Unsupervised Learning - Categorising Countries on Basis of Socio-economic Factors

About organization:

HELP International is an international humanitarian NGO that is committed to fighting poverty and providing the people of backward countries with basic amenities and relief during the time of disasters and natural calamities.

Problem Statement:

HELP International have been able to raise around $ 10 million. Now the CEO of the NGO needs to decide how to use this money strategically and effectively. So, CEO has to make decision to **choose the countries** that are in the **direst need of aid**. Hence, the Job as a Data scientist is to **categorise the countries** using some **socio-economic and health factors** that determine the overall development of the country. Then you need to suggest the countries which the CEO needs to focus on the most.

Aim:

To create groups of countries based on their socio-economic factors to help in deciding on the recipents of the financial aid.

Table of contents:

Description of the features:

- **country** : Name of the country
- **child_mort** : Death of children under 5 years of age per 1000 live births
- **exports** : Exports of goods and services per capita. Given as %age of the GDP per capita
- **health** : Total health spending per capita. Given as %age of GDP per capita
- **imports** : Imports of goods and services per capita. Given as %age of the GDP per capita
- **Income** : Net income per person
- **Inflation** : The measurement of the annual growth rate of the Total GDP
- **life_expec** : The average number of years a new born child would live if the current mortality patterns are to remain the same.
- **total_fer** : The number of children that would be born to each woman if the current age-fertility rates remain the same.
- **gdpp** : The GDP per capita. Calculated as the Total GDP divided by the total population.

In [1]:

```
# importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

import warnings
warnings.filterwarnings('ignore')
```

Gathering the data

Source: https://www.kaggle.com/datasets/rohan0301/unsupervised-learning-on-country-data (https://www.kaggle.com/datasets/rohan0301/unsupervised-learning-on-country-data)

In [2]:

```
country_data = pd.read_csv("../input/unsupervised-learning-on-country-data/Country-data.csv")
data_descrip = pd.read_csv("../input/unsupervised-learning-on-country-data/data-dictionary.csv")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remo
unt, call drive.mount("/content/drive", force_remount=True).
```

In [4]:

```
display(country_data.head())
```

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	g
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12

In [5]:

```
print(data_descrip)
```

Data Preprocessing

In [6]:

```
country_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   country     167 non-null    object 
 1   child_mort  167 non-null    float64
 2   exports     167 non-null    float64
 3   health      167 non-null    float64
 4   imports     167 non-null    float64
 5   income      167 non-null    int64  
 6   inflation   167 non-null    float64
 7   life_expec  167 non-null    float64
 8   total_fer   167 non-null    float64
 9   gdpp        167 non-null    int64  
dtypes: float64(7), int64(2), object(1)
memory usage: 13.2+ KB
```

In [7]:

```
country_data.describe()
```

Out[7]:

	child_mort	exports	health	imports	income	inflation	life_expec
count	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000
mean	38.270060	41.108976	6.815689	46.890215	17144.688623	7.781832	70.555689
std	40.328931	27.412010	2.746837	24.209589	19278.067698	10.570704	8.893172
min	2.600000	0.109000	1.810000	0.065900	609.000000	-4.210000	32.100000
25%	8.250000	23.800000	4.920000	30.200000	3355.000000	1.810000	65.300000
50%	19.300000	35.000000	6.320000	43.300000	9960.000000	5.390000	73.100000
75%	62.100000	51.350000	8.600000	58.750000	22800.000000	10.750000	76.800000
max	208.000000	200.000000	17.900000	174.000000	125000.000000	104.000000	82.800000

In [8]:

```
country_data.isnull().any()
```

Out[8]:

```
country       False
child_mort    False
exports       False
health        False
imports       False
income        False
inflation     False
life_expec    False
total_fer     False
gdpp          False
dtype: bool
```

- There are no null values in the data.

In [9]:

```
country_data.duplicated().sum()
```

Out[9]:

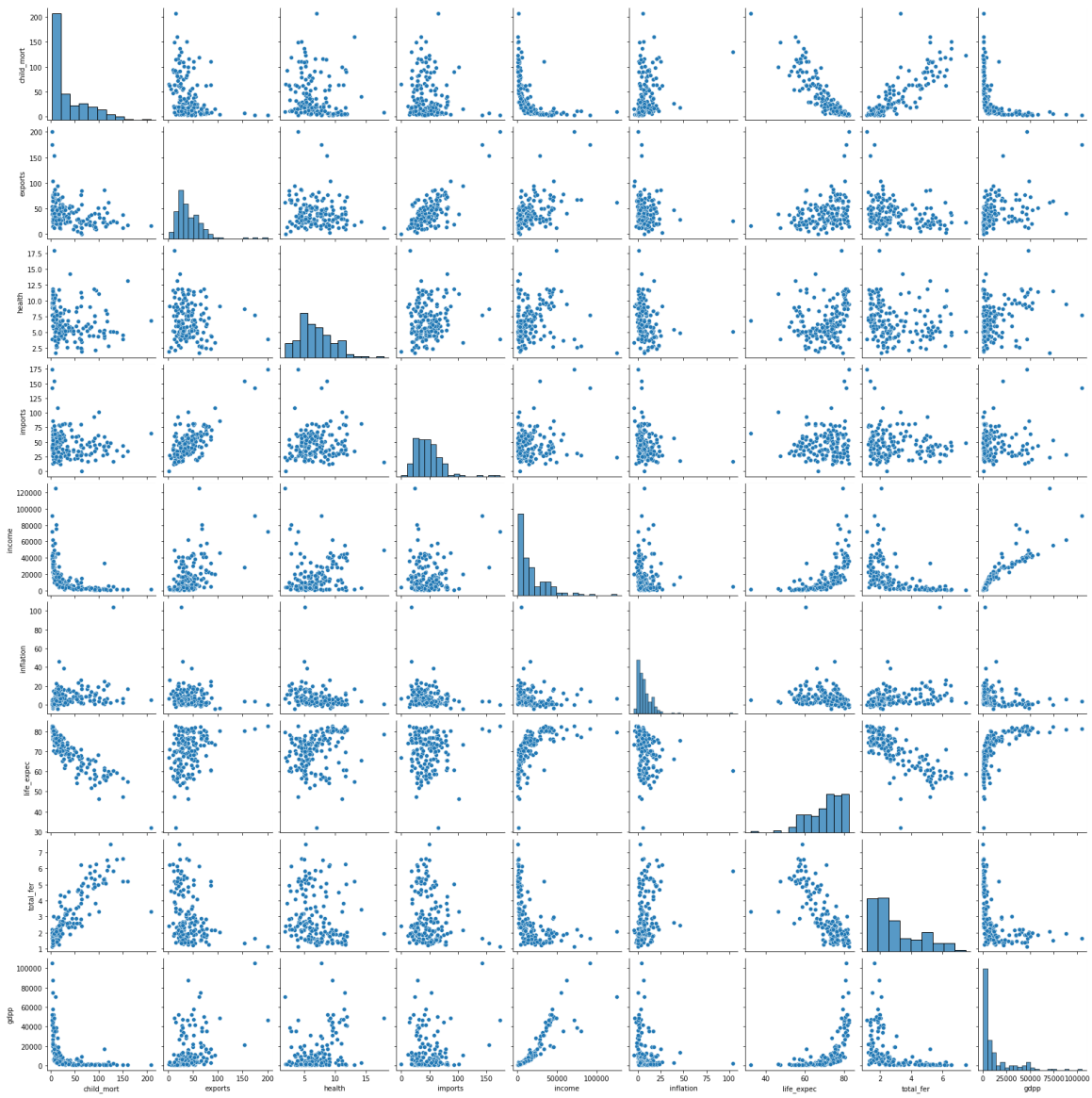
0

- There are now duplicate rows.

Plotting the correlation between each numerical feature

In [10]:

```
sns.pairplot(country_data)
plt.show()
```



Exploratory Data Analysis

In [11]:

```
cols = list(country_data.columns)
cols
```

Out[11]:

```
['country',
 'child_mort',
 'exports',
 'health',
 'imports',
 'income',
 'inflation',
 'life_expec',
 'total_fer',
 'gdpp']
```

In [12]:

```
numerical_cols = cols[1:]
categorical_cols = ['country']
```

In [13]:

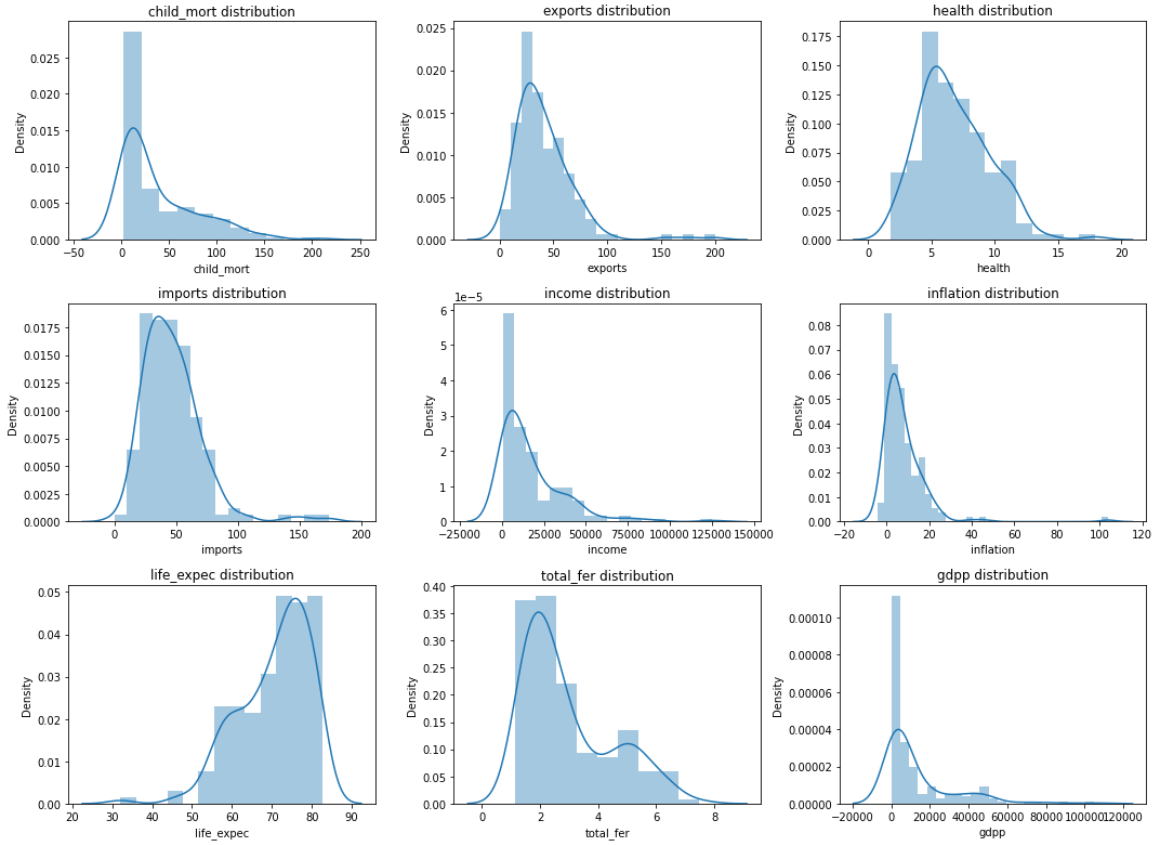
```
print(numerical_cols)
```

```
['child_mort', 'exports', 'health', 'imports', 'income', 'inflatio
n', 'life_expec', 'total_fer', 'gdpp']
```

- For all features except 'total_fer', we can find outliers in the distributions.
- **Outliers** significantly impact the clustering algorithm and hence they need to be suppressed.

In [14]:

```
# distplot
fig, ax = plt.subplots(nrows = 3, ncols = 3, figsize = (15,11))
for i in range(len(numerical_cols)):
  plt.subplot(3,3,i+1)
  sns.distplot(country_data[numerical_cols[i]])
  title = numerical_cols[i] + ' distribution'
  plt.title(title)
plt.tight_layout()
plt.show()
```



Key takeaways:

- Life expectancy distribution is negatively skewed.
- Health expenditure has an approximate normal distribution.
- All distributions except Life Expectancy show close to positive skewness.

Visualizing the important variables

GDP per capita

In [15]:

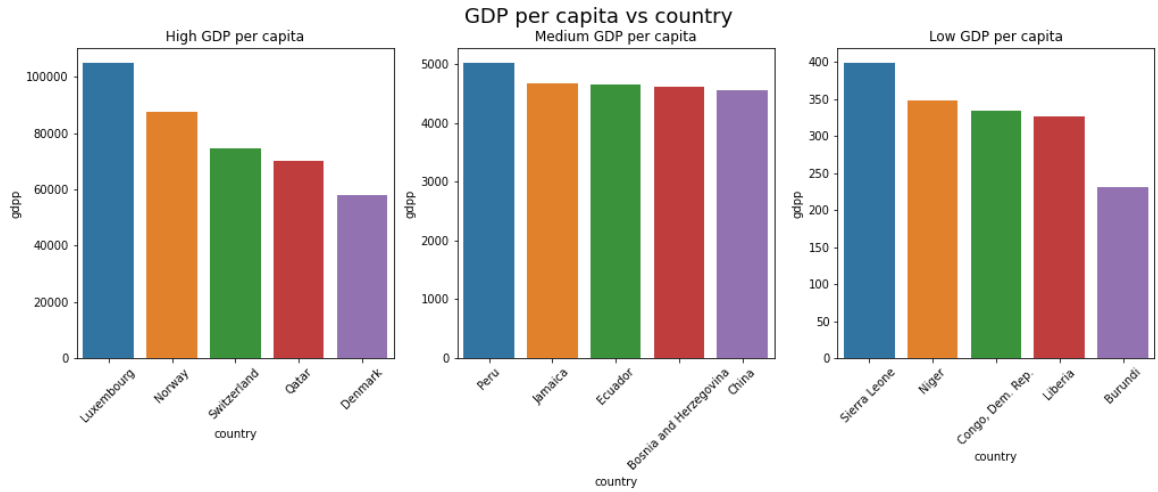
```
# Categories for visualization
catog = ['High', 'Medium', 'Low']
```

In [16]:

```
# Function to plot each feature vs country for the three categories - High, Medi
um and Low
def plots_catogs(feature, ttl_text, subplt_ttl):
  """
  Arguments:
  feature - Feature/variable considered for plotting (Example - income, gdpp, he
alth, child_mort etc.)
  ttl_text - Main title for the subplots
  This function takes in the feature name as the argument and creates three plot
s for each High, Medium and Low category for that particular feature passed.
  """
  fig, ax = plt.subplots(1,3,figsize = (17,5))
  sns.barplot(x = 'country', y = feature, data = country_data.sort_values(by = f
eature, ascending = False).iloc[:5], ax = ax[0])
  sns.barplot(x = 'country', y = feature, data = country_data.sort_values(by = f
eature, ascending = False).iloc[81:86], ax = ax[1])
  sns.barplot(x = 'country', y = feature, data = country_data.sort_values(by = f
eature, ascending = False).iloc[-5:], ax = ax[2])
  for i in range(3):
    title = catog[i] + ' ' + subplt_ttl
    ax[i].set_title(title)
    ax[i].tick_params(axis='x', labelrotation = 45)
  fig.suptitle(ttl_text, fontsize = 18)
```

In [17]:

```
title = 'GDP per capita vs country'
plots_catogs('gdpp', title, 'GDP per capita')
plt.show()
```

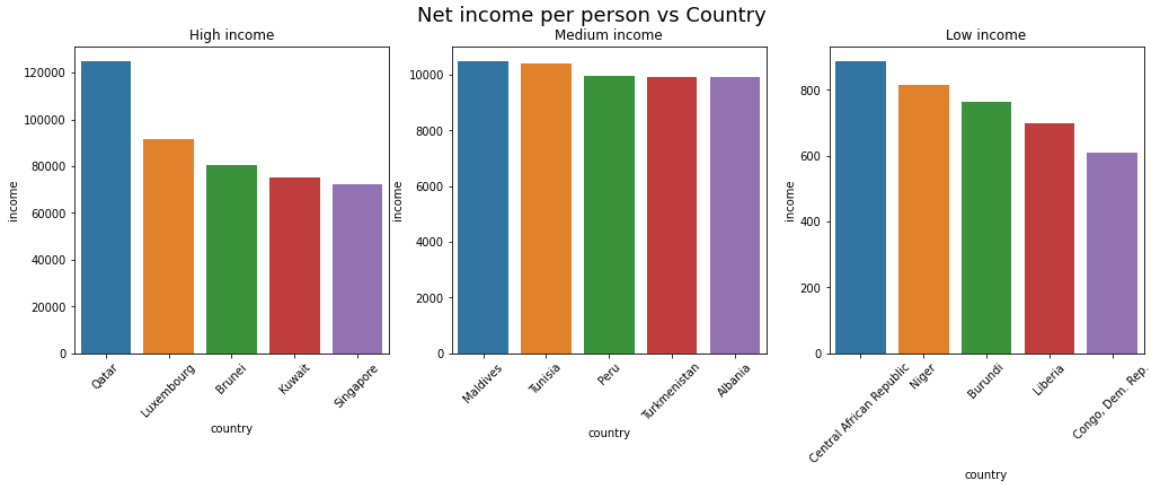


- **GDP per capita** is a **measure** of the **standard of living, prosperity and overall well-being in a country**. Low GDP per capita means that the country is unable to supply the country's population with basic goods and services.
- GDP growth also depends on the population of the country.
- **Luxembourg leads** in GDP per capita and **4 out of 5 countries** come from **Europe** which lie at the top end. **Qatar** is the **only country** from **Asia/Middle East** occupying this list.
- **African countries** lie at the **bottom end** for GDP per capita. This gives an idea of poor economic conditions with a weak GDP.

Income - net income per person

In [18]:

```
title = 'Net income per person vs Country'
plots_catogs('income', title, 'income')
plt.show()
```

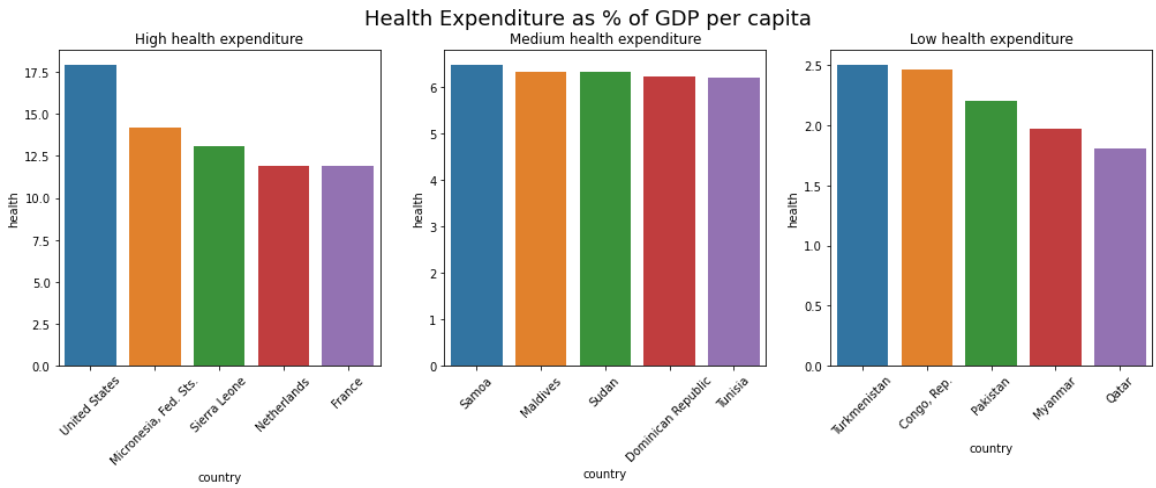


- **3 out of 5 countries** are from **Asia/Middle East** with **Qatar leading** the way at **USD120000 net income earned per person**. In addition, **Singapore** and Luxembourg also occupy this list.
- **Majority** of the countries in the **bottom 5** are from **Africa**. The reasons for this could be **due to poor employment conditions** and their **weaker economies**. Moreover, African nations lie in the bottom list for GDP per capita.
- In the High, Medium and Low categories, the **range of the values** show **large variations**.

Health expenditure

In [19]:

```
title = 'Health Expenditure as % of GDP per capita'
plots_catogs('health', title, 'health expenditure')
plt.show()
```

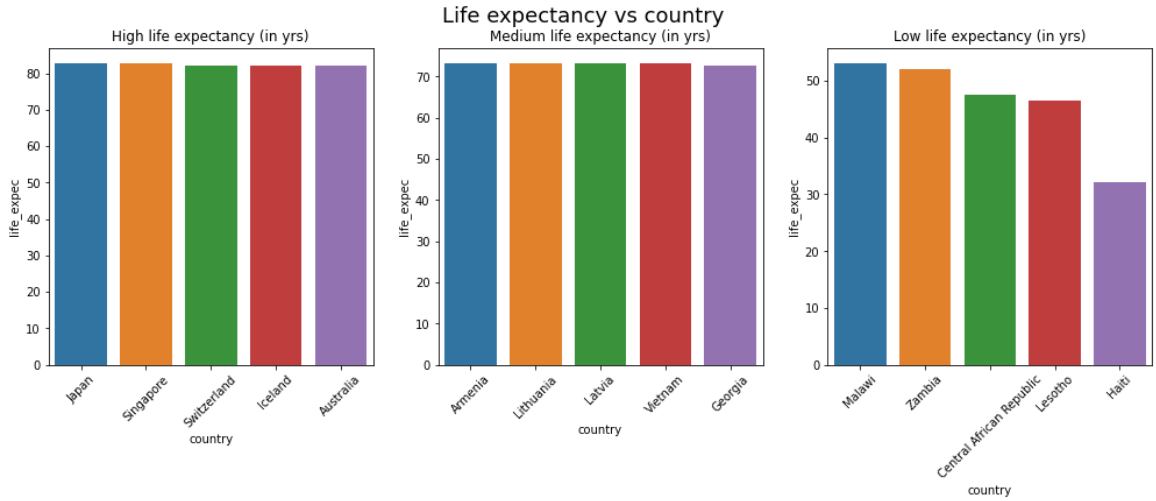


- The expenditure on healthcare by countries are a key factor in economic growth. **Proper healthcare system and services**, leads to a **healthy human capital** and hence **higher income per person**.
- **3 out of 5 countries** in the **bottom** for expenditure in healthcare are in **Asia/Middle East**.
- **Qatar** is at the **bottom** for **spending in healthcare**. On the other hand, it is the **leading country** in **net income** earned per person.
- **United States** is the **leading country** with regards to **spending in healthcare**, which is around **18%** of its GDP.

Life Expectancy

In [20]:

```
title = 'Life expectancy vs country'
plots_catogs('life_expec', title, 'life expectancy (in yrs)')
plt.show()
```

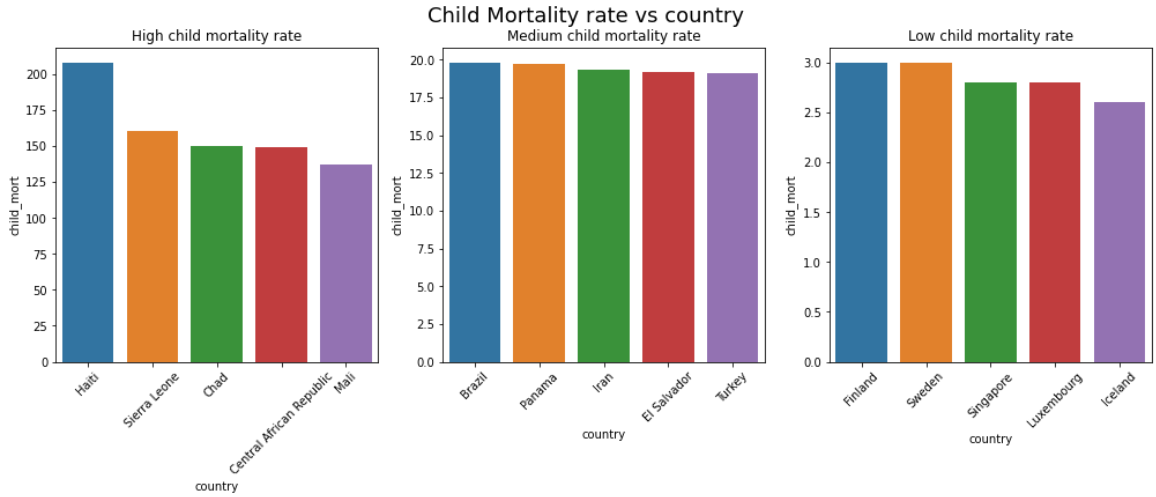


- Life expectancy has a correlation with economic growth in a country. **Higher life expectancy** has **causation** with **greater standards of living** and **proper healthcare infrastructure** in the country.
- **Higher life expectancies** are found in **countries in Asia and Europe**, with Japan and Singapore few of them. People in wealthier countries can afford proper living standards which acts as a factor in greater Life Expectancy.
- In the **bottom 5 countries**, **all of them** are in **Africa**, with Haiti being an exception.
- **Haiti** has a **life expectancy** as low as **30yrs**.

Child mortality rate

In [21]:

```
title = 'Child Mortality rate vs country'
plots_catogs('child_mort', title, 'child mortality rate')
plt.show()
```

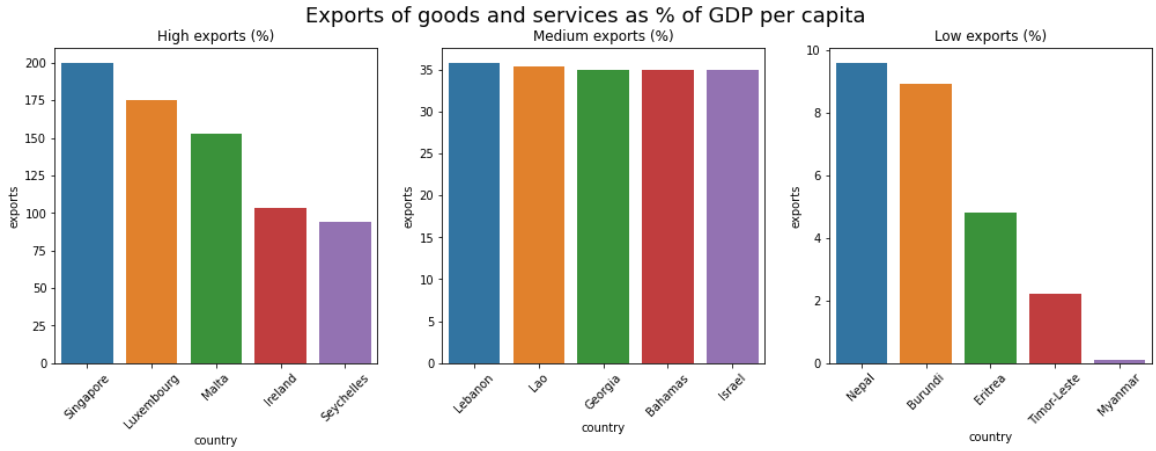


- **Child mortality rate** indicate the **state of the country's healthcare system** and the **government schemes** employed for **child health and welfare**. Greater expenditure in healthcare, leads to better conditions and facilities and hence lowers the child mortality rate.
- All countries except **Haiti** in the **top 5** for child mortality are from **Africa**. This clearly outlines the poor condition of the healthcare system for children in the countries. In addition, Africa also lies in the bottom for Life Expectancy as per previous plot.
- **Haiti** has the highest child mortality rate and as per the previous plot, it also has the lowest Life Expectancy among all countries.
- **Singapore** and **other European countries** like **Luxembourg** etc. lie in the **lower extremes in child mortality**.

Exports of goods and services as % of GDP per capita

In [22]:

```
title = 'Exports of goods and services as % of GDP per capita'
plots_catogs('exports', title, 'exports (%)')
plt.show()
```

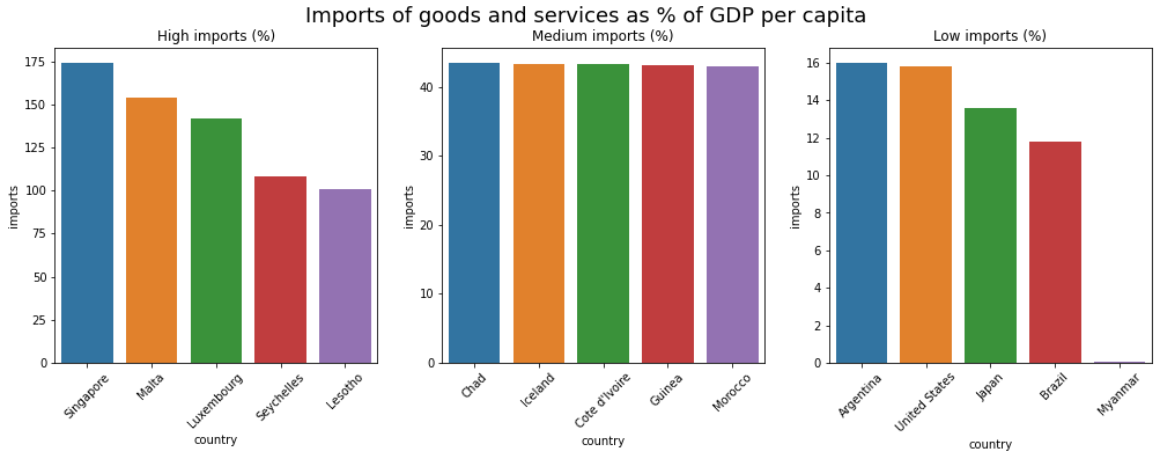


- **Higher amount of exports** from a country **positively impact** the **economic growth**. There is a **trade surplus** and **more industrial output** in the domestic sector, which can imply more employment for the people.
- Amount of exports and import trades carried out by a country depend on the geographical location, availability of resources and the various other factors.
- **Singapore**, **Malta** and **Luxembourg** are in the **top 3** for the **Exports** of goods and services.
- Countries having low exports earn less revenue and have less foreign exchange reserves.
- **Myanmar** is at the **bottom** with a meagre **0.1% of its GDP** comprising of exports of its goods and services.
- **Nepal** is in the **bottom 5** for exports of its goods and services. The reason could be the landlocked geography of Nepal, which limits its trades with other countries.

Imports of goods and services as % of GDP per capita

In [23]:

```
title = 'Imports of goods and services as % of GDP per capita'
plots_catogs('imports', title, 'imports (%)')
plt.show()
```

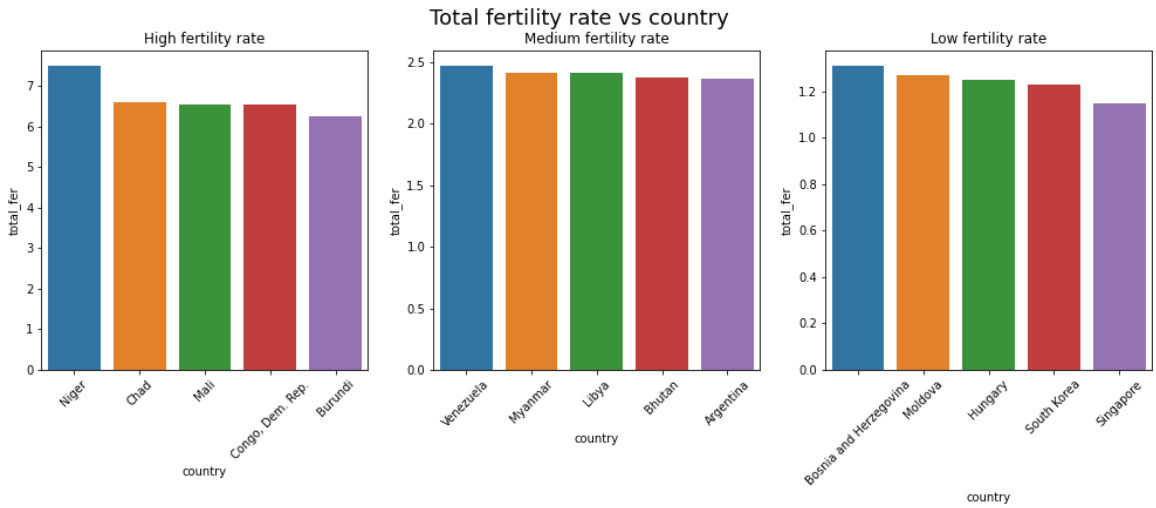


- **Increase in imports of goods** relate to **greater domestic demand**. If imports exceed the exports, then there is a trade-deficit.
- **Singapore** has the **highest imports** as % its GDP, with it being closely **followed by Malta** and **Luxembourg**.
- Countries like **United States** and **Japan**, being one of the world's largest economies, are in the **bottom 5** for imports of goods and services.
- **Myanmar** is the one country which has **lowest export and import %** of its GDP per capita. This can be **attributed to poor economic growth**.

Total fertility rate

In [24]:

```
title = 'Total fertility rate vs country'
plots_catogs('total_fer', title, 'fertility rate')
plt.show()
```

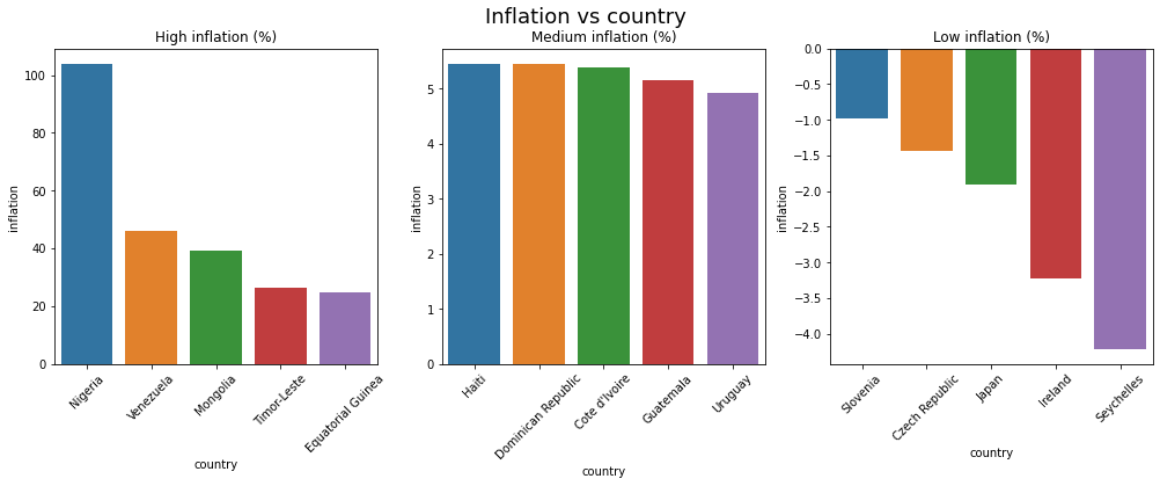


- Population growth in a country is strongly influenced by the fertility rate.
- All the countries with High Fertility Rate come from **Africa**.
- At the other extreme, **Singapore** has the lowest rate around 1.2. It is accompanied by South Korea and certain European countries.

Inflation

In [25]:

```
title = 'Inflation vs country'
plots_catogs('inflation', title, 'inflation (%)')
plt.show()
```



- Inflation gives a broad measure of the rise in prices of goods and services. When inflation increases, the purchasing power of consumers drops and hence GDP consequently decreases.
- On the other hand, negative values for the same indicate a case of '**Deflation**'. In this scenario, consumers do not show urgency in making purchases, hence is no economic activity occurring. This implies less revenue for producers and low economic growth.
- **Nigeria** has the highest inflation rate of more than double then the next country Venezuela.
- Interesting to note is that **Japan** alongwith certain European countries, show deflation in their economic growth. **Seychelles** howers around -4% deflation.

EDA Summary

- **Singapore, Malta and Luxembourg** are in the **top 5** countries for **Imports and Exports** of goods and services. Singapore, specifically shows a trade surplus as noted from the data.
- **Majority of countries** having **high child mortality and fertility rates** come from **Africa**.
- In addition, **Singapore** lies in the **bottom** for **child mortality** and **fertility**, which could indicate better healthcare infrastructure and government policies.
- In **healthcare spending**, **United States** leads with **18%** of its GDP set aside for the same. On the other hand, **countries** in **Asia/Middle East** have spent only a **maximum of 2.5% of their GDP** towards **healthcare**.
- **European** and **Asian countries** like Japan and Singapore are in the **top 5** for **life expectancy**, with it being approximately 80yrs. However, **African countries** and **Haiti** are at the **bottom extreme**.
- In **GDP per capita** growth rates, **majority** of the **countries** at the **top** end are in **Europe** with **Luxembourg leading at USD120000**. **Qatar** is **only Asian country** in this list.
- For **net income earned per person**, **majority** of **countries** come from **Middle East**, with Qatar at the top. At the **other extreme**, **African nations occupy** with a person **earning USD700 on average approximately**.

What factors decide which country is in need of financial aid?

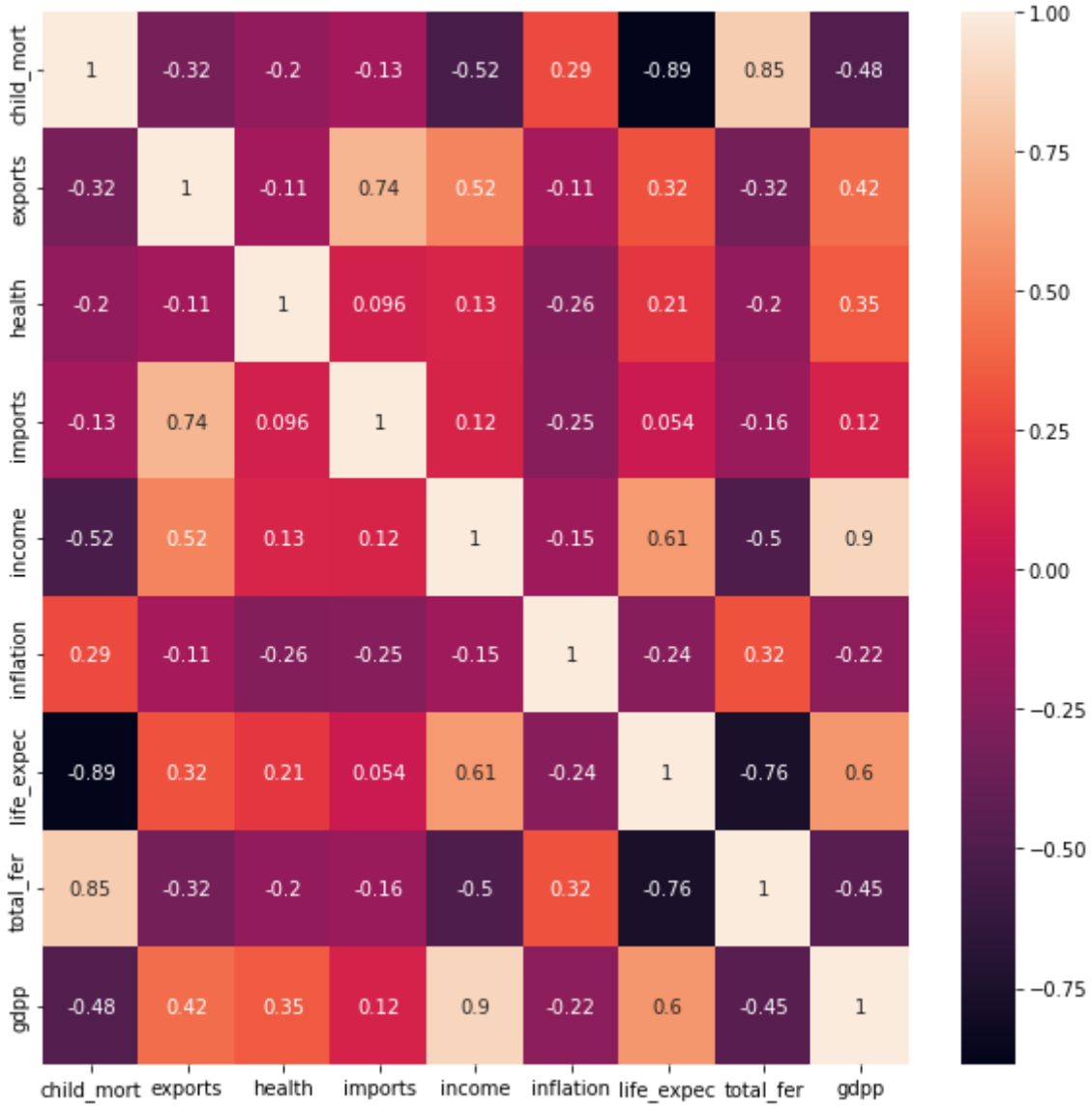
Countries which are economically backward require a financial aid to help them in getting to a normal level of economic growth. Below are the factors which define an economically backward country.

- **Per capita income is lower** in comparison to developed countries
- **High birth** and **death rates**
- **Lack of expenditure in healthcare** than developed nations leading to poorer health infrastructure and standards of living
- **Rapid growth in population** causing a problem when making resources available to every citizen of the country
- **Unemployment** prevails in these countries due to the lack of viable resources.
- Due to **low wealth**, they experience a drop in their capital
- The **distribution of wealth** and **income is unequal**.

Correlation matrix for the features

In [26]:

```
corr = country_data.corr()
plt.figure(figsize = (10,10))
sns.heatmap(corr, annot = True)
plt.show()
```



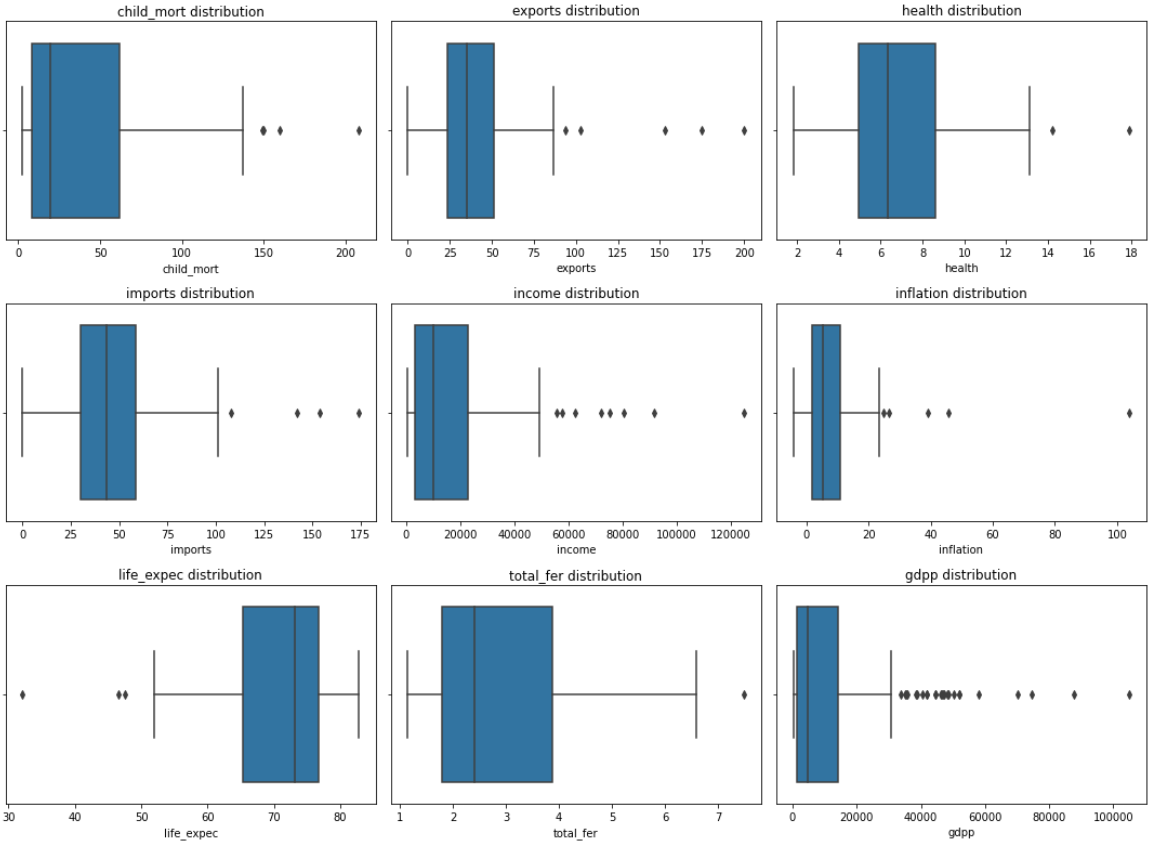
- Fertility rate and child mortality show 85% of positive correlation, which agrees with the theory. In addition, Life Expectancy has a strong negative correlation with child mortality.
- Imports and Exports have a 74% positive correlation in this data.

Managing outliers

Outliers have significant effect on the performance of KMeans Clustering.

In [27]:

```
# boxplots
fig, ax = plt.subplots(nrows = 3, ncols = 3, figsize = (15,11))
for i in range(len(numerical_cols)):
  plt.subplot(3,3,i+1)
  sns.boxplot(country_data[numerical_cols[i]])
  title = numerical_cols[i] + ' distribution'
  plt.title(title)
plt.tight_layout()
plt.show()
```



- The distribution of GDP per capita show significant outliers above USD30000. This means there quite a few countries with GDP per capita higher than the normal average.
- There are 25 countries exceeding the 99th percentile value for gdpp.
- Income, Inflation, Imports and Exports also show few outliers which will need to be suppressed.

In [28]:

```
data_copy = country_data.copy()
```

Suppressing outliers using the 'gdpp' column

In [29]:

```
# Suppressing the outliers
indexes = np.where(country_data['gdpp']>32000)
print('Number of outliers for gdpp = ', len(indexes[0]))
# display(country_data.iloc[indexes[0],:].sort_values('gdpp', ascending = True))
```

```
Number of outliers for gdpp =  25
```

In [30]:

```
# Dropping the rows containing outliers in 'gdpp'
country_data.drop(indexes[0], axis = 0, inplace = True)
country_data.shape
```

Out[30]:

```
(142, 10)
```

Scaling the data

In [31]:

```
country_data.describe()
```

Out[31]:

	child_mort	exports	health	imports	income	inflation	life_expec
count	142.000000	142.000000	142.000000	142.000000	142.00000	142.000000	142.000000
mean	44.112676	37.970415	6.424014	46.598351	10935.65493	8.599683	68.806338
std	41.038923	21.839432	2.369044	21.142601	9895.15270	11.112812	8.490539
min	3.200000	0.109000	1.970000	0.065900	609.00000	-4.210000	32.100000
25%	13.900000	22.800000	4.872500	31.325000	2715.00000	2.342500	62.825000
50%	26.300000	33.050000	5.990000	44.100000	7940.00000	5.935000	70.450000
75%	63.850000	50.200000	7.887500	58.825000	16150.00000	11.975000	75.475000
max	208.000000	153.000000	14.200000	154.000000	45400.00000	104.000000	81.900000

Why scaling is important?

- Every feature has a different range for the values. For 'income' it goes from 100s to 10000s. You compare that to 'health', which is in the 10s.
- While model fitting, the algorithm might give more weight to features which have higher range of values. This introduces **bias** in model building, which needs to be corrected.

So, for our analysis, StandardScaler is used from sklearn library where the values of all features will be scaled in such a way that the mean and standard deviation become 0 and 1 respectively for each feature.

In [32]:

```
countries = country_data.country
```

In [33]:

```
# Initialising StandardScaler function
scaler = StandardScaler()
country_data_scaled = scaler.fit_transform(country_data.iloc[:,1:])
```

In [34]:

```
country_data_scaled = pd.DataFrame(country_data_scaled, columns = cols[1:])
country_data_scaled.insert(loc = 0, column = 'country', value = countries)
print(country_data_scaled.shape)
```

```
(142, 10)
```

In [35]:

```
country_data_scaled.head()
```

Out[35]:

	country	child_mort	exports	health	imports	income	inflation	life_expec	t
0	Afghanistan	1.126990	-1.285264	0.489682	-0.080613	-0.945783	0.075885	-1.490007	1
1	Albania	-0.672778	-0.458149	0.053368	0.095009	-0.101991	-0.371124	0.885714	-0
2	Algeria	-0.411127	0.019740	-0.954812	-0.721394	0.199219	0.677314	0.909353	-0
3	Angola	1.831247	1.117964	-1.513971	-0.175543	-0.510703	1.246234	-1.029046	1
4	Antigua and Barbuda	-0.826834	0.345991	-0.166906	0.583901	0.828006	-0.646553	0.944811	-0

Model Building - KMeans Clustering

In [36]:

```
from sklearn.cluster import KMeans
```

Finding Optimal Value for k - Elbow Curve

Before we apply clustering algorithm, the optimal number of clusters needs to be determined. This is facilitated through an Elbow Curve Plot.

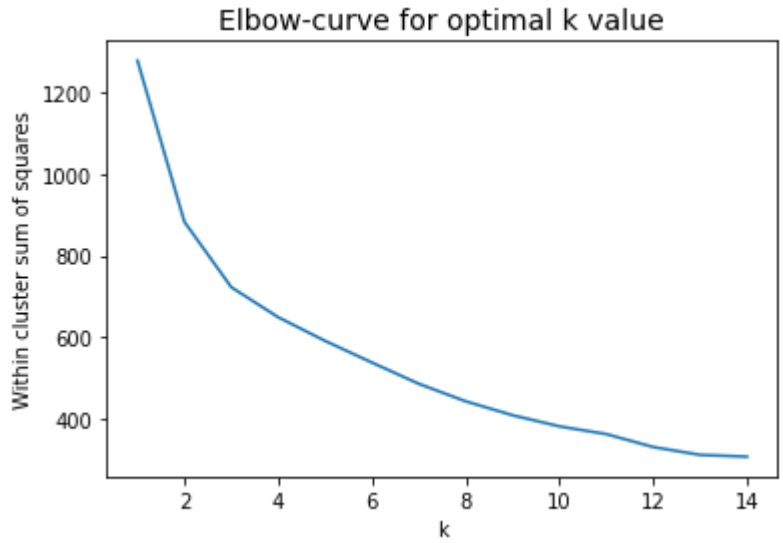
In [37]:

```
k_vals = np.arange(1,15)
wss = []
for i in k_vals:
  kmodel = KMeans(n_clusters = i)
  kmodel.fit(country_data_scaled.iloc[:,1:])
  wss.append([i,kmodel.inertia_])

print(wss)
```

```
[[1, 1278.0], [2, 883.0509474282311], [3, 722.8234526828812], [4, 64
9.191958581152], [5, 591.3044030844285], [6, 538.1037324071744], [7,
486.06508345268355], [8, 443.1693787898646], [9, 409.0567376372268],
[10, 381.82415014268247], [11, 362.7977381680502], [12, 331.13940471
58648], [13, 311.9051575120275], [14, 307.6044484459786]]
```

In [38]:

```
wss = pd.DataFrame(wss, columns = ['k', 'WSS'])

sns.lineplot(x = 'k', y = 'WSS', data = wss)
plt.ylabel('Within cluster sum of squares')
plt.title('Elbow-curve for optimal k value', fontsize = 14)
plt.show()
```



- At k=3, an elbow shape can be identified. The slope of the curve changes rapidly from k=3. Therefore, k=3 is selected as the optimum value for the number of clusters.

Model fitting - using the optimal k value

In [39]:

```
kmodel_new = KMeans(n_clusters = 3)

kmodel_new.fit(country_data_scaled.iloc[:,1:]) # the first column 'country' is l
eft out as Euclidean distance method will not work with string values
pred_labels = kmodel_new.predict(country_data_scaled.iloc[:,1:])
# print(len(pred_labels))
```

In [40]:

```
# Adding the cluster labels column to the original data
country_data['cluster'] = pred_labels
country_data.reset_index(drop = True, inplace = True)
display(country_data.head())
```

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	g
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12

The cluster labels are predicted for each country in the scaled data.

Exploring the characteristics of each cluster

In [42]:

```
display(country_data.cluster.value_counts())
```

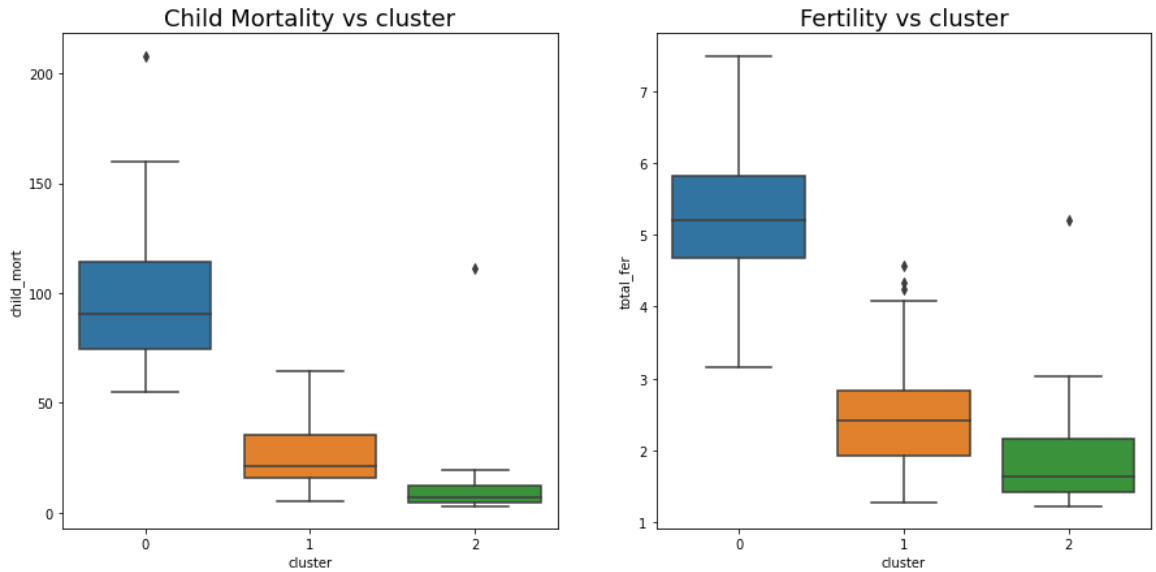
```
1    73
0    41
2    28
Name: cluster, dtype: int64
```

- Out of the 142 countries, almost 50% are in cluster 2.

Child mortality and Fertility vs cluster

In [43]:

```
fig, ax = plt.subplots(1, 2, figsize = (15,7))

plt.subplot(1,2,1)
sns.boxplot(x = 'cluster', y = 'child_mort', data = country_data)
plt.title('Child Mortality vs cluster', fontsize = 18)

plt.subplot(1,2,2)
sns.boxplot(x = 'cluster', y = 'total_fer', data = country_data)
plt.title('Fertility vs cluster', fontsize = 18)
plt.show()
```

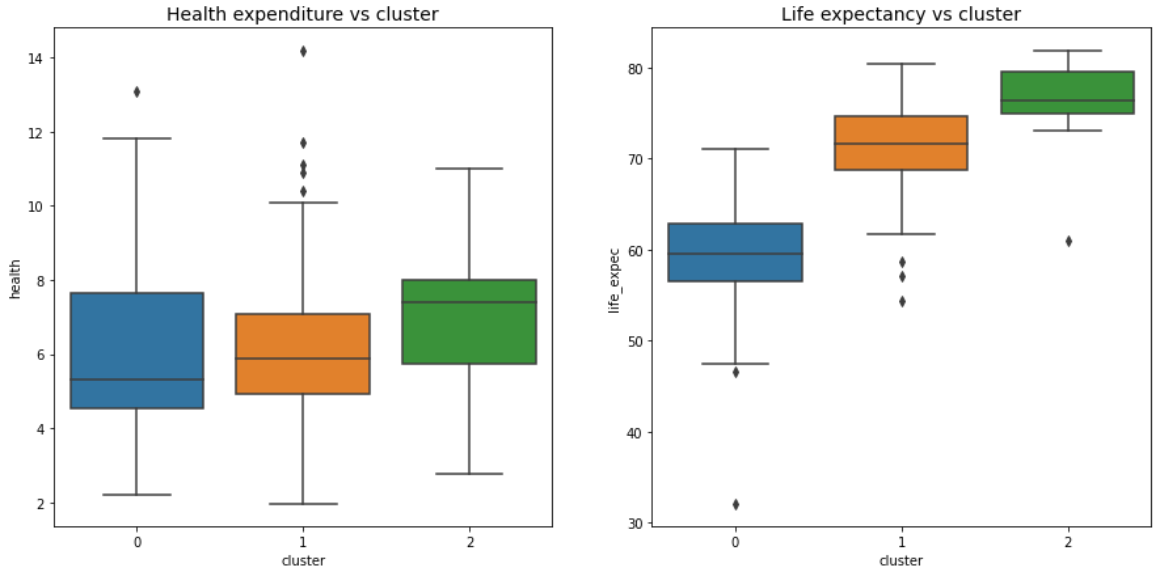


- Cluster 0 has countries with greater child mortality and fertility rates.
- Cluster 2 on the other hand has the lowest average of child mortality and tota fertility rates.

Health expenditure and Life expectancy vs cluster

In [44]:

```
fig, ax = plt.subplots(1, 2, figsize = (15,7))

plt.subplot(1,2,1)
sns.boxplot(x = 'cluster', y = 'health', data = country_data)
plt.title('Health expenditure vs cluster', fontsize = 14)

plt.subplot(1,2,2)
sns.boxplot(x = 'cluster', y = 'life_expec', data = country_data)
plt.title('Life expectancy vs cluster', fontsize = 14)
plt.show()
```

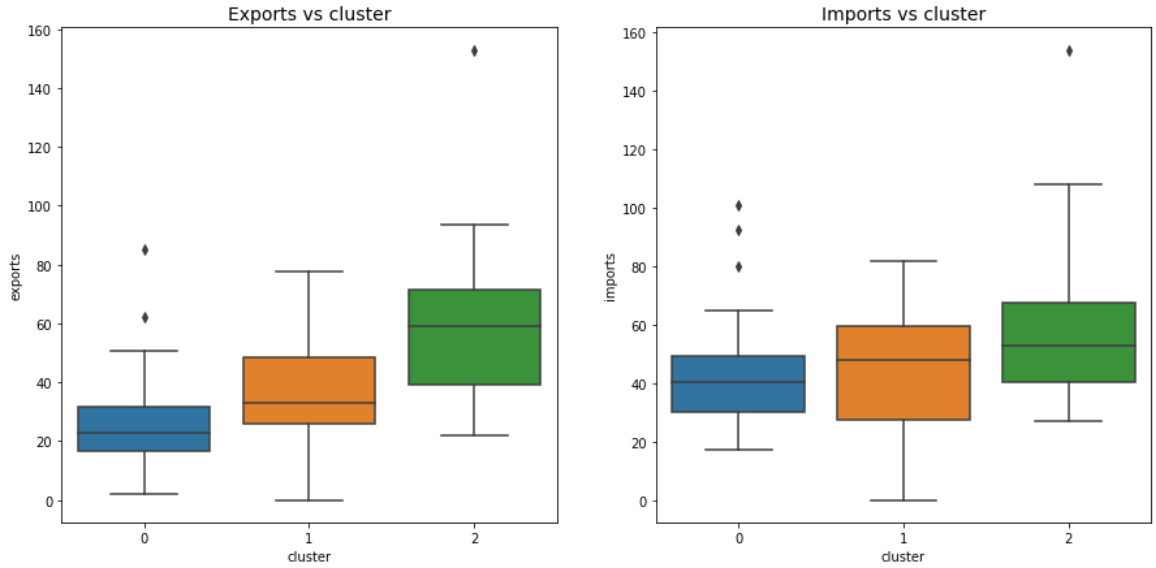


- Health expenditure and life expectancy are the lowest in cluster 0.

Exports and Imports vs cluster

In [45]:

```
fig, ax = plt.subplots(1, 2, figsize = (15,7))

plt.subplot(1,2,1)
sns.boxplot(x = 'cluster', y = 'exports', data = country_data)
plt.title('Exports vs cluster', fontsize = 14)

plt.subplot(1,2,2)
sns.boxplot(x = 'cluster', y = 'imports', data = country_data)
plt.title('Imports vs cluster', fontsize = 14)
plt.show()
```

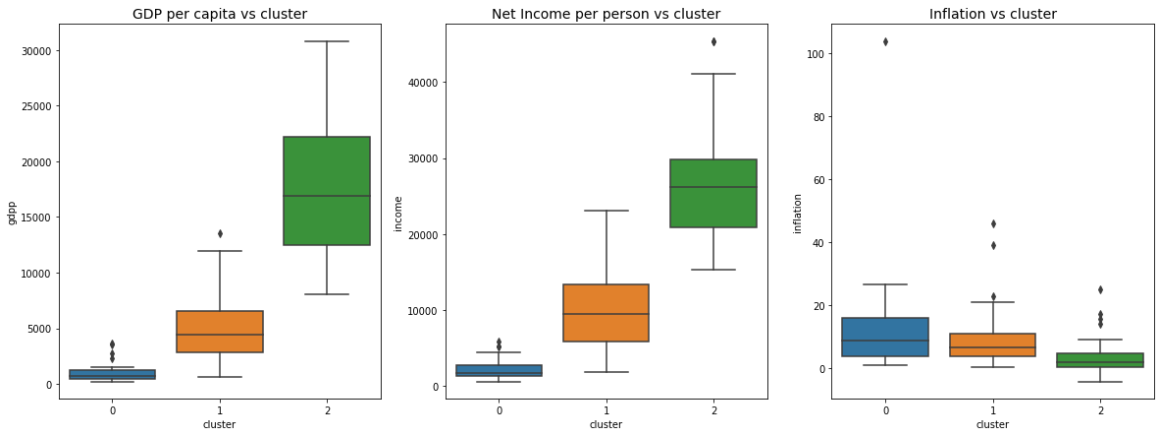


- Cluster 0 shows lower exports and imports in comparison to the rest of the clusters.

GDP per capita, Net Income per person and Inflation vs cluster

In [46]:

```
fig, ax = plt.subplots(1, 3, figsize = (20,7))

plt.subplot(1,3,1)
sns.boxplot(x = 'cluster', y = 'gdpp', data = country_data)
plt.title('GDP per capita vs cluster', fontsize = 14)

plt.subplot(1,3,2)
sns.boxplot(x = 'cluster', y = 'income', data = country_data)
plt.title('Net Income per person vs cluster', fontsize = 14)

plt.subplot(1,3,3)
sns.boxplot(x = 'cluster', y = 'inflation', data = country_data)
plt.title('Inflation vs cluster', fontsize = 14)
plt.show()
```



- Finally, the GDP per capita and Net Income per person ranks the lowest for cluster 0 while it leads in Inflation.

Lets create a table which shows the mean values of all features for each cluster

In [47]:

```
averages_of_all = pd.DataFrame(country_data.groupby('cluster').agg({'gdpp':'mea
n', 'income':'mean', 'inflation':'mean', 'exports':'mean', 'imports':'mean',\
                                    'health':'mean', 'child_mort':'mean', 'life
_expec':'mean', 'total_fer':'mean'}))

display(averages_of_all.sort_values(by = ['child_mort', 'total_fer', 'health',
'life_expec', 'income'],\
                                    ascending = [False, False, True, True, Tru
e]))
```

cluster	gdpp	income	inflation	exports	imports	health	child_mort	lif
0	1004.439024	2188.609756	11.902317	26.029512	42.387805	6.337805	97.448780	5
1	4823.041096	9799.041096	8.485055	36.365603	44.881725	6.271644	26.517808	7
2	18112.500000	26707.142857	4.062536	59.639286	57.239286	6.947500	11.885714	7

Therefore we can conclude that Cluster 1 is going to be the target group of countries who are require the financial aid.

- Cluster 0 -> Require financial aid
- Cluster 1 -> May require financial aid
- Cluster 2 -> Do not require financial aid

In [54]:

```
test_copy = country_data.copy()
```

Visualizing the countries who need financial aid on world map

In [49]:

```
import plotly.express as px
!pip install -U kaleido
import kaleido
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.d
ev/colab-wheels/public/simple/
Requirement already satisfied: kaleido in /usr/local/lib/python3.8/d
ist-packages (0.2.1)
```

In [57]:

```
country_data['cluster'].loc[country_data['cluster'] == 0] = 'Help Needed'
country_data['cluster'].loc[country_data['cluster'] == 1] = 'May Need Help'
country_data['cluster'].loc[country_data['cluster'] == 2] = 'Help Not Needed'
```

In [58]:

```
fig = px.choropleth(country_data[['country','cluster']],
                    locationmode='country names',
                    locations='country',
                     title='Needed Help Per Country (World)',
                     color_discrete_sequence=["orange", "red", "green",'black'],
color=country_data['cluster'],
                     color_discrete_map={'Help Needed':'Red',
                                         'May Need Help':'Yellow',
                                         'Help Not Needed':'Green'} )
fig.update_geos(fitbounds="locations", visible=True)
fig.update_layout(legend_title_text='Labels',legend_title_side='top',title_pad_l
=260,title_y=0.86)

#fig.write_html("NeededHelpPerCountry(World)kmeans.html")
#fig.write_image("NeededHelpPerCountry(World)kmeans.png",scale=3)
fig.show(engine='kaleido')
```

Conclusion

- From the above visualization, we find majority of countries needing financial aid are concentrated in Central Africa. There are few countries in Asia and Middle East who also fall in this category.
- Majority of countries in Asia and South America may require a financial aid. However, certain Middle East countries like Oman, Saudi Arabia and some European nations do not require the financial help.
- There are other countries like United States of America, Australia, United Kingdom etc greyed out. This could be due to suppressing the outliers in the data.
- The **Exploratory Data Analysis** carried out has shown important insights and trends in the socio-economic factors.
- **Standardization** or normalization of the data has major effect in terms of the model performance. This step is important before feeding the data to the model.
- This problem statement required **Unsupervised Learning**, and for my analysis I considered only **KMeans Clustering** as the algorithm. Other clustering techniques like Hierarchical Clustering etc. can be used, which might give better accuracy.

References

- For the plotly world map, I had referred the Kaggle notebook: Username - TANMAY DESHPANDE (https://www.kaggle.com/code/tanmay111999/clustering-pca-k-means-dbscan-hierarchical/notebook)

Would be grateful for any feedback and comments on my work! Thank you