Literature Review

In order to formulate this problem, three different data sources have been made use of. All of these sources are public and have been allowed the communal usage of by the authors. The details of these sources are in the order of information about the source, what the data represents and the various key data types used and are as follows:

- **WEF_GGGR_Dataset_2016**
  This data is made available by the World Economic Forum. It gives different indexes as well as the ranks for the global gender gap for the year of 2016, as mentioned in the report itself the idea can be briefly surmised that it:
  "assesses countries on how well they are dividing their resources and opportunities among their male and female populations, regardless of the overall levels of these resources and opportunities"

  The data encompasses both proprietary and non-proprietary data - the proprietary includes the overall index as well as the subindexes and is published under a Creative Commons' Attribution Non-Commercial 4.0 International type of licence (CC BY-NC 4.0). Commercial use of this data is prohibited.

  The following are the details about the columns in this data:

Sr. no.	Column name		Data type	Column Desc.
1	Country		Text	Name of the country for which the different indices are provided
2	Overall Index	Rank	Numeric	The country rank for this index
		Score		An aggregate score over all the indices provided
3	Economic Participation and Opportunity Subindex	Rank		The country rank for this index
		Score		The country's score for global economic participation and providing opportunities to the general public in terms of skilled employment as well as salaries
4	Educational Attainment Subindex	Rank		The country rank for this index
		Score		The country's score for the overall acquisition of education, both basic and higher levels.

5	Health and Survival Subindex	Rank		The country rank for this index
		Score		The country's score for overall medical facilitation and general health wellbeing of the public as well as life expectancy
6	Political Empowerment Subindex	Rank		The country rank for this index
		Score		The country's score for roles in institutions manifesting decision making

- **Proportion of seats held by women in national parliaments**
  This data is provided by the world development indicators.
  The provision source is Inter-Parliamentary Union (IPU).
  The data-set contains the number of seats in custody of women as a percentage of the total number of the seats in the national parliament. This proportion is enlisted separately against different years.
  The details are mentioned in the table below:

Sr. no.	Column name	Data type	Column Desc.
1	Country Name	Text	The name of the country
2	Country Code		A three letter acronym shorthand that uniquely identifies the country
3	Indicator Name		The criteria being looked at for each country, In this file it's consistent for all the countries i.e '*Proportion of seats held by women in national parliaments (%)*'
4	Indicator Code		A shorthand identifier for the Indicator Name
5	1960 . . . 2018	Numeric	The years 1960 through 2018, containing the proportion of the seats held by women in the national parliament, for each country

- **GDP**
  This data is also compiled by the world development indicators and contains the GDP for the year of 2017 ($ US million) for different countries.
  The schema of the data is as follows:

Sr. no.	Column name	Data type	Column Desc.
1	Country code	Text	A three letter acronym shorthand that uniquely identifies the country
2	Ranking	Numeric	The Rank of the country in terms of the absolute magnitude of its GDP
3	Country	Text	The name of the country
4	(millions of US dollars)	Numeric	GDP scaled by $ US in millions

Exploratory data analysis

Global Gender Gap data

We start by exploring the gender gap data first.

```
[ ]  #about the dataset lies at the first key, the data in the second
     gender_gap_df_dict = pd.read_excel('/content/drive/My Drive/maryam/WEF_GGGR_Dataset_2016 (1).xlsx', sheet_name=None)

[ ]  # Saving the keys as consts so as to avoid duplication of these consts in the code

     # about this dataset on the zeroth index
     about = get_dict_key_at_idx(gender_gap_df_dict, 0)

     # GGGR 2016 Data on the first index
     data = get_dict_key_at_idx(gender_gap_df_dict,1)
```

The excel file is loaded into a pandas dataframe, which is just a tabular data structure, much like the file itself. The sheet names are stored in the keys ‘about’ and ‘data’ where they can be taken to mean the following:

- ‘About’ : the metadata of the file
- ‘Data’ : the actual gender gap file data itself

After loading the data we view it to get an idea of what it looks like. Shown below is the data

```
[ ] gender_gap_df_dict[data]
```

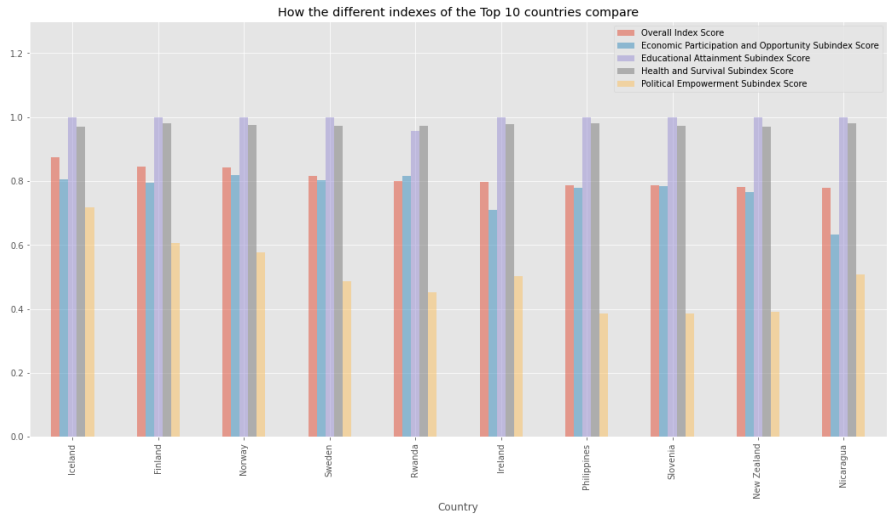
Country	Overall Index Score	Overall Index Rank	Economic Participation and Opportunity Subindex Score	Economic Participation and Opportunity Subindex Rank	Educational Attainment Subindex Score	Educational Attainment Subindex Rank	Health and Survival Subindex Score	Health and Survival Subindex Rank	Political Empowerment Subindex Score	Political Empowerment Subindex Rank
Iceland	0.87367	1	0.806184	9	1	1	0.969801	104	0.718696	1
Finland	0.845047	2	0.793658	16	1	1	0.979558	1	0.606971	2
Norway	0.841951	3	0.818156	7	0.999801	28	0.974075	68	0.575774	3
Sweden	0.815413	4	0.802349	11	0.999233	36	0.973884	69	0.486186	6
Rwanda	0.799611	5	0.816934	8	0.95783	110	0.972313	89	0.452167	8
...	...	...	...	...	...	...	...	...	...	...
Chad	0.586693	140	0.667051	74	0.618352	144	0.968133	111	0.0932349	111
Saudi Arabia	0.582871	141	0.328045	142	0.960768	105	0.965882	128	0.0767906	121
Syria	0.567311	142	0.273224	144	0.963289	103	0.970179	101	0.0625513	130
Pakistan	0.555905	143	0.319604	143	0.810829	135	0.96664	124	0.126549	90
Yemen	0.516217	144	0.351792	141	0.719745	141	0.966842	122	0.0264871	139

144 rows × 10 columns

read from the file into the dataframe:

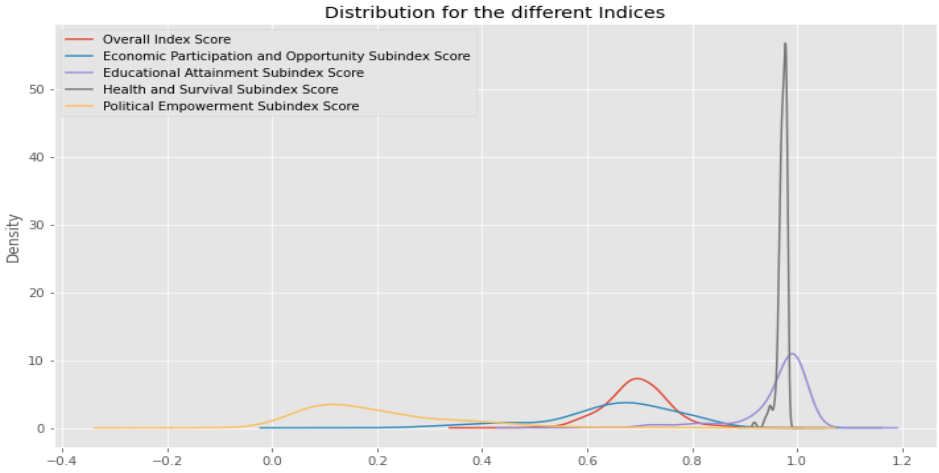
As we can see the indexes mentioned above in the schema and their corresponding values for many countries- but just to get a sense of how these indexes are for the top ten countries we plot a barchart as follows:

```
[ ] title = 'How the different indexes of the Top 10 countries compare'
    ax = gender_gap_df_dict[data][score_cols].head(10).plot.bar(figsize=(18, 9), title = title, alpha=0.5)
    ax.set_ylim(0,1.3)
```



From the above chart for the top ten countries with the highest overall index, we can observe that all of them have very high ‘Educational Attainment Subindex’ as well as ‘Health and Survival Subindex’ scores, ‘Economic Participation and Opportunity Subindex’ scores seem to be more or less consistent while ‘Political Empowerment Subindex Score’ seems to be a good discriminating factor here as we can see it going down from the most ranked country to the least.

As we know that there are many indices in this data, we might also be interested in seeing their distribution. The figure below shows the distribution of these different indexes and how they are dispersed:



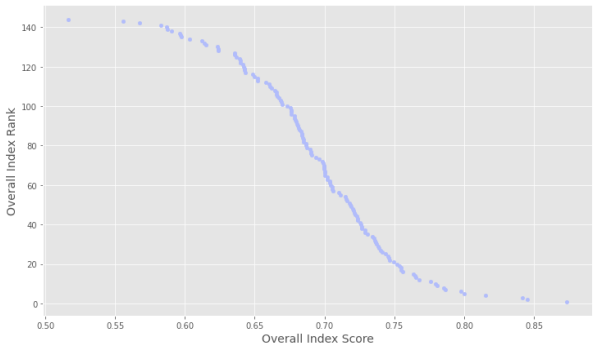
Some observations about the distribution of these indices:

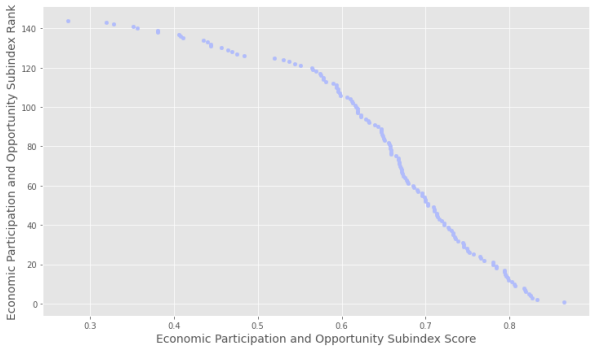
- Overall as well as Educational Attainment index scores seem close to normally distributed.
- Health and Survival Subindex scores have very less variation and are centered to a value close to 1.
- Political Empowerment Subindex score has greater spread and has smaller values on average

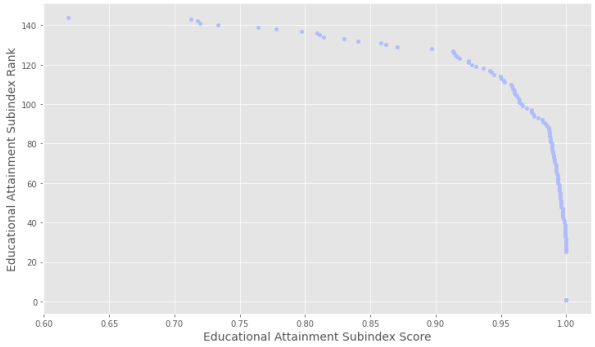
Next, we might want to observe how these indices vary with respect to their corresponding ranks. The snippet below illustrates the plotting of these separately:

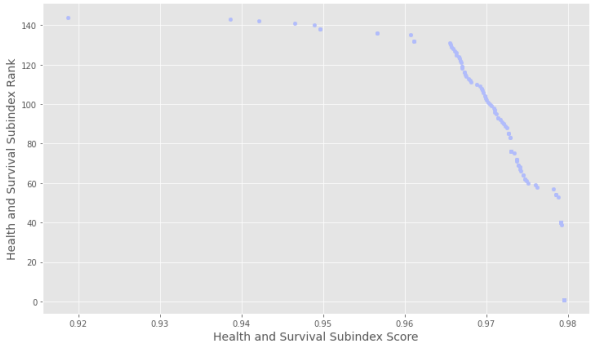
```
#Lets see how the different indexes vary with the ranks of different countries

for x_col, y_col in zip(score_cols, rank_cols):
  gender_gap_df_dict[data].plot.scatter(x=x_col,y=y_col, figsize = (12,7), color=colormap[5])
```

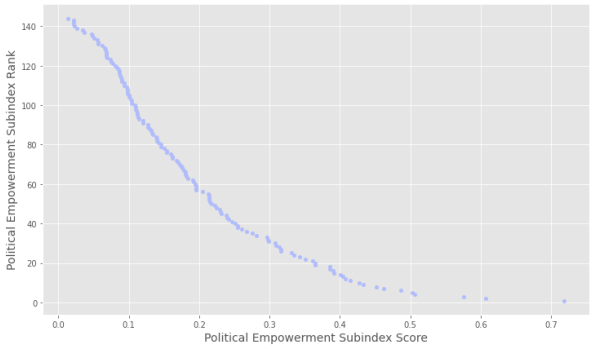
The respective plots are shown below:











Some deductions we can make about these plots can be surmised as:

- The *Overall index* varies with its respective rank in a symmetric fashion. We can see that the curve imitates a sigmoid shape- from that we know that the score for the countries both on the top and bottom of the list have larger differences in the index score on average.
- *Economic Participation and Opportunity* Score changes very rapidly amidst the bottom 20 countries, but changes more or less linearly for the the top 100 countries
- Both *Educational Attainment Subindex* and *Health* and *Survival Subindex Scores* vary very quickly for the bottom 20-30 countries, after which it kind of just hits a ceiling effect.

- Political Empowerment Subindex score seems like a slightly lesser steep exponential growth curve when plotted against its respective ranks. The variance especially seems to be a lot for the top 20 countries.

We might especially be interested in how these indices compare with each other, how do they relate- are they in some way or other correlated?
The snippet below demonstrates how we can first calculate the correlation matrix of these indices and then plot them using a heat-map:

```
plt_resize()
sns.heatmap(gender_gap_df_dict[data][score_cols].corr(), cmap="BuPu")
```

The legend bar is the is the scale for the correlation scores and their corresponding color coding

From this heatmap we know the following:

- *Economic Participation and Opportunity* and *Political Empowerment* subindices are strongly positively correlated with the *Overall index*.
- *Education Attainment Subindex* score is also substantially positively correlated with the *overall index*
- The subindices themselves are all mildly positively correlated with each other as well.

Proportion of seats held by women in national parliaments

We'd now load and explore the data for the proportion of seats held by women in national parliaments.
Let's first start by loading again the file, with different sheets next to their respective names. We then store them in a dictionary and keep their names in keys that can be referred to later.

```
#about the dataset lies at the first key, the data in the second
woman_seats_in_national_parliaments_df_dict = pd.read_excel('/content/drive/My Drive/maryam/proportion_of_seats_held_by_women_in_national_parliaments (1).xls', sheet_name=None)

#storing the keys
data_woman_parliament_seats = get_dict_key_at_idx(woman_seats_in_national_parliaments_df_dict,0)
meta_data_countires =  get_dict_key_at_idx(woman_seats_in_national_parliaments_df_dict,1)
meta_data_indicators =  get_dict_key_at_idx(woman_seats_in_national_parliaments_df_dict,2)
```

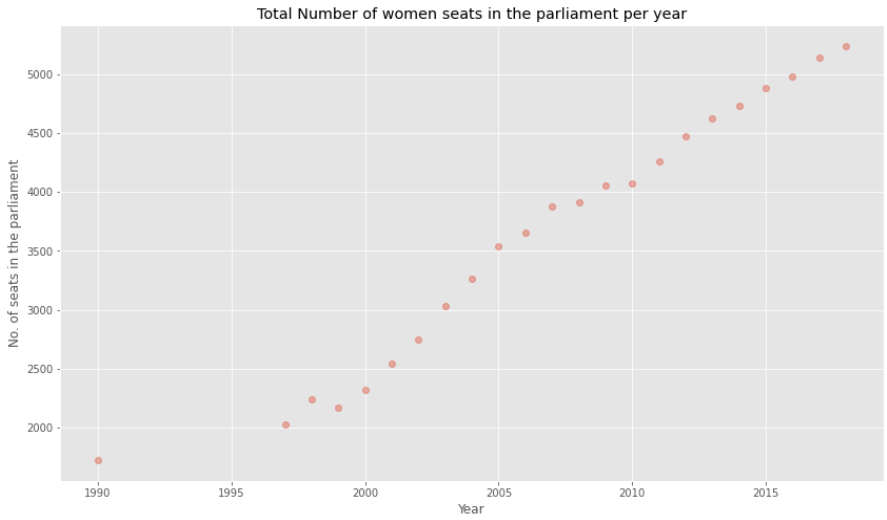
We then segregate the columns by column type so it's easier for us to perform aggregate operations later on.

```
]  woman_seats_in_nat_parl_column_types = get_columns_against_type_dict(woman_seats_in_national_parliaments_df_dict[data_woman_parliament_seats])

]  # making column groups for the woman_seats_in_the_national_parliament dataset
   wsnp_float_cols = woman_seats_in_nat_parl_column_types['float64']
   wsnp_obj_cols = woman_seats_in_nat_parl_column_types['object']
```

We store them as float and object type columns, where object is a Pandas dataframe tool built in generic data type.

Next, we want to see how the proportion of the seats have been changing throughout the years. The code snippet below shows this followed by the chart which further gives an illustration:

```
]  #removing the years with no data i.e NULL columns
   total_worldwide_seats_per_year = woman_seats_in_national_parliaments_df_dict[data_woman_parliament_seats][wsnp_float_cols].sum()

]  plt.title("Total Number of women seats in the parliament per year")
   plt.xlabel("Year")
   plt.ylabel("No. of seats in the parliament")
   total_worldwide_seats_per_year[total_worldwide_seats_per_year > 0].plot(style='o', figsize=(14,8), alpha=0.4)
```



From this chart we can deduce the following points:

- Even though the file contained years 1960 - 2018, but we only see the plots for years 1990 and 1997-2018. This is because the rest of the years had no data, they had to be treated as columns having missing values and thus were ignored.
- We can see that the total number of seats have been increasing over time. The trend has been more or less linear over time and seems to have slowed down a little from 2006 to 2010, but caught on with the general trend afterwards.

Next, we want to see how the trends for the the number of women seats in the national parliament have been like for the countries but since it's hard to make out these trends for all the countries from a single plot, we'll resort to showing the trends for the list of top 10 countries with consistently the most women representation in the national parliament. A code snippet showing how this was done, followed by the chart illustrates this below:

```
#Top 10 countties with the most female seats in the parliament

top_k_countries_with_the_most_female_rep_in_parl = country_wise_most_parliament_seats[-10:].index.tolist()
```

We further extract the desired columns

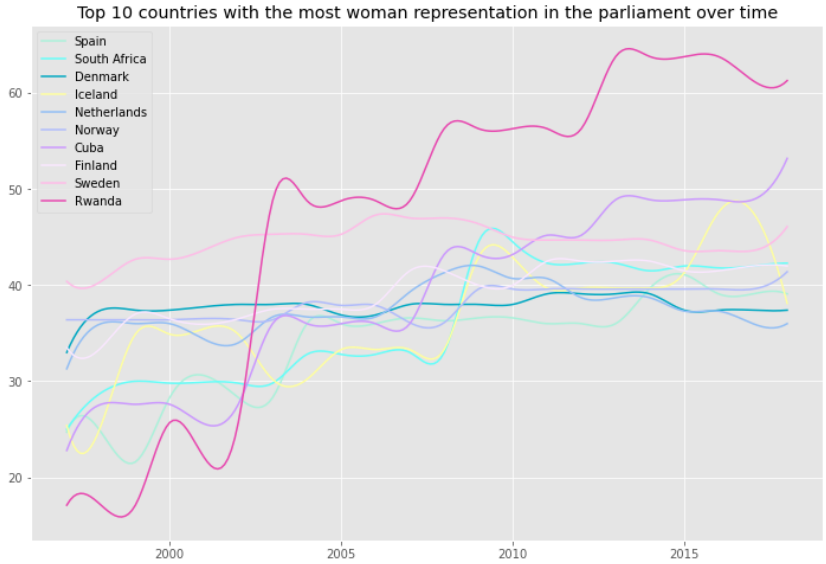
```
top_k_countries_woman_parl_seats_data = woman_seats_in_national_parliaments_df_dict[data_woman_parliament_seats][wsnp_float_cols].T[top_k_countries_with_the_most_female_rep_in_parl]
```

But since this some of these columns have missing values we have to impute them, the snippet below shows how that was done:

```
top_k_countries_woman_parl_seats_data.set_index(top_k_countries_woman_parl_seats_data.index.astype(int), inplace=True)

temp = interpolated_df(top_k_countries_woman_parl_seats_data.loc[1997:], top_k_countries_woman_parl_seats_data.columns, intvl=0.1)

temp  = temp.applymap(lambda v: np.nan if v < 0 else v)
```

And now to plot this data:

```
title = "Top 10 countries with the most woman representation in the parliament over time"
temp.plot(color=[color_cols_mapper.get(x, '#333333') for x in top_k_countries_woman_parl_seats_data.columns], figsize=(12,8), title=title)
plt.show()
```

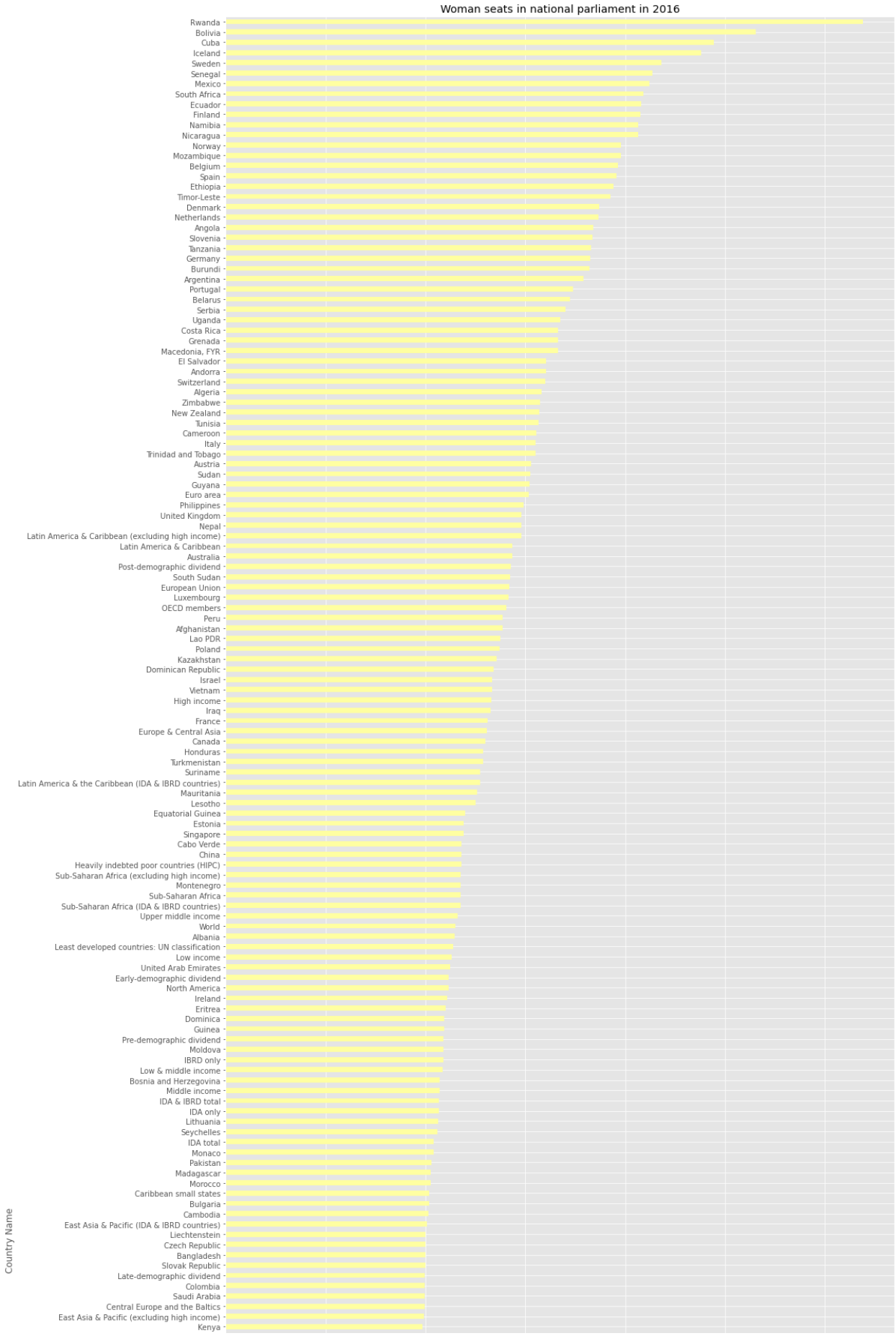


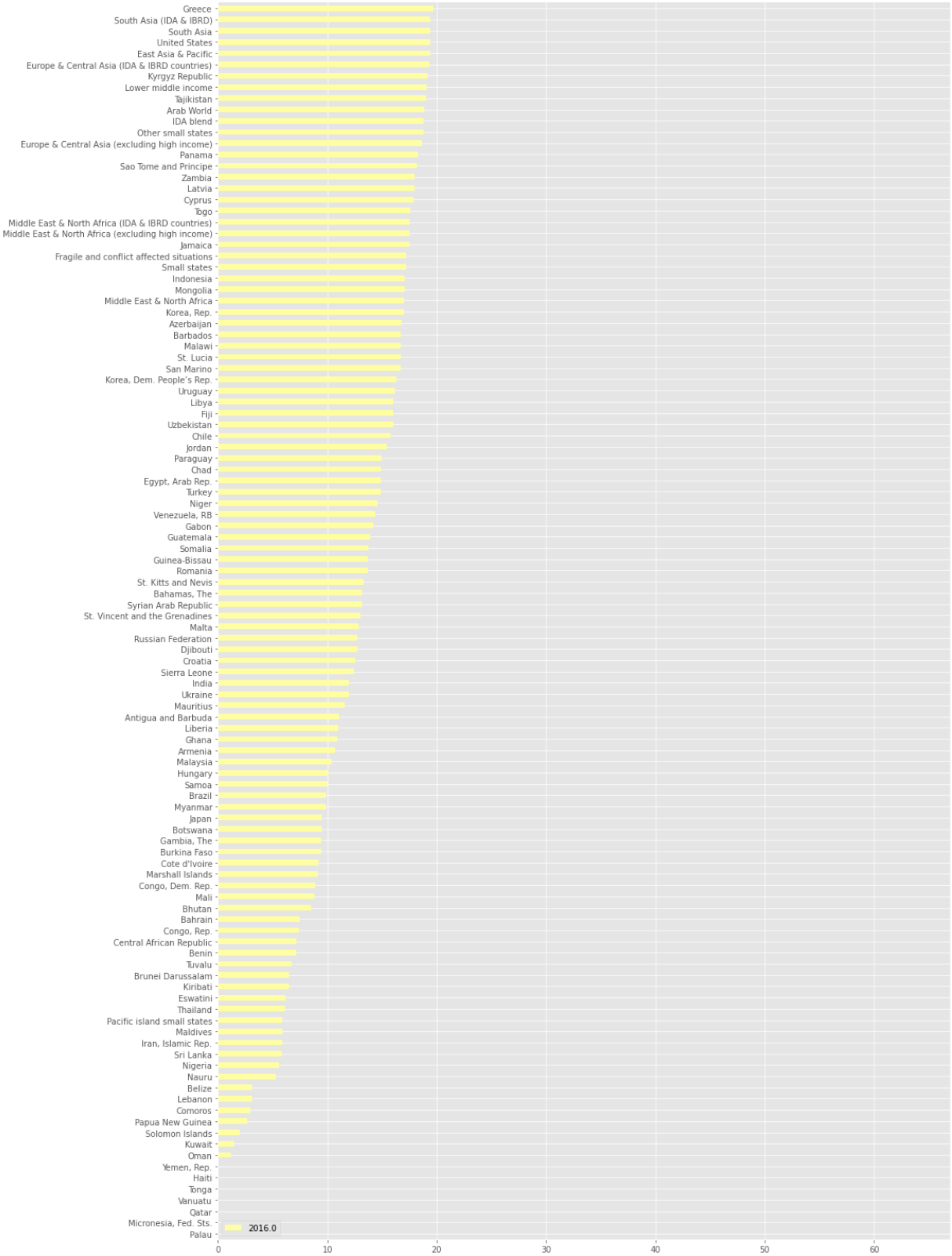
Some observations pertaining this line chart:

- Rwanda saw a huge gain in female representation in the national parliament between 2001-2004 and kept on making progress later as well so much so in 2018, it's still far ahead on the other nine countries being observed.
- Cuba has been a close second in making substantial progress in the female representation in the national parliament.
- Around 2009, a surge in the proportion can be felt amongst all the countries alike.
- The trend for some countries such as Norway and Denmark remains more or less the same over the years

Wondering how these seats might vary for each country from the highest to lowest, we can simply plot a bar chart showing a relative comparison. This snippet below demonstrates how can this be achieved.

```
title = "Woman seats in national parliament in 2016"
woman_seats_in_national_parliament_2016.sort_values(by=2016).plot.barh(figsize=(16,60), color = colormap[3], title=title)
```


Woman seats in national parliament in 2016
Country Name
Rwanda
Bolivia
Cuba
Iceland
Sweden
Senegal
Mexico
South Africa
Ecuador
Finland
Namibia
Nicaragua
Norway
Mozambique
Belgium
Spain
Ethiopia
Timor-Leste
Denmark
Netherlands
Angola
Slovenia
Tanzania
Germany
Burundi
Argentina
Portugal
Belarus
Serbia
Uganda
Costa Rica
Grenada
Macedonia, FYR
El Salvador
Andorra
Switzerland
Algeria
Zimbabwe
New Zealand
Tunisia
Cameroon
Italy
Trinidad and Tobago
Austria
Sudan
Guyana
Euro area
Philippines
United Kingdom
Nepal
Latin America & Caribbean (excluding high income)
Latin America & Caribbean
Australia
Post-demographic dividend
South Sudan
European Union
Luxembourg
OECD members
Peru
Afghanistan
Lao PDR
Poland
Kazakhstan
Dominican Republic
Israel
Vietnam
High income
Iraq
France
Europe & Central Asia
Canada
Honduras
Turkmenistan
Suriname
Latin America & the Caribbean (IDA & IBRD countries)
Mauritania
Lesotho
Equatorial Guinea
Estonia
Singapore
Cabo Verde
China
Heavily indebted poor countries (HIPC)
Sub-Saharan Africa (excluding high income)
Montenegro
Sub-Saharan Africa
Sub-Saharan Africa (IDA & IBRD countries)
Upper middle income
World
Albania
Least developed countries: UN classification
Low income
United Arab Emirates
Early-demographic dividend
North America
Ireland
Eritrea
Dominica
Guinea
Pre-demographic dividend
Moldova
IBRD only
Low & middle income
Bosnia and Herzegovina
Middle income
IDA & IBRD total
IDA only
Lithuania
Seychelles
IDA total
Monaco
Pakistan
Madagascar
Morocco
Caribbean small states
Bulgaria
Cambodia
East Asia & Pacific (IDA & IBRD countries)
Liechtenstein
Czech Republic
Bangladesh
Slovak Republic
Late-demographic dividend
Colombia
Saudi Arabia
Central Europe and the Baltics
East Asia & Pacific (excluding high income)
Kenya

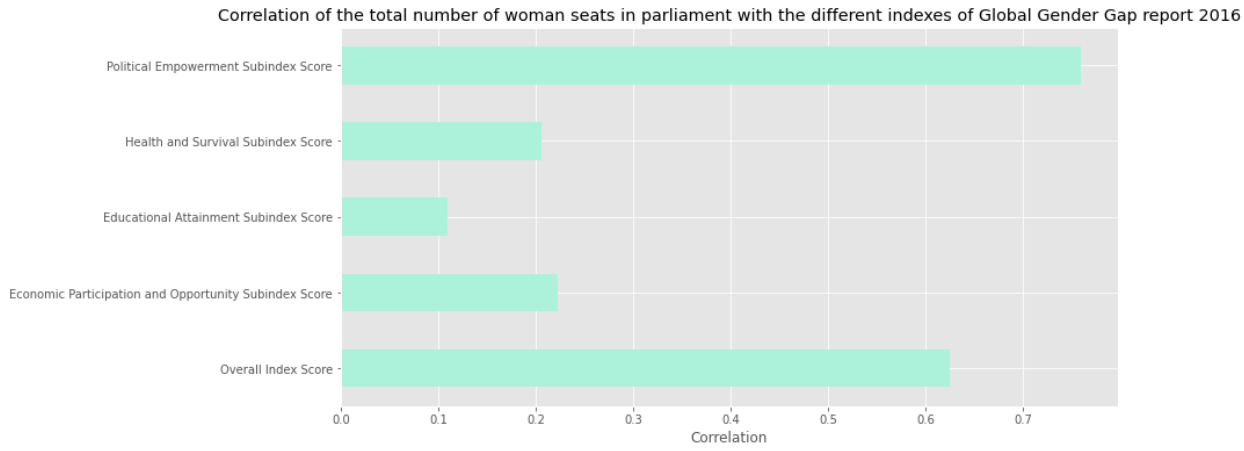

Greece
South Asia (IDA & IBRD)
South Asia
United States
East Asia & Pacific
Europe & Central Asia (IDA & IBRD countries)
Kyrgyz Republic
Lower middle income
Tajikistan
Arab World
IDA blend
Other small states
Europe & Central Asia (excluding high income)
Panama
Sao Tome and Principe
Zambia
Latvia
Cyprus
Togo
Middle East & North Africa (IDA & IBRD countries)
Middle East & North Africa (excluding high income)
Jamaica
Fragile and conflict affected situations
Small states
Indonesia
Mongolia
Middle East & North Africa
Korea, Rep.
Azerbaijan
Barbados
Malawi
St. Lucia
San Marino
Korea, Dem. People's Rep.
Uruguay
Libya
Fiji
Uzbekistan
Chile
Jordan
Paraguay
Chad
Egypt, Arab Rep.
Turkey
Niger
Venezuela, RB
Gabon
Guatemala
Somalia
Guinea-Bissau
Romania
St. Kitts and Nevis
Bahamas, The
Syrian Arab Republic
St. Vincent and the Grenadines
Malta
Russian Federation
Djibouti
Croatia
Sierra Leone
India
Ukraine
Mauritius
Antigua and Barbuda
Liberia
Ghana
Armenia
Malaysia
Hungary
Samoa
Brazil
Myanmar
Japan
Botswana
Gambia, The
Burkina Faso
Cote d'Ivoire
Marshall Islands
Congo, Dem. Rep.
Mali
Bhutan
Bahrain
Congo, Rep.
Central African Republic
Benin
Tuvalu
Brunei Darussalam
Kiribati
Eswatini
Thailand
Pacific island small states
Maldives
Iran, Islamic Rep.
Sri Lanka
Nigeria
Nauru
Belize
Lebanon
Comoros
Papua New Guinea
Solomon Islands
Kuwait
Oman
Yemen, Rep.
Haiti
Tonga
Vanuatu
Qatar
Micronesia, Fed. Sts.
Palau
2016.0
0
10
20
30
40
50
60

It might also make sense to see whether the proportion of the women in the national parliament might somehow relate to the global gender gap indices. As a caveat, we're already familiar that the gender gap data is from 2016, so we'll just merge it with the 2016 year column from the proportion of women in the national parliament data.
We then plot the correlation between the number of seats with the different indices.

```
#renaming the seats and country columns to make them look more meaningful

merged_woman_seats_and_gender_gap_data_2016 = merged_woman_seats_and_gender_gap_data_2016.rename(columns={2016:'Number of woman seats in parliament', 'key_0': 'Country'})

# Gonna make a subset df of the required columns to make it look more succinct
required_columns = score_cols + ['Number of woman seats in parliament', 'Country']
```

+ Code + Text

```
#lets see how closely are the different index scores correlated with the number of seats occupied by woman from different countries in the parliament
merged_corr_df = merged_woman_seats_and_gender_gap_data_2016[required_columns].corr()
title = "Correlation of the total number of woman seats in parliament with the different indexes of Global Gender Gap report 2016"
plt.xlabel('Correlation')
merged_corr_df[merged_corr_df.index.isin(score_cols)]['Number of woman seats in parliament'].plot.barh(color=colormap[0], figsize=(12,6), title = title)
```



From this, it can be observed that:

- All the indices are positively correlated with the proportion of seats.
- Political Empowerment Subindex seems to have a strong interdependence.

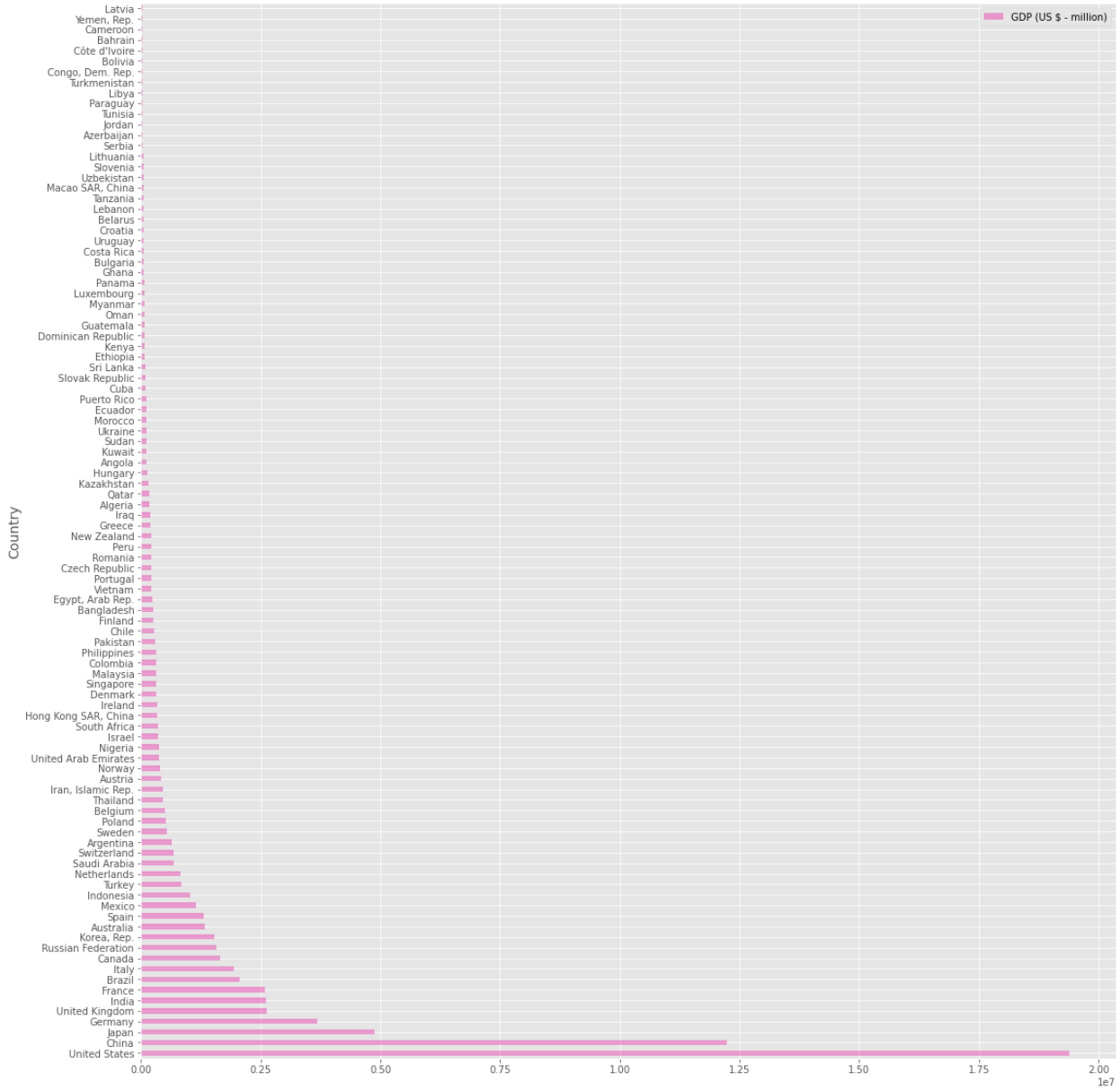
GDP (2017)

Finally we'll be going over our last dataset and exploring it as well.
Loading the file first:

```
# Let's now load the GDP data and start having a look at what it looks like

gdp_2017_df = pd.read_excel('/content/drive/My Drive/maryam/GDP (1).xls')
```

To get an idea of how the distribution of GDP (in $ US million), we plot a bar chart for the top countries with the most GDP to the least:

```
# Let's see the top 100 countries with the most GDP in 2017
gdp_2017_df[['Country', 'GDP (US $ - million)']].head(100).plot(kind='barh', y='GDP (US $ - million)', x='Country', figsize=(18 , 20), color=colormap[9], alpha=0.5)
```



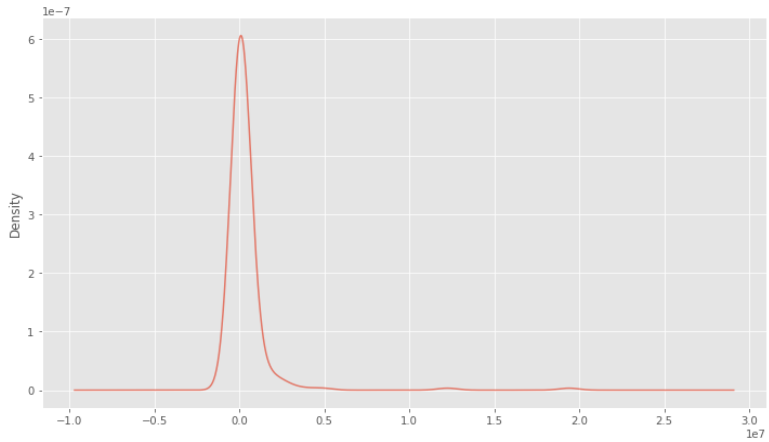
From this bar chart, we know:

- There is a big disparity in the distribution of GDP, it almost follows an exponential decay curve.

- The USA and China far dominate any other country in sheer GDP terms. An adjusted measure, such as the GDP per capita might be more insightful to look at while doing a comparison of different countries.

We can get a better idea of the distribution of the GDP if we use a KDE plot. The code snippet below will do the job for us:

```
# Let's finally see how the GDP of all countries is distributed

gdp_2017_df[~gdp_2017_df['rank'].isna()]['GDP (US $ - million)'].plot(kind='kde', figsize=(12,7), alpha = 0.7)
```



We observe the following points:

- On first glance the distribution is unimodal, this where the majority of the countries' GDP lies.
- As we look towards the right of the curve, we can see two small bumps, these are because of the USA and China being outliers.

Further, we merge the GDP dataframe with that of the global gender gaps'. The joined dataframe is shown below:

```
merged_gender_gap_gdp_df = pd.merge(left=gender_gap_df_dict[data], right=gdp_2017_df[['Country', 'GDP (US $ - million)']], how='inner', left_on='Country', right_on='Country')

merged_gender_gap_gdp_df[score_cols+['GDP (US $ - million)', 'Country']]
```

	Overall Index Score	Economic Participation and Opportunity Subindex Score	Educational Attainment Subindex Score	Health and Survival Subindex Score	Political Empowerment Subindex Score	GDP (US $ - million)	Country
0	0.873670	0.806184	1.000000	0.969801	0.718696	23909.3	Iceland
1	0.845047	0.793658	1.000000	0.979558	0.606971	251885	Finland
2	0.841951	0.818156	0.999801	0.974075	0.575774	398832	Norway
3	0.815413	0.802349	0.999233	0.973884	0.486186	538040	Sweden
4	0.799811	0.816934	0.957830	0.972313	0.452167	9135.45	Rwanda
...	...	...	...	...	...	...	...
132	0.590517	0.593791	0.733018	0.948905	0.086353	15334.3	Mali
133	0.587451	0.356671	0.975409	0.970897	0.046826	454013	Iran, Islamic Rep.
134	0.586693	0.667051	0.618352	0.968133	0.093235	9871.25	Chad
135	0.582871	0.328045	0.960768	0.965882	0.076791	686738	Saudi Arabia
136	0.555905	0.319604	0.810829	0.966640	0.126549	304952	Pakistan

137 rows × 7 columns

We then plot a pair-plot to get an idea of which gender gap indices, if any, impact the GDP. The code shown below elucidates this further.

```
# Lets see the Correlation of different indexes with the GDP
plt.rcParams["axes.labelsize"] = 14

g = sns.pairplot(merged_gender_gap_gdp_df[score_cols + ['GDP (US $ - million)']])
for ax in g.axes.flatten():
    # rotate x axis labels
    ax.set_xlabel(ax.get_xlabel(), rotation = 90)
    # rotate y axis labels
    ax.set_ylabel(ax.get_ylabel(), rotation = 0)
    # set y labels alignment
    ax.yaxis.get_label().set_horizontalalignment('right')
```

We can observe that:

- Overall index and the Economic participation Subindex are clustered around the center, where the GDP is highest.

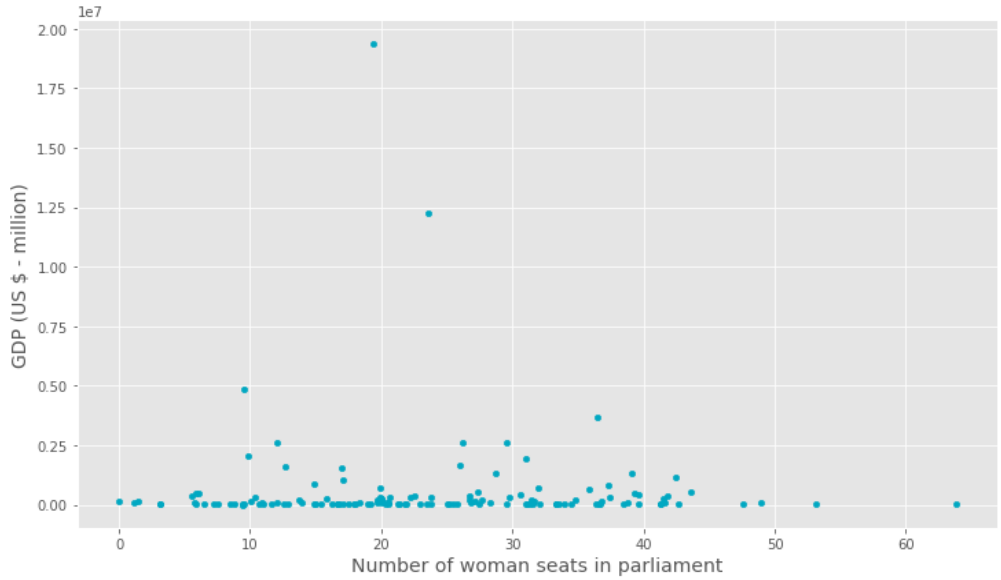
- Both Educational Attainment Subindex as well as Health Survival Subindex Scores tend to be high where the GDP is high.
- Political Empowerment Subindex does not seem to have any structured impact on the GDP, although there is a slightly linear relationship between the two.

Finally we merge the proportion of women in national parliaments data into this merged dataframe.

```
#Let's see how the number of woman seats in the national parliament relate to the GDP of that country


# merging all the different sources into one
merged_complete_df = pd.merge(left = merged_woman_seats_and_gender_gap_data_2016, right = merged_gender_gap_gdp_df[['Country','GDP (US $ - million)']], on= 'Country', how = 'inner')
```

We haven't yet inspected whether or not there is any relationship between the proportion of women seats in the national parliament or the GDP. The code below shows how to do that.

```
# Is there a relation between the Number of woman seats in parliament and the GDP (US $ - million)?

merged_complete_df.plot.scatter(x="Number of woman seats in parliament", y = "GDP (US $ - million)", color = colormap[2], figsize=(12,7))
```



Observations:

- Both commodities do not seem to hold a significant effect on each other but we can observe that for countries with higher GDP, the seats are neither at the lower or the higher end.

Problem statement

Given our analysis above, we do have some data that is capacitive of holding meaningful relationships.
We have analysed several attributes independently and in synergy as well. One thing we have noticed is that all of these data sources are attributes calculated separately for each country. The different columns in all the sources are in some way metrics that shed light on some aspect of a tangible element pertaining to that country.

From seeing the different features belonging to different datasets, we can also use them to somehow compare how these countries fall together and see how alike they might be given the metrics above

Modelling

To achieve the solution to the above found problem statement we have to find a way to somehow bring the most similar countries together using the available attributes.

One way to achieve this would be by clustering. Clustering is an unsupervised machine learning method wherewith we have no labelled targets associated with the data and we just aim to find groupings in data that maximise the intra-group similarity and minimize the inter-group similarity.

The algorithm we're using to cluster our data is the hierarchical agglomerative type. In this approach we start making the clusters in a bottom up fashion, as we go up we keep on making more bigger clusters from the smaller ones. This is an iterative process until our algorithm converges.

Since our data for 2016 for the proportion of women seats in the national parliament has a missing value for the country of Georgia, we'll impute it. The method we've used to impute is my taking a running average of the proportion of seats for this country for the years of 2015 and 2017. The snippet belows shows how that was achieved.

```
# apparently there's a missing value for the country of georgia for the Number of woman seats in parliament dataset for 2016, so we'll interpolate by taking the running average of 2015,2017 and fill in that value
# so that our clustering algorithm can work properly

seats_df = woman_seats_in_national_parliaments_df_dict[data_woman_parliament_seats]
merged_complete_df.loc[45, "Number of woman seats in parliament"] = (seats_df.loc['Georgia', 2015] + seats_df.loc['Georgia', 2015]) / 2
```

Further, we also remove the categorical variables such as the name of the country because this is a criteria that doesn't add any meaning to the clustering algorithm, since it's different for each country. The code snippet below demonstrates how we did that.

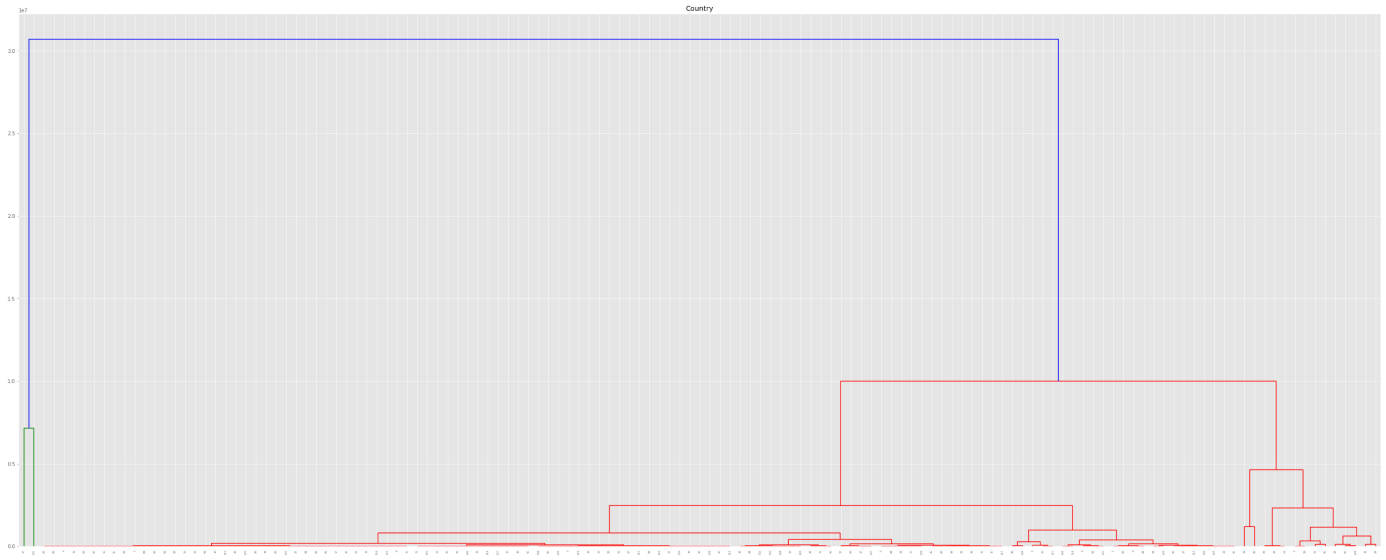
```
#Removing the categorical attributes so that clustering algorithms can run on the data

inp_data = merged_complete_df[list(merged_complete_df.columns.drop('Country'))]
```

We also remove the ranks from our data, since they are encodings for each country and are unique, thus, aren't required for the clustering algorithm. Another reason to remove them is that the information provided by these ranks is already complimented by their corresponding scores and therefore keeping the ranks is also redundant. The code snippet below shows how we did this.

```
# removing the ranks
inp_data = inp_data[inp_data.columns.drop(rank_cols)]
```

Then we use the scipy API to draw a dendrogram to visualize how the groups are being formed for our data. The linkage() function computes a distance matrix between the clusters at each iteration. The method argument passed as 'ward' conveys the usage of the 'Ward variance minimization algorithm' which is a criteria for finding the similarity between two clusters.
The code snippet below shows how the clusters were formed and then plotted as a dendrogram followed by the plot itself.

```
import scipy.cluster.hierarchy as shc

plt.figure(figsize=(50, 20))
plt.title("Country")
dend = shc.dendrogram(shc.linkage(inp_data, method='ward'))
```



From looking at this we can see two discernable clusters forming in the red branch. Both of these have little disparity but can still be taken as being separate. On the other hand, we have the green branch which is another cluster which is much more dissimilar.

Deducing from the above dendrogram plot, we can assume that a good estimate for the number of clusters could be 3. We then go on to use the ‘scikit-earn’ API and import the ‘AgglomerativeClustering’ algorithm and initialise it with the arguments shown in the code snippet below.
We fit our models on the filtered data we fed method above to produce the dendrogram and get the labels for each country. The label is allocates the country to a cluster.

```
model = AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='ward')
model.fit(inp_data)
labels = model.labels_
```

Here’s a snapshot of all the predicted cluster-labels for all the countries. We can see that the most frequent cluster is ‘2’ while the frequent is ‘0’.

```
labels
```

```
array([2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2,
       2, 2, 1, 2, 2, 0, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1,
       1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2,
       2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0,
       2, 2, 2])
```

Let's visualize the clusters. We know since we're making clusters in a high dimensional space as we used 7 attributes to make these clusters. We have to project them to a 2-D space to be able to visualize them easily. To achieve this, we use PCA- principal components analysis. This algorithm reduces the dimensions of the data by finding the components capturing the most variance in the data, such that more information can be preserved while also losing the dimensions of our data. The snippet below shows how we used PCA to achieve this:

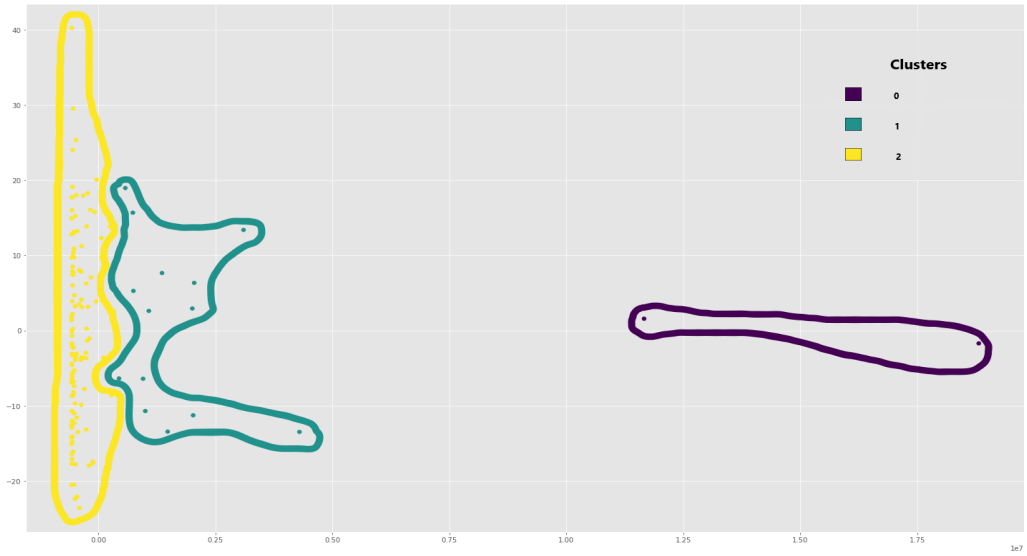
```
# projecting the inp_data to a lower dimentional data

x = PCA(2)
projected = x.fit_transform(inp_data)
```

The data in 'projected' is a 2D vector which summarises the information held in our filtered dataframe.

Finally we plot these projected components using the code snippet below.

```
plt.figure(figsize=(25,15))
plt.scatter(projected[:,0], projected[:,1], c=labels)
```

Along the X and Y coordinates are plotted the the 2 components to which we projected our multidimensional input dataframe which we fed to the clustering algorithm. The scatter plot shows dots which represent the countries. Each country is allocated a cluster label, which are passed as an argument to the plt.scatter() method. The plot is shown below.



We further copy our dataframe to another one and store the cluster labels in it.

```
resultant_df = merged_complete_df.copy()

resultant_df['cluster'] = labels

resultant_df["cluster"].unique()

array([2, 1, 0])
```

Finally we see which countries have been assigned which cluster:

Cluster 0 is small with only two countries:

```
resultant_df[resultant_df['cluster']==0]["Country"]

27             China
131    United States
Name: Country, dtype: object
```

Cluster 1 is slightly bigger:

```
resultant_df[resultant_df['cluster']==1]["Country"]
```

```
5               Australia
19                 Brazil
24                 Canada
34                Germany
39                  Spain
43                 France
44         United Kingdom
53              Indonesia
54                  India
59                  Italy
62                  Japan
67            Korea, Rep.
81                 Mexico
109    Russian Federation
Name: Country, dtype: object
```

Finally we have the cluster 2 which is relatively bigger:

```
resultant_df[resultant_df['cluster']==2]["Country"]
```

```
0                    Angola
1                   Albania
2       United Arab Emirates
3                 Argentina
4                   Armenia
6                   Austria
7                Azerbaijan
8                   Burundi
9                   Belgium
10                    Benin
11             Burkina Faso
12               Bangladesh
13                 Bulgaria
14                  Bahrain
15    Bosnia and Herzegovina
16                  Belarus
17                   Belize
18                  Bolivia
20                 Barbados
21         Brunei Darussalam
22                   Bhutan
23                 Botswana
25              Switzerland
26                    Chile
28                 Cameroon
29                 Colombia
30               Costa Rica
31                     Cuba
32                   Cyprus
33           Czech Republic
35                  Denmark
36       Dominican Republic
37                  Algeria
38                  Ecuador
40                  Estonia
41                 Ethiopia
42                  Finland
45                    Ghana
46                   Guinea
47              Gambia, The
48                   Greece
49                Guatemala
50                 Honduras
51                  Croatia
52                  Hungary
55                  Ireland
56       Iran, Islamic Rep.
57                  Iceland
58                   Israel
60                  Jamaica
61                   Jordan
63               Kazakhstan
64                    Kenya
65          Kyrgyz Republic
```

```
resultant_df[resultant_df['cluster']==2]["Country"]
```

```
66                 Cambodia
68                   Kuwait
69                 Lao PDR
70                  Lebanon
71                  Liberia
72                Sri Lanka
73                  Lesotho
74                Lithuania
75               Luxembourg
76                   Latvia
77                  Morocco
78                  Moldova
79               Madagascar
80                 Maldives
82           Macedonia, FYR
83                     Mali
84                    Malta
85               Montenegro
86                 Mongolia
87               Mozambique
88               Mauritania
89                Mauritius
90                   Malawi
91                 Malaysia
92                  Namibia
93                  Nigeria
94                Nicaragua
95              Netherlands
96                   Norway
97                    Nepal
98              New Zealand
99                     Oman
100                Pakistan
101                  Panama
102                    Peru
103             Philippines
104                  Poland
105                Portugal
106                Paraguay
107                   Qatar
108                 Romania
110                  Rwanda
111            Saudi Arabia
112                 Senegal
113               Singapore
114             El Salvador
115                  Serbia
116                Suriname
117          Slovak Republic
118                Slovenia
119                  Sweden
120                    Chad
121                Thailand
122              Tajikistan
```