In [2]:

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

In [3]:

```
#Load the data with pandas
df = pd.read_csv('../gapminder.csv', low_memory=False)
df.head()
```

Out[3]:

|  | country | incomeperperson | alcconsumption | armedforcesrate | breastcancerper100th | co2emissions | femaleemployra |
|---|---|---|---|---|---|---|---|
| 0 | Afghanistan |  | .03 | .5696534 | 26.8 | 75944000 | 25.60000038146 |
| 1 | Albania | 1914.99655094922 | 7.29 | 1.0247361 | 57.4 | 223747333.333333 | 42.09999847412 |
| 2 | Algeria | 2231.99333515006 | .69 | 2.306817 | 23.5 | 2932108666.66667 | 31.70000076293 |
| 3 | Andorra | 21943.3398976022 | 10.17 |  |  |  |  |
| 4 | Angola | 1381.00426770244 | 5.57 | 1.4613288 | 23.1 | 248358000 | 69.40000152587 |

In [4]:

```
# Convert columns from strings to numeric
columns = ['incomeperperson', 'alcconsumption', 'armedforcesrate', 'breastcancerper100th
', 'co2emissions', 'femaleemployrate', 'hivrate', 'internetuserate', 'lifeexpectancy', '
oilperperson', 'polityscore', 'relectricperperson', 'suicideper100th', 'employrate', 'urb
anrate']
data = df[columns].apply(pd.to_numeric, errors='coerce')
data['country'] = df['country']
data
```

Out[4]:

|  | incomeperperson | alcconsumption | armedforcesrate | breastcancerper100th | co2emissions | femaleemployrate | hivrate | inte |
|---|---|---|---|---|---|---|---|---|
| 0 | NaN | 0.03 | 0.569653 | 26.8 | 7.594400e+07 | 25.600000 | NaN |  |
| 1 | 1914.996551 | 7.29 | 1.024736 | 57.4 | 2.237473e+08 | 42.099998 | NaN |  |
| 2 | 2231.993335 | 0.69 | 2.306817 | 23.5 | 2.932109e+09 | 31.700001 | 0.1 |  |
| 3 | 21943.339898 | 10.17 | NaN | NaN | NaN | NaN | NaN |  |
| 4 | 1381.004268 | 5.57 | 1.461329 | 23.1 | 2.483580e+08 | 69.400002 | 2.0 |  |
| ... | ... | ... | ... | ... | ... | ... | ... |  |
| 208 | 722.807559 | 3.91 | 1.085367 | 16.2 | 1.425435e+09 | 67.599998 | 0.4 |  |
| 209 | NaN | NaN | 5.936085 | NaN | 1.424133e+07 | 11.300000 | NaN |  |
| 210 | 610.357367 | 0.20 | 2.316235 | 35.1 | 2.348647e+08 | 20.299999 | NaN |  |
| 211 | 432.226337 | 3.56 | 0.341335 | 13.0 | 1.320257e+08 | 53.500000 | 13.5 |  |
| 212 | 320.771890 | 4.96 | 1.032785 | 19.0 | 5.902197e+08 | 58.099998 | 14.3 |  |

213 rows × 16 columns

## Null Values

In [18]:

```
#Remove null values
data= data.replace(0, np.NaN)
data = data.dropna()
```

## Limit Data

In [23]:

```
#Countries with lowest incomes
lowestData = data[data['incomeperperson'] < 749]
```

In [24]:

```
#Countries with the hieghest incomes
highestData = data[data['incomeperperson'] > 9379]
```

## Calculate frequencies

In [26]:

```
print("Values for incomeperperson:")
incomeperperson_freq = pd.concat(dict(counts = data["incomeperperson"].value_counts(sort
=False, bins=10, dropna=False), percentages = data["incomeperperson"].value_counts(sort=
False, bins=10, dropna=False, normalize=True)), axis=1)
print(incomeperperson_freq)
```

```
Values for incomeperperson:
                        counts  percentages
(518.648, 4499.492]         22     0.392857
(4499.492, 8440.921]         9     0.160714
(8440.921, 12382.35]         3     0.053571
(12382.35, 16323.779]        3     0.053571
(16323.779, 20265.208]       2     0.035714
(20265.208, 24206.637]       1     0.017857
(24206.637, 28148.066]       8     0.142857
(28148.066, 32089.495]       1     0.017857
(32089.495, 36030.924]       3     0.053571
(36030.924, 39972.353]       4     0.071429
```

In [32]:

```
print("Values for lifeexpectancy:")
lifeexpectancy_freq = pd.concat(dict(counts = data["lifeexpectancy"].value_counts(sort=F
alse, bins=10, dropna=False), percentages = data["lifeexpectancy"].value_counts(sort=Fal
se, bins=10, dropna=False, normalize=True)), axis=1)
print(lifeexpectancy_freq)
```

```
Values for lifeexpectancy:
                  counts  percentages
(52.765, 55.857]       1     0.017857
(55.857, 58.916]       0     0.000000
(58.916, 61.976]       0     0.000000
(61.976, 65.036]       0     0.000000
(65.036, 68.095]       3     0.053571
(68.095, 71.155]       8     0.142857
(71.155, 74.215]      12     0.214286
(74.215, 77.275]       7     0.125000
(77.275, 80.334]       9     0.160714
(80.334, 83.394]      16     0.285714
```

In [34]:

```
print("Values for internetuserate:")
internetuserate_freq = pd.concat(dict(counts = data["internetuserate"].value_counts(sort
=False, bins=10, dropna=False), percentages = data["internetuserate"].value_counts(sort=
False, bins=10, dropna=False, normalize=True)), axis=1)
print(internetuserate_freq)
```

```
Values for internetuserate:
```

```
Values for internetuserate:
                  counts  percentages
(3.609, 12.658]        5     0.089286
(12.658, 21.616]       4     0.071429
(21.616, 30.573]       3     0.053571
(30.573, 39.531]       7     0.125000
(39.531, 48.489]       9     0.160714
(48.489, 57.447]       3     0.053571
(57.447, 66.404]       4     0.071429
(66.404, 75.362]       5     0.089286
(75.362, 84.32]       10     0.178571
(84.32, 93.278]        6     0.107143
```

In [35]:

```
print("Values for breastcancerper100th:")
breastcancerper100th_freq = pd.concat(dict(counts = data["breastcancerper100th"].value_c
ounts(sort=False, bins=10, dropna=False), percentages = data["breastcancerper100th"].val
ue_counts(sort=False, bins=10, dropna=False, normalize=True)), axis=1)
print(breastcancerper100th_freq)
```

```
Values for breastcancerper100th:
                             counts  percentages
(16.514999999999997, 25.05]      11     0.196429
(25.05, 33.5]                     7     0.125000
(33.5, 41.95]                     7     0.125000
(41.95, 50.4]                     9     0.160714
(50.4, 58.85]                     4     0.071429
(58.85, 67.3]                     1     0.017857
(67.3, 75.75]                     5     0.089286
(75.75, 84.2]                     3     0.053571
(84.2, 92.65]                     8     0.142857
(92.65, 101.1]                    1     0.017857
```

## New Categorical Variables

In [39]:

```
#Categorical variable to label incomeperperson
data['incomelabel'] =pd.cut(data.incomeperperson,4,labels=['low','medium','high','very hi
gh'])
income_freq = pd.concat(dict(counts = data["incomelabel"].value_counts(sort=False, dropn
a=False),
                                   percentages = data["incomelabel"].value_counts(sort=F
alse, dropna=False,

normalize=True)),
                            axis=1)
```

In [ ]: