In [1]:

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

Read Data

In [2]:

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

Out[2]:

	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 [3]:

```
#Number of rows and columns
print('Number of observations (rows)', len(df))
print('Number of variables (cols)', len(df.columns))
```

```
Number of observations (rows) 213
Number of variables (cols) 16
```

In [4]:

```
# bug fix for display formats to avoid run time errors
pd.set_option('display.float_format', lambda x:'%f'%x)
```

In [5]:

```
# 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[5]:

	incomeperperson	alcconsumption	armedforcesrate	breastcancerper100th	co2emissions	femaleemployrate	hivra
0	nan	0.030000	0.569653	26.800000	75944000.000000	25.600000	r
1	1914.996551	7.290000	1.024736	57.400000	223747333.333333	42.099998	r
2	2231.993335	0.690000	2.306817	23.500000	2932108666.666670	31.700001	0.1000
3	21943.339898	10.170000	nan	nan	nan	nan	r
4	1381.004268	5.570000	1.461329	23.100000	248358000.000000	69.400002	2.0000
...	...	...	...	...	...	...	
208	722.807559	3.910000	1.085367	16.200000	1425435000.000000	67.599998	0.4000

	incomeperperson	alcconsumption	armedforcesrate	breastcancerper100th	co2emissions	femaleemployrate	hivr
209	nan	nan	5.936085	nan	142413333.333333	11.300000	
210	610.357367	0.200000	2.316235	35.100000	234864666.666667	20.299999	r
211	432.226337	3.560000	0.341335	13.000000	132025666.666667	53.500000	13.5000
212	320.771890	4.960000	1.032785	19.000000	590219666.666666	58.099998	14.3000

213 rows × 16 columns

In [6]:

```
#Number of null values per variable
listinternet = data['internetuserate']
print("internetuserate", np.sum(listinternet.isnull()))
listIncomes = data['incomeperperson']
print("incomeperperson", np.sum(listIncomes.isnull()))
listLife = data['lifeexpectancy']
print("lifeexpectancy", np.sum(listLife.isnull()))
listCancer = data['breastcancerper100th']
print("breastcancerper100th", np.sum(listCancer.isnull()))
```

```
internetuserate 21
incomeperperson 23
lifeexpectancy 22
breastcancerper100th 40
```

In [7]:

```
# Statistics summary about the incomeperperson
data['incomeperperson'].describe()
```

Out[7]:

```
count      190.000000
mean      8740.966076
std      14262.809083
min        103.775857
25%        748.245151
50%       2553.496056
75%       9379.891165
max     105147.437697
Name: incomeperperson, dtype: float64
```

Limit Data

In [8]:

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

In [9]:

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

Life Expectancy

Lowest Incomes

In [10]:

```
#Create dataframe to store values
lifeL = pd.DataFrame()
lifeL['value'] = ['(47.7, 50.5]  ', '(50.5, 53.2]', '(53.2, 56]', '(56, 58.7]', '(58.7,
```

```
61.4]',
'(61.4, 64.2]', '(64.2, 66.9]', '(66.9, 69.7]', '(69.7, 72.4]', '(72.4, 75.1]']
```

In [11]:

```
# frequency and percentage distritions for lifeexpectancy with lower incomes
cLifeL = lowestData['lifeexpectancy'].value_counts(sort=False,bins=10)
pLifeL = lowestData['lifeexpectancy'].value_counts(sort=False,bins=10,normalize=True)*10
0

# add values to dataframe
lifeL['freq'] = cLifeL.tolist()
lifeL['percent'] = pLifeL.tolist()

#cumulative frequency and cumulative percentage for lifeexpectancy with lower incomes
lifeL['cum freq'] = lifeL['freq'].cumsum()
lifeL['cum percent'] = lifeL['percent'].cumsum()
```

In [12]:

```
lifeL
```

Out[12]:

	value	freq	percent	cum freq	cum percent
0	(47.7, 50.5]	9	18.750000	9	18.750000
1	(50.5, 53.2]	4	8.333333	13	27.083333
2	(53.2, 56]	7	14.583333	20	41.666667
3	(56, 58.7]	7	14.583333	27	56.250000
4	(58.7, 61.4]	4	8.333333	31	64.583333
5	(61.4, 64.2]	5	10.416667	36	75.000000
6	(64.2, 66.9]	4	8.333333	40	83.333333
7	(66.9, 69.7]	7	14.583333	47	97.916667
8	(69.7, 72.4]	0	0.000000	47	97.916667
9	(72.4, 75.1]	1	2.083333	48	100.000000

Highest Incomes

In [13]:

```
#Create dataframe to store values
lifeH = pd.DataFrame()
lifeH['value'] = [ '(70.1, 71.4]', '(71.4, 72.7]', '(72.7, 74.1]', '(74.1, 75.4]', '(75.
4, 76.7]', '(76.7, 78]', '(78, 79.4]', '(79.4, 80.7]', '(80.7, 82]', '(82, 83.3]' ]
```

In [14]:

```
# frequency and percentage distritions for lifeexpectancy with higher incomes
cLifeH = highestData['lifeexpectancy'].value_counts(sort=False,bins=10)
pLifeH = highestData['lifeexpectancy'].value_counts(sort=False,bins=10,normalize=True)*1
00

# add values to dataframe
lifeH['freq'] = cLifeH.tolist()
lifeH['percent'] = pLifeH.tolist()

#cumulative frequency and cumulative percentage for lifeexpectancy with lower incomes
lifeH['cum freq'] = lifeH['freq'].cumsum()
lifeH['cum percent'] = lifeH['percent'].cumsum()
```

In [15]:

```
lifeH
```

Out[15]:

	value	freq	percent	cum freq	cum percent
0	(70.1, 71.4]	1	2.083333	1	2.083333
1	(71.4, 72.7]	0	0.000000	1	2.083333
2	(72.7, 74.1]	2	4.166667	3	6.250000
3	(74.1, 75.4]	1	2.083333	4	8.333333
4	(75.4, 76.7]	3	6.250000	7	14.583333
5	(76.7, 78]	1	2.083333	8	16.666667
6	(78, 79.4]	5	10.416667	13	27.083333
7	(79.4, 80.7]	13	27.083333	26	54.166667
8	(80.7, 82]	12	25.000000	38	79.166667
9	(82, 83.3]	3	6.250000	41	85.416667

Internet Use Rate

Lowest Incomes

In [16]:

```
#Create dataframe to store values
internetL = pd.DataFrame()
internetL['value'] = [ '(0.1, 4.2]', '(4.2, 8.1]', '(8.1, 12.1]', '(12.1, 16.1]', '(16.1
, 20.1]', '(20.1, 24.1]', '(24.1, 28.1]', '(28.1, 32.1]', '(32.1, 36.1]', '(36.1, 40.1]'
]
```

In [17]:

```
# frequency and percentage distritions for a number of internetuserate with lower incomes
cInternetL = lowestData['internetuserate'].value_counts(sort=False,bins=10)
pInternetL = lowestData['internetuserate'].value_counts(sort=False,bins=10,normalize=Tru
e)*100

# add values to dataframe
internetL['freq'] = cInternetL.tolist()
internetL['percent'] = pInternetL.tolist()

#cumulative frequency and cumulative percentage for internetuserate with lower incomes
internetL['cum freq'] = internetL['freq'].cumsum()
internetL['cum percent'] = internetL['percent'].cumsum()
```

In [18]:

```
internetL
```

Out[18]:

	value	freq	percent	cum freq	cum percent
0	(0.1, 4.2]	23	47.916667	23	47.916667
1	(4.2, 8.1]	6	12.500000	29	60.416667
2	(8.1, 12.1]	7	14.583333	36	75.000000
3	(12.1, 16.1]	4	8.333333	40	83.333333
4	(16.1, 20.1]	2	4.166667	42	87.500000
5	(20.1, 24.1]	0	0.000000	42	87.500000
6	(24.1, 28.1]	2	4.166667	44	91.666667
7	(28.1, 32.1]	1	2.083333	45	93.750000

	value	freq	percent	cum freq	cum percent
8	(32.1, 36.1]	0	0.000000	45	93.750000
9	(36.1, 40.1]	1	2.083333	46	95.833333

Highest Incomes

In [19]:

```
#Create dataframe to store values
internetH = pd.DataFrame()
internetH['value'] = [ '(35.9, 41.9]', '(41.9, 47.9]', '(47.9, 53.8]', '(53.8, 59.8]', '
(59.8, 65.8]', '(65.8, 71.7]', '(71.7, 77.7]', '(77.7, 83.7]', '(83.7, 89.6]', '(89.6, 9
5.6]' ]
```

In [20]:

```
# frequency and percentage distritions for internetuserate with higher incomes
cInternetH = highestData['internetuserate'].value_counts(sort=False,bins=10)
pInternetH = highestData['internetuserate'].value_counts(sort=False,bins=10,normalize=Tr
ue)*100

# add values to dataframe
internetH['freq'] = cInternetH.tolist()
internetH['percent'] = pInternetH.tolist()

#cumulative frequency and cumulative percentage for internetuserate with lower incomes
internetH['cum freq'] = internetH['freq'].cumsum()
internetH['cum percent'] = internetH['percent'].cumsum()
```

In [21]:

```
internetH
```

Out[21]:

	value	freq	percent	cum freq	cum percent
0	(35.9, 41.9]	2	4.166667	2	4.166667
1	(41.9, 47.9]	3	6.250000	5	10.416667
2	(47.9, 53.8]	5	10.416667	10	20.833333
3	(53.8, 59.8]	2	4.166667	12	25.000000
4	(59.8, 65.8]	5	10.416667	17	35.416667
5	(65.8, 71.7]	3	6.250000	20	41.666667
6	(71.7, 77.7]	7	14.583333	27	56.250000
7	(77.7, 83.7]	10	20.833333	37	77.083333
8	(83.7, 89.6]	4	8.333333	41	85.416667
9	(89.6, 95.6]	5	10.416667	46	95.833333

Breast Cancer

Lowest Incomes

In [22]:

```
#Create dataframe to store values
cancerL = pd.DataFrame()
cancerL['value'] = [ '(3.8, 8.5]', '(8.5, 13.1]', '(13.1, 17.7]', '(17.7, 22.3]', '(22.3
, 27]', '(27, 31.6]', '(31.6, 36.2]', '(36.2, 40.8]', '(40.8, 45.4]', '(45.4, 50.1]' ]
```

In [23]:

```
# frequency and percentage distritions for a number of breastcancerper100th with lower in
comes
cCancerL = lowestData['breastcancerper100th'].value_counts(sort=False,bins=10)
pCancerL = lowestData['breastcancerper100th'].value_counts(sort=False,bins=10,normalize=
True)*100

# add values to dataframe
cancerL['freq'] = cCancerL.tolist()
cancerL['percent'] = pCancerL.tolist()

#cumulative frequency and cumulative percentage for breastcancerper100th with lower incom
es
cancerL['cum freq'] = cancerL['freq'].cumsum()
cancerL['cum percent'] = cancerL['percent'].cumsum()
```

In [24]:

```
cancerL
```

Out[24]:

	value	freq	percent	cum freq	cum percent
0	(3.8, 8.5]	3	6.250000	3	6.250000
1	(8.5, 13.1]	6	12.500000	9	18.750000
2	(13.1, 17.7]	7	14.583333	16	33.333333
3	(17.7, 22.3]	12	25.000000	28	58.333333
4	(22.3, 27]	6	12.500000	34	70.833333
5	(27, 31.6]	9	18.750000	43	89.583333
6	(31.6, 36.2]	2	4.166667	45	93.750000
7	(36.2, 40.8]	0	0.000000	45	93.750000
8	(40.8, 45.4]	0	0.000000	45	93.750000
9	(45.4, 50.1]	2	4.166667	47	97.916667

Highest Incomes

In [25]:

```
#Create dataframe to store values
cancerH = pd.DataFrame()
cancerH['value'] = [ '(13.1, 21.9]', '(21.9, 30.7]', '(30.7, 39.5]', '(39.5, 48.3]', '(4
8.3, 57.1]', '(57.1, 65.9]', '(65.9, 74.7]', '(74.7, 83.5]', '(83.5, 92.3]', '(92.3, 101
.1]' ]
```

In [26]:

```
# frequency and percentage distritions for a number of breastcancerper100th with higher i
ncomes
cCancerH = highestData['breastcancerper100th'].value_counts(sort=False,bins=10)
pCancerH = highestData['breastcancerper100th'].value_counts(sort=False,bins=10,normalize
=True)*100

# add values to dataframe
cancerH['freq'] = cCancerH.tolist()
cancerH['percent'] = pCancerH.tolist()

#cumulative frequency and cumulative percentage for breastcancerper100th with lower incom
es
cancerH['cum freq'] = cancerH['freq'].cumsum()
cancerH['cum percent'] = cancerH['percent'].cumsum()
```

In [27]:

```
cancerH
```

```
cancerH
```

Out[27]:

	value	freq	percent	cum freq	cum percent
0	(13.1, 21.9]	3	6.250000	3	6.250000
1	(21.9, 30.7]	2	4.166667	5	10.416667
2	(30.7, 39.5]	2	4.166667	7	14.583333
3	(39.5, 48.3]	1	2.083333	8	16.666667
4	(48.3, 57.1]	7	14.583333	15	31.250000
5	(57.1, 65.9]	1	2.083333	16	33.333333
6	(65.9, 74.7]	4	8.333333	20	41.666667
7	(74.7, 83.5]	7	14.583333	27	56.250000
8	(83.5, 92.3]	11	22.916667	38	79.166667
9	(92.3, 101.1]	1	2.083333	39	81.250000