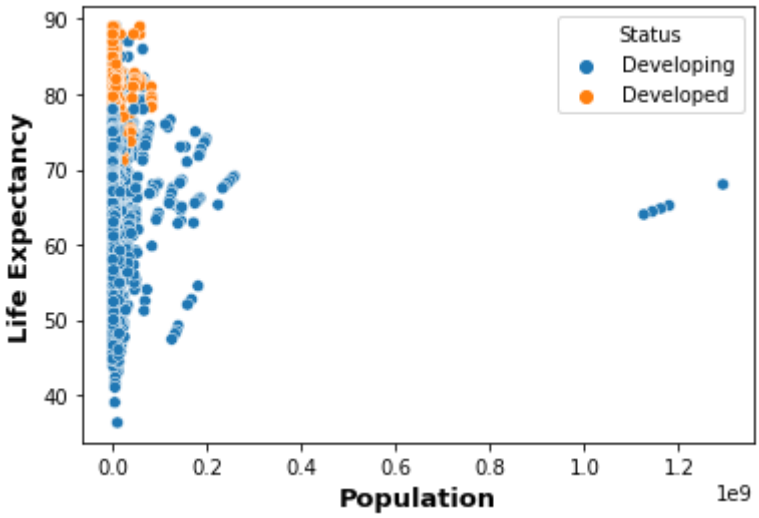
In [6]:
```
sns.scatterplot(x= df["GDP"], y= df["Life expectancy "] , hue= df["Status"])
plt.ylabel("Life Expectancy", fontsize= 13, fontweight="bold")
plt.xlabel("GDP", fontsize=13, fontweight="bold")
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_5260/2220680279.py in <module>
----> 1 sns.scatterplot(x= df["GDP"], y= df["Life expectancy "] , hue= df["Status"])
      2 plt.ylabel("Life Expectancy", fontsize= 13, fontweight="bold")
      3 plt.xlabel("GDP", fontsize=13, fontweight="bold")
      4 plt.show()

NameError: name 'df' is not defined
```

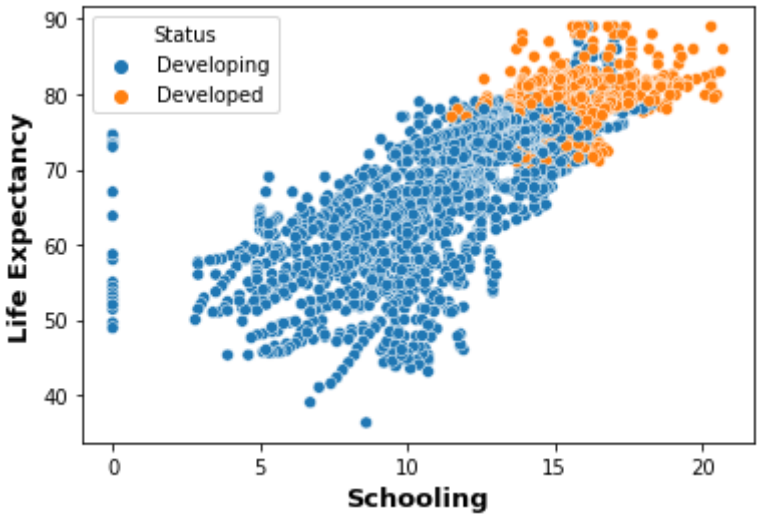
Population Vs Life Expectancy based on status

In [26]:
```
sns.scatterplot(x= df["Population"], y= df["Life expectancy "] , hue= df["Status"])
plt.ylabel("Life Expectancy", fontsize= 13, fontweight="bold")
plt.xlabel("Population", fontsize=13, fontweight="bold")
plt.show()
```



Schooling Vs Life Expectancy based on status

In [27]:
```
sns.scatterplot(x= df["Schooling"], y= df["Life expectancy "] , hue= df["Status"])
plt.ylabel("Life Expectancy", fontsize= 13, fontweight="bold")
plt.xlabel("Schooling", fontsize=13, fontweight="bold")
plt.show()
```



In [31]:
```
df.corr()['Life expectancy '].nlargest(15)
```

Out[31]:
```
Life expectancy                  1.000000
Schooling                        0.747556
Income composition of resources  0.724790
 BMI                             0.565697
Diphtheria                       0.478427
Polio                            0.464486
GDP                              0.461126
Alcohol                          0.403077
percentage expenditure           0.381418
Hepatitis B                      0.255452
Total expenditure                0.217304
Year                             0.170819
Population                      -0.021600
Measles                         -0.157767
infant deaths                   -0.196769
Name: Life expectancy , dtype: float64
```

Statiscal Test

There is no signifiance difference between life expectancy and country-H0

There is signifiance difference between life expectancy and country-H1

Level of Signifiance. alpha = 0.05

In [12]:

```
country1_life_expectancy = [72.5, 74.2, 75.1, 73.6, 71.8, 70.9, 73.2, 72.1, 74.5, 75
country2_life_expectancy = [68.9, 69.8, 71.2, 72.1, 67.3, 69.5, 70.1, 71.8, 69.9, 68

# Calculate the mean and standard deviation for each country's life expectancy
country1_mean = np.mean(country1_life_expectancy)
country1_std = np.std(country1_life_expectancy)
country2_mean = np.mean(country2_life_expectancy)
country2_std = np.std(country2_life_expectancy)

# Perform a two-sample t-test to compare the means of the two groups
t_stat, p_value = ttest_ind(country1_life_expectancy, country2_life_expectancy, equa

# Print the results of the t-test
print("t-statistic: {:.3f}".format(t_stat))
print("p-value: {:.3f}".format(p_value))
if p_value < 0.05:
    print("There is significant  difference in life expectancy between the two count
else:
    print("There is no significant difference in life expectancy between the two cou
```

```
t-statistic: 5.131
p-value: 0.000
There is significant  difference in life expectancy between the two countries is sig
nificant.
```

The End