ECOMOMIC DEVELOPMENT AND POLITICAL LIBERALIZATION

INTRODUCTION

In the past decades, different nations around the world have witnessed instances of economic developments and political liberalizations with most of such studies pegged on economic development. There have emerged questions on whether a country being a democracy results in any form of economic growth or none, with most studies presenting a mix of results (Zhang, 2020). These developments have been attached to growth in trade and liberalization, with more concerns placed on trade openness as being important measures of economic growth. However, the assumption of a positive relation between liberalization and trade has underwent a number of criticisms among different authors. Recently, in examining the joint impacts of economic liberalizations and economic liberalizations, there were conclusions that a number of reforms were quite helpful, especially when countries open up their economies to international trade as causing significant economic growth. Countries that become democracies first before opening up their economies have realized a significant sluggishness in their economic growth measures (Adegboye et.al, 2020). This indicates that countries should undertake liberalizations of their economies before considering political liberalizations.

Politics stem from the change in the in the government structure.

Economic development measures within countries of the world require sustainment over the years. Most of the developing countries have often underwent challenges from substantial volatility in growth that has a negative effect in economic growth. In this case, policy makers have been advised to consider output volatilities when determining the reform sequences from different global countries (Hummel, 2020). Furthermore, trade openness has also been discussed as aiding economic growth across countries. Nonetheless, it is associated with significant levels of volatility at the macroeconomic levels. Moreover, democratization has also presented ambiguous effects on growth with some positive impacts especially in regards to economic stability from enhanced cohesion and creation of an enabling business environment for economic growth. Available literature in terms of growth volatility suggests the existence of trade-offs in the liberalization sequences over the years.

Literature Review

Economic theory

In economics, economic theory suggests that financial developments have a direct impact on economic growth metrics in the past years. The theory is also linked to political liberalizations since harmonious business environment provided by political stability is essential for business and economic growth (Li & Yao, 2020). These arguments are dated back to Schumpeter in 1912 in which services offered by financial intermediaries were quire helpful in both innovations and relevant developments over the past years. There are five main functions of the financial system across countries that influence economic growth. First, the financial systems enhance mobilization of savings for customers, ensuring that they have appropriate access to their finances across existing commercial banks (Haddad, 2020). The banks across different countries enhance development of the economy through offering savings that customers can use to access credit when in need. Secondly, financial sectors also aid in reducing risks across the economy. This is achieved through the inclusion of insurance corporations, key in terms of offering compensations in the event of any risk. Furthermore, this is also realized through most of customer savings kept within the existing financial systems to assure customers of the funds when needed.

Furthermore, the financial systems across the global economies aid in the facilitation of exchanging goods and services across clients around the world. In this case, governments aid in creating the good-will through political stable landscapes, to ease exchange of goods and services (Khayitboy & Ilhom, 2020). In this case, the aspect of liberalization plays a critical role especially in ensuring that the individuals realize their specific economic targets, hence improving the economic prospects over the years as witnessed across both developed and developing economies around the world. Most of these provisions are quite helpful in terms of promoting checks and balances, in terms of economic milestones in comparison to the previous years. Appropriate financial systems aids in the realization of underlying functions and roles as stipulated from economic perspectives over the years.

METHODOLOGY AND APPROACH FOR ANALYSIS

*# <center>Exploratory Data Analysis*

**Table of contents**

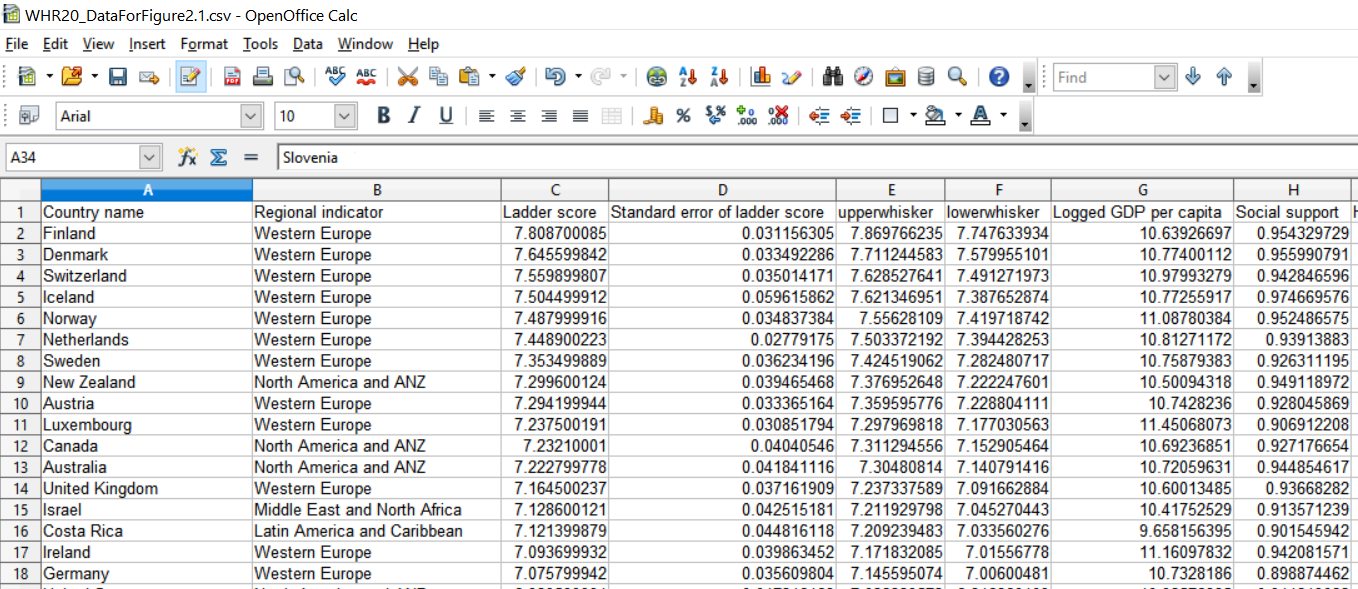
1. [Why EDA ?](file:///E:\ONLINE%20WORKING\Economic_Analysis\exploratory-data-analysis-pandas-numpy-seaborn.html#1.-Why-EDA-?)
2. [Pandas, Numpy, Matplotlib, Seaborn](file:///E:\ONLINE%20WORKING\Economic_Analysis\exploratory-data-analysis-pandas-numpy-seaborn.html#2.-Pandas,-Numpy,-Matplotlib,-Seaborn)
3. [Data types](file:///E:\ONLINE%20WORKING\Economic_Analysis\exploratory-data-analysis-pandas-numpy-seaborn.html#3.-Data-types)
4. [Exploring categorical features](file:///E:\ONLINE%20WORKING\Economic_Analysis\exploratory-data-analysis-pandas-numpy-seaborn.html#4.-Exploring-categorical-features)
5. [Exploring numerical features](file:///E:\ONLINE%20WORKING\Economic_Analysis\exploratory-data-analysis-pandas-numpy-seaborn.html#5.-Exploring-numerical-features)
6. [Bivariate analysis](file:///E:\ONLINE%20WORKING\Economic_Analysis\exploratory-data-analysis-pandas-numpy-seaborn.html#6.-Bivariate-analysis)
7. [Outliers](file:///E:\ONLINE%20WORKING\Economic_Analysis\exploratory-data-analysis-pandas-numpy-seaborn.html#7.-Outliers)

**1. Why EDA ?**

Because in order to start working with our data, we need to know what kind of data we are dealing with. And this detective work got itself the dry name of exploratory data analysis (which I don't think does justice to it).

These are only some of the questions that we ask ourselves. Depending on the answer, we have to proceed with different processing steps before we can use any algorithms on our data:

* Do we have 1000 or 1 million entries in our data ?
* Are we dealing with text or numbers ?
* Do we have dates ? What format to these dates have ?
* Do we have outliers ? (Data points that are extremely different than all the other ones)
* Do we have missing data ? That is, is any of the cells in our dataset empty ?

If I just open my data, the csv file, in a spreadsheet application and look at it with the naked eye, I won't be able to tell much.  


I will open the csv file and read all my data.

In [2]:

*#let's set the precision to 2 decimal places*

pd.set\_option("display.precision", 2)

Pandas makes it very easy to handle tabular data.

Tabular data means that our data fits or belongs in a table. Other types of data can be visual (that is, images, for which it doesn't really make sense to be stored as csv files).

The standard way to store tabular data is that:

* **each row** represents a different **observation**. Observation is a fancy Statistics term, but it just means a new data point, a new measurement we did. If our data is about happiness in various countries, each row contains data for a new country.
* **each column** is a different **feature** (or attribute) of our observations. For the World Happiness Report dataset, examples of features can be the Country name, the Regional indicator or the Social Support score.

Let's use the numpy library to see the maximum value of the **feature** *Ladder score* across **all observations** in our dataset (all countries).

In [3]:

*#Let's import the numpy library*

**import** **numpy** **as** **np**

*#and use a numpy function to see what's the maximum value for our Ladder score feature*

np.max(df["Ladder score"])

Out[3]:

7.808700085

And since we're here, I'll do a quick demo of how convenient it is to use pandas DataFrame structure.  
We found the maximum values for "Ladder score" feature. What is the row number of the entry with the max Ladder score ?

In [4]:

df['Ladder score'].argmax()

Out[4]:

0

It only took one line of code to find the row number. Let's see this observation's features, to convince ourselves we got the right entry. # Mind that when displaying one single entry from the DataFrame, the feature values won't appear o a row anymore, but will be displayed as a column (I find this switch a bit confusing).

In [5]:

df.iloc[df['Ladder score'].argmax()]

Out[5]:

Country name Finland

Regional indicator Western Europe

Ladder score 7.8

Standard error of ladder score 0.031

upperwhisker 7.9

lowerwhisker 7.7

Logged GDP per capita 11

Social support 0.95

Healthy life expectancy 72

Freedom to make life choices 0.95

Generosity -0.059

Perceptions of corruption 0.2

Ladder score in Dystopia 2

Explained by: Log GDP per capita 1.3

Explained by: Social support 1.5

Explained by: Healthy life expectancy 0.96

Explained by: Freedom to make life choices 0.66

Explained by: Generosity 0.16

Explained by: Perceptions of corruption 0.48

Dystopia + residual 2.8

Name: 0, dtype: object

**3. Data types**

We have some idea about or features types just by looking a the CSV file. But a better method is the one below.

In [6]:

*#DataFrame has this very handy method.*

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 153 entries, 0 to 152

Data columns (total 20 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Country name 153 non-null object

1 Regional indicator 153 non-null object

2 Ladder score 153 non-null float64

3 Standard error of ladder score 153 non-null float64

4 upperwhisker 153 non-null float64

5 lowerwhisker 153 non-null float64

6 Logged GDP per capita 153 non-null float64

7 Social support 153 non-null float64

8 Healthy life expectancy 153 non-null float64

9 Freedom to make life choices 153 non-null float64

10 Generosity 153 non-null float64

11 Perceptions of corruption 153 non-null float64

12 Ladder score in Dystopia 153 non-null float64

13 Explained by: Log GDP per capita 153 non-null float64

14 Explained by: Social support 153 non-null float64

15 Explained by: Healthy life expectancy 153 non-null float64

16 Explained by: Freedom to make life choices 153 non-null float64

17 Explained by: Generosity 153 non-null float64

18 Explained by: Perceptions of corruption 153 non-null float64

19 Dystopia + residual 153 non-null float64

dtypes: float64(18), object(2)

memory usage: 24.0+ KB

What I see in the output above:

* my data is a DataFrame, with 153 entries (from 0 to 152)
* I have 20 columns (from 0 to 19)
* all my columns have 153 non-null values (I don't have "missing" data in any of these columns)
* my column types are: object (2 of them) and float64\* (18 of them)

\*float64 means they can store fractional numbers and each number takes 64 bits

The 'object' type I see above most likely refers to a string. I'll use [DataFrame indexing / selection](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html) to look at one particular value to verify my assumption.

In [7]:

print(df['Country name'][0])

print(df['Regional indicator'][0])

Finland

Western Europe

Ok, so, in this case, 'object' means String.

**4. Exploring categorical features**

We have 2 features which contain text:

* Contry
* Region

**Country**

Our intuition is that each country is unique in our dataset (one country per row). This is what we would expect from a study of happiness levels in different countries across the worls. We can verify this assumption, to make sure we don't have errors in our data. For example, the social scientist running this study could have accidentally entered the same observation twice because she was working late to finish her data analysis.

In [8]:

*#how many entries we have for each country*

*#shown in descending order (highest value first)*

df["Country name"].value\_counts().sort\_values(ascending = **False**)

Out[8]:

Lithuania 1

South Africa 1

Luxembourg 1

El Salvador 1

India 1

..

Czech Republic 1

Dominican Republic 1

Bosnia and Herzegovina 1

Niger 1

Mauritania 1

Name: Country name, Length: 153, dtype: int64

In [9]:

*#Uncomment the line below to see what data type we used. This is a nice way to explore the functioning of pandas.*

*#print("\nThe code above returns a date of type: ", type(df['Country name'].value\_counts()))*

**Region**

Let's have a look at the **regions** now. It would be interesting to see what different regions we have. This would open the door for questions like: 'Are people happier in Western Europen than in Eastern Europe ?'. We don't know yet what question we can ask and exploring our data informs our next steps.

By the way, since we are dealing with long column names, it's worth mentioning that I don't have to type the whole column name. I just input the first 3 letters and press Tab for autocomplete.

We see in the output below that:

* Europe is split into 2: 'Western Europe' and 'Central and Eastern Europe'
* The Americas are divided into 2: 'Latin America and Caribbean' and 'North America and ANZ' (which is North America, Australia and New Zealand)
* Africa is split into 2: 'Sub-Saharan Africa' and 'Middle East and North Africa'
* Asia is divided into 3: 'Southeast Asia', 'South Asia' and 'East Asia'
* There is a group of post-Soviet republics in Eurasia making up the 'Commonwealth of Independent States'

In [10]:

*#here's each individual region and its corresponding frequency (the statistical term*

*#for the number of times this region appears in our dataset)*

df['Regional indicator'].value\_counts()

Out[10]:

Sub-Saharan Africa 39

Western Europe 21

Latin America and Caribbean 21

Central and Eastern Europe 17

Middle East and North Africa 17

Commonwealth of Independent States 12

Southeast Asia 9

South Asia 7

East Asia 6

North America and ANZ 4

Name: Regional indicator, dtype: int64

In [11]:

*#we have 10 regions and pandas DataFrame has a method to find this out*

print(f"The number of regions in our dataset is: **{**df['Regional indicator'].nunique()**}**")

The number of regions in our dataset is: 10

I just used Python's fancy formatting in the line of code above. If you like it and want to read more, know that it's called Literal String Interpolation (but the popular name is f-string). You can read more [here](https://www.programiz.com/python-programming/string-interpolation).

**Visualisation for categorical features**

Since the frequencies (the number of times they appear in our dataset) of our regions is greater than one, it invites us to look at them in a more intuitive way rather than the text displayed above.

It is generally much better for the audience to present any data in visual form, whenever possible. For countries, nothing else made sense since each country appeared once in our data. But for regions, we can use a **bar chart**.

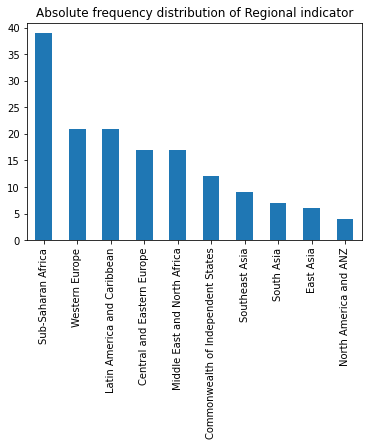
The bar chart below shows the same information as the table we've seen earlier.  
But in visual form it's so much easier to gain insights like "Sub-Saharan Africa is present in our dataset approximately twice as much as the next region in line, Western Europe".

In [12]:

df['Regional indicator'].value\_counts().plot(kind='bar', title='Absolute frequency distribution of Regional indicator')

Out[12]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f3650201890>



In the code above I've used **Pandas built-in capabilities for data visualization**. I didn't feel a need to turn to matplotlit or seaborn for basic visualisation that can be provided by pandas.  
If you feel like you want to read more abour Pandas visualisation, see the [official documentation.](https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html)

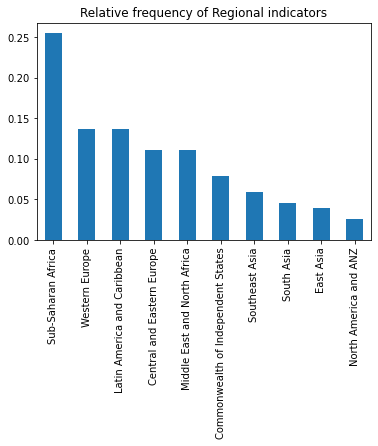
Another obsvervation for the plot above is that those numbers are absolute frequencies. That is, the bar chart shows the number of times each region is present in our dataset. Sometimes it's enough to know that we have 39 countries from Sub-Saharan Africa. But there are times when we're wondering how much this represents in terms of percentage.

In [13]:

(df['Regional indicator'].value\_counts()/df.shape[0]).plot(kind='bar', title='Relative frequency of Regional indicators')

Out[13]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f364bf6eb10>



Now we know that Sub-Saharan Africa represents 25% of our data. For this dataset this is not unusual. But imagine you're trying to see how happy people are in a single country, you broadcast a digital survey that people can take and during data analysis you realize that 25% of the people who filled in the survey are from the same city in this country.

**5. Exploring numerical features**

Pandas has a nice built-in method that performs descriptive statistics on a DataFrame.  
It shows us:

* the number of values for each feature (again, an opportunity to see if we have missing values for any feature)
* the mean value
* the standard error
* the min and max value
* the median of our data (50%)
* the lower and upper quartile (25% and 75%)

Insights from the descriptive statistics above:

* Ladder score actually goes from 2.5 to 7.8. There's no 0 or 10.
* Healthy life expectancy has a minimum of 45 and a maximum of 76. This is a large range. There are countries in our dataset where life expenctancy is 45 years !
* Generosity can be negative. It's the only feature that has negative values.
* Other features are more difficult to interpret from the descriptive stats above.

Numerical data is best viewed as histograms. We will use both matplotlit and seaborn for this.

In [15]:

**import** **matplotlib.pyplot** **as** **plt**

**import** **seaborn** **as** **sns**

columns = ['Logged GDP per capita', 'Social support', 'Healthy life expectancy', 'Freedom to make life choices', 'Generosity',\

'Perceptions of corruption']

scols = int(len(columns)/2)

srows = 2

fig, axes = plt.subplots(scols, srows, figsize=(10,6))

**for** i, col **in** enumerate(columns):

ax\_col = int(i%**sc**ols)

ax\_row = int(i/scols)

sns.distplot(df[col], hist=**True**, ax=axes[ax\_col, ax\_row])

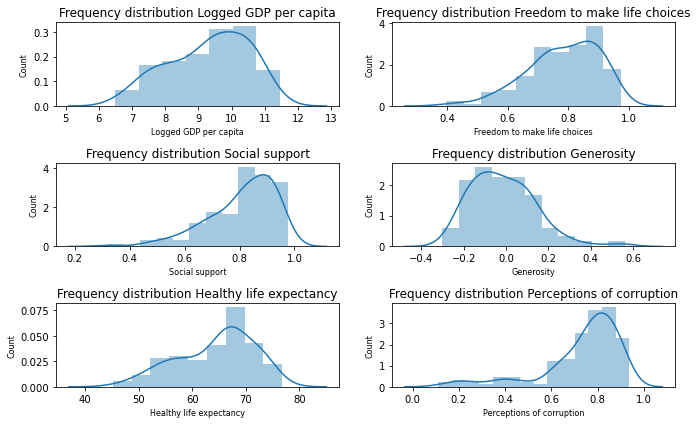
axes[ax\_col, ax\_row].set\_title('Frequency distribution '+ col, fontsize=12)

axes[ax\_col, ax\_row].set\_xlabel(col, fontsize=8)

axes[ax\_col, ax\_row].set\_ylabel('Count', fontsize=8)

fig.tight\_layout()

plt.show()



Insights from the visual exploration of our numerical data:

* the distributions of GDP, social support, healthy life expectancy, freedom and corruption are all [left skewed](http://www.cvgs.k12.va.us/DIGSTATS/main/descriptv/d_skewd.html) (or negative skew). That is to say, most of our values do not happen to be in the middle of the min-max range, but are pushed towards the upper end of our range. For all but Perception of corruption this is good news.
* generosity, though, is right skewed. The majority of the countries are in the bottom half of the generosity scale (unfortunately)

If you feel the need to read more about why we might want to look at the distribution of our data, [here is a very quick overview](http://www.cvgs.k12.va.us/DIGSTATS/main/descriptv/).

**6. Bivariate analysis**

All the explorations above belong to univariate analysis (that is, we looked at each variable individually). We can also perform bivariate analysis - we can look at pairs of two variables to explore a possible relation between them.

When Data Scientists perform a bivariate analysis, they look at scatterplots like the ones below and they search for clouds of dots that arrange themselves into straight diagonal lines. This is a visual representation of two variables that correlate.

Here's how to read the plots below:  
Let's look at the **second plot on the first row**. On the **far left** of the image we see "Logged GDP per capita". All plots on the first row have on the y axis (the vertical axis) the Logged GDP per capita as the label of the Y axis. Now look at the bottom of the plots, all the way down, under the second column we have "Social support" as the name of the X axis. All plots on the second columns have the Social support on the x axis (the horizontal axis).

Armed with this information, let's look at the contents of the second plot, first row. As the 'Social support' increases, so does 'Logged GDP per capita'. What does this mean ? Nothing more than the fact that the two feature seems to be correlated (correlation, not causation). Most likely (intuition dictates) as the country gets riches it can afford to offer more social support to its inhabitants.

Now look at the fourth subplot on the same row. The datapoints are all over place and there seems to be no correlation between 'GDP per capita' and 'Freedom to make life choices'.

Correlation is not assessed only by looking at a scatterplot, but this is a good start.

Take a few moments to explore the plots below. Look on the diagonal, from upper left to lower right. Do you recognize them from the univariate analysis section ? These are the histograms we've seen earlier.

In [16]:

*#This will take slightly longer than other plots, don't worry if the plots don't show up immediately.*

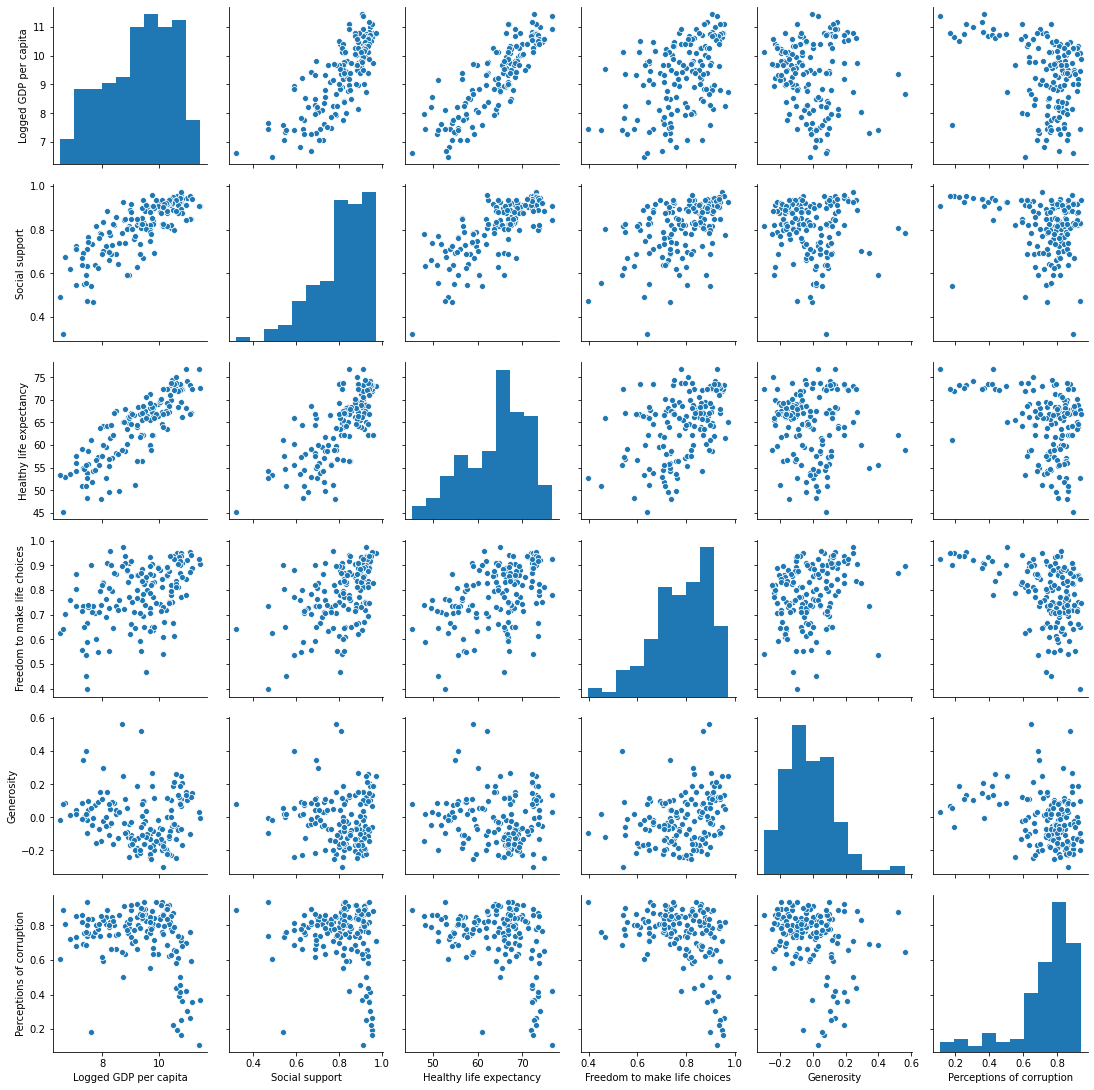
columns = ['Logged GDP per capita', 'Social support', 'Healthy life expectancy', 'Freedom to make life choices', 'Generosity',\

'Perceptions of corruption']

sns.pairplot(df[columns])

Out[16]:

<seaborn.axisgrid.PairGrid at 0x7f3649e3e7d0>



Seaborn allows us to add a 'hue' to our plots. We will set our scatterplots to assign **different colors to datapoints that belong to different global regions.**

You can read about [Seaborn pairplot here](https://seaborn.pydata.org/generated/seaborn.pairplot.html)

This helps us gain insight like: Sub-Saharan African countries (the purple dots, according to the legend on the right) have the lowest GDP and the lowest Healthy life expectancy, but they are not less generous than more fortunate countries.

In [17]:

*#This will take slightly longer than other plots, don't worry if the plots don't show up immediately.*

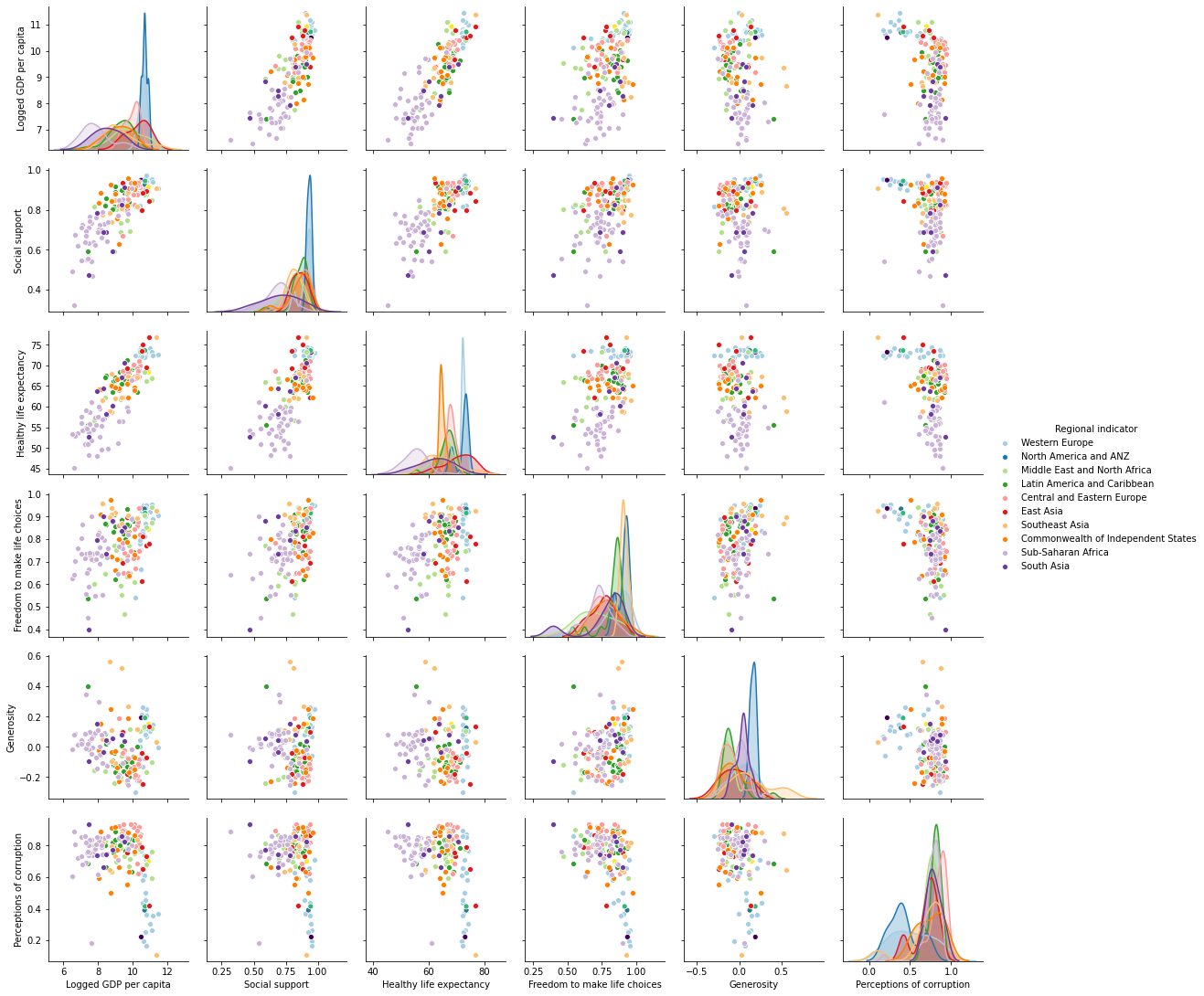
columns = ['Regional indicator','Logged GDP per capita', 'Social support', 'Healthy life expectancy', 'Freedom to make life choices', 'Generosity',\

'Perceptions of corruption']

sns.pairplot(df[columns], hue="Regional indicator", palette="Paired")

Out[17]:

<seaborn.axisgrid.PairGrid at 0x7f3644d2fa90>



Correlation is not assessed only by looking at a scatterplot, but the mono-coloured pairplot above was a good start.  
Another useful tool in the EDA toolset is the **correlation matrix.**

In [18]:

meaningful\_columns = ['Ladder score','Logged GDP per capita', 'Social support', 'Healthy life expectancy',

'Freedom to make life choices', 'Generosity',

'Perceptions of corruption', 'Ladder score in Dystopia']

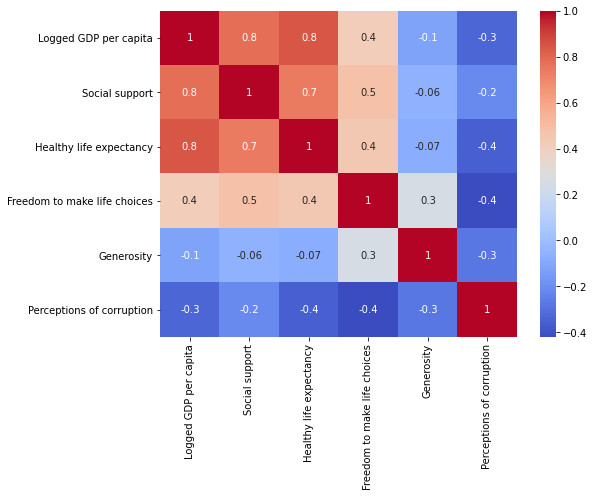
plt.figure(figsize=(8,6))

*#sns.heatmap(df.corr(), annot = True, fmt='.1g', cmap= 'coolwarm')*

sns.heatmap(df[columns].corr(), annot = **True**, fmt='.1g', cmap= 'coolwarm')

Out[18]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f363ea1f9d0>



**7. Outliers**

A nice way to spot outliers is a Box and Whiskers plot.

In [19]:

small = ['Social support', 'Freedom to make life choices', 'Generosity', 'Perceptions of corruption']

medium = ['Ladder score', 'Logged GDP per capita']

large = ['Healthy life expectancy']

f, axs = plt.subplots(1,3,figsize=(15,5))

*# equivalent but more general*

ax1=plt.subplot(1, 3, 1)

df.boxplot(column=small, ax = ax1)

plt.xticks(rotation=90)

ax2=plt.subplot(1, 3, 2)

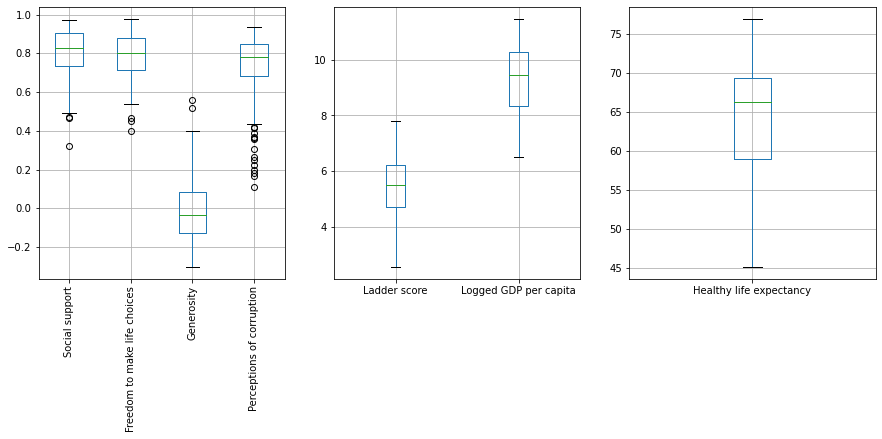
df.boxplot(column=medium, ax = ax2)

ax3=plt.subplot(1, 3, 3)

df.boxplot(column=large, ax = ax3)

Out[19]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f363e28a650>



The classical interpretation in Statistics is that whatever falls outside the 'whiskers' represents an outlier.

If you'd like to read more about box plots and what the box, the line that splits the box and the whiskers represent, [this resource](https://publiclab.org/notes/mimiss/06-18-2019/creating-a-boxplot-to-identify-outliers-using-codap) seemed to have nice visuals.

In practice, deciding what to do with outliers depends on many factory (whether you think they can be a mistake in data collection, for example).

Let's examine the case of Perceptions of corruption.

In [20]:

f, axs = plt.subplots(1,2,figsize=(12,4))

*# equivalent but more general*

ax1=plt.subplot(1, 2, 1)

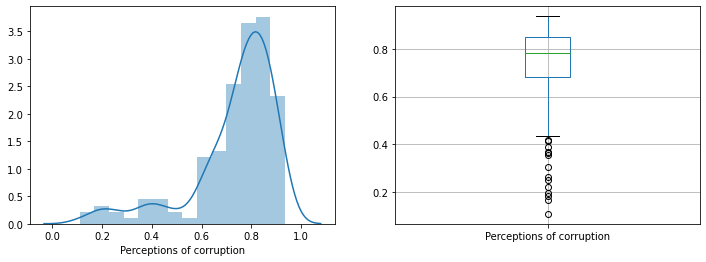
sns.distplot(df['Perceptions of corruption'], hist=**True**, ax=ax1)

ax2=plt.subplot(1, 2, 2)

df.boxplot(column=['Perceptions of corruption'], ax = ax2)

Out[20]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f363e0efa10>



Because 'Perceptions of corruption' feature is left skewed, countries with lowest perception of corruption are automatically categorized as outliers in the boxplot.

But just because they are technically outliers does not necessarily mean we should do something about them. The next question is: is the data correct ? Let's see who these outliers are.

In [21]:

(df[df['Perceptions of corruption'] < 0.4])[['Country name', 'Perceptions of corruption']].sort\_values(by = 'Perceptions of corruption', axis=0, ascending=**True**)

Out[21]:

|  | **Country name** | **Perceptions of corruption** |
| --- | --- | --- |
| **30** | Singapore | 0.11 |
| **1** | Denmark | 0.17 |
| **149** | Rwanda | 0.18 |
| **0** | Finland | 0.20 |
| **7** | New Zealand | 0.22 |
| **6** | Sweden | 0.25 |
| **4** | Norway | 0.26 |
| **2** | Switzerland | 0.30 |
| **15** | Ireland | 0.36 |
| **5** | Netherlands | 0.36 |
| **9** | Luxembourg | 0.37 |
| **10** | Canada | 0.39 |

It's no surprise to find almost all these countries in the bottom of the Perceptions of Corruption top. I admit I did not know about the low corruption in Rwanda !

*If you find the line of code above confusing, I did too, in the begining. When I found lines like this in someone else's code, I used to dissect them to examine the output and the data types. Maybe this tip helps.*

In [22]:

*#Uncomment the code below, line by line, if you want to dissect the previous line of code.*

*#I find it useful to first make a hypothesis about what I expect the line of code does before running it.*

*#print(f'df has {len(df)} entries')*

*#df['Perceptions of corruption'] < 0.4*

*#df[df['Perceptions of corruption'] < 0.4]*

*#print(f"our selection has {len(df[df['Perceptions of corruption'] < 0.4])} entries")*

*#(df[df['Perceptions of corruption'] < 0.4])[['Country name', 'Perceptions of corruption']]*

That's it for EDA for this rather simple tabular dataset.

RESULTS AND DISCUSSION

The goal of this project is to gain a clearer intuition of the relation of distribution of wealth and perception of corruption through visualization. If the Sokoloff-Engerman hypothesis is true, then we would expect to find a correlation (though rigorously testing whether Sokoloff-Engerman hypothesis’s explanation for such a correlation existing is correct is beyond the scope of this project).

The Data Used Here[¶](file:///C:\Users\User\Downloads\econ323-final%20(1).html#The-Data-Used-Here)

This project uses [this](https://www.kaggle.com/transparencyint/corruption-index) data set from Transparency International, and [this](https://www.kaggle.com/theworldbank/poverty-and-equity-database) data set from the World Bank. The former is representative of perceptions of corruption in 2017. The latter contains data about inequality observed over the course of 1974 through 2018.

This notebook was executed on [kaggle.com](http://kaggle.com), and a version of it will be maintained on [github](https://github.com/keeganland/econ323)

Kaggle Defaults[¶](file:///C:\Users\User\Downloads\econ323-final%20(1).html#Kaggle-Defaults)

In [271]:

# This Python 3 environment comes with many helpful analytics libraries installed

# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python

# For example, here's several helpful packages to load in

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

%matplotlib inline

# Input data files are available in the "../input/" directory.

# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.

/kaggle/input/corruption-index/index.csv

/kaggle/input/corruption-index/history.csv

/kaggle/input/poverty-and-equity-database/povstats-excel-zip-826-kb-/PovStatsEXCEL.xlsx

/kaggle/input/poverty-and-equity-database/povstats-csv-zip-242-kb-/PovStatsCountry.csv

/kaggle/input/poverty-and-equity-database/povstats-csv-zip-242-kb-/PovStatsFootNote.csv

/kaggle/input/poverty-and-equity-database/povstats-csv-zip-242-kb-/PovStatsCountry-Series.csv

/kaggle/input/poverty-and-equity-database/povstats-csv-zip-242-kb-/PovStatsSeries.csv

/kaggle/input/poverty-and-equity-database/povstats-csv-zip-242-kb-/PovStatsData.csv

The Transparency International data on corruption[¶](file:///C:\Users\User\Downloads\econ323-final%20(1).html#The-Transparency-International-data-on-corruption)

This dataset requires very little cleaning. It contains one of the variables of interest (as well as information on measurement error) as well as country codes that can be used for data frame merges. We will do an initial bar-graph visualization of it in order to do an initial intituition check.

In [272]:

corruption\_file = "/kaggle/input/corruption-index/index.csv"

df\_corruption\_index = pd.read\_csv(corruption\_file)

df\_corruption\_index = df\_corruption\_index.iloc[:,:8]

df\_corruption\_index.head()

Out[272]:

|  | CPI Rank | Country | Country Code | Region | Corruption Perceptions Index (CPI) | Standard Error | Lower Confidence Interval | Upper Confidence Interval |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | New Zealand | NZL | Asia Pacific | 90 | 2.56 | 86 | 94 |
| 1 | 1 | Denmark | DNK | Europe and Central Asia | 90 | 2.46 | 86 | 94 |
| 2 | 3 | Finland | FIN | Europe and Central Asia | 89 | 1.46 | 87 | 92 |
| 3 | 4 | Sweden | SWE | Europe and Central Asia | 88 | 1.33 | 85 | 90 |
| 4 | 5 | Switzerland | CHE | Europe and Central Asia | 86 | 1.57 | 83 | 89 |

In [273]:

# Used documentation from https://stackabuse.com/python-data-visualization-with-matplotlib/ to

#resize the figure so we can see all the countries listed in a large horizontal bar graph

fig\_size = plt.rcParams["figure.figsize"]

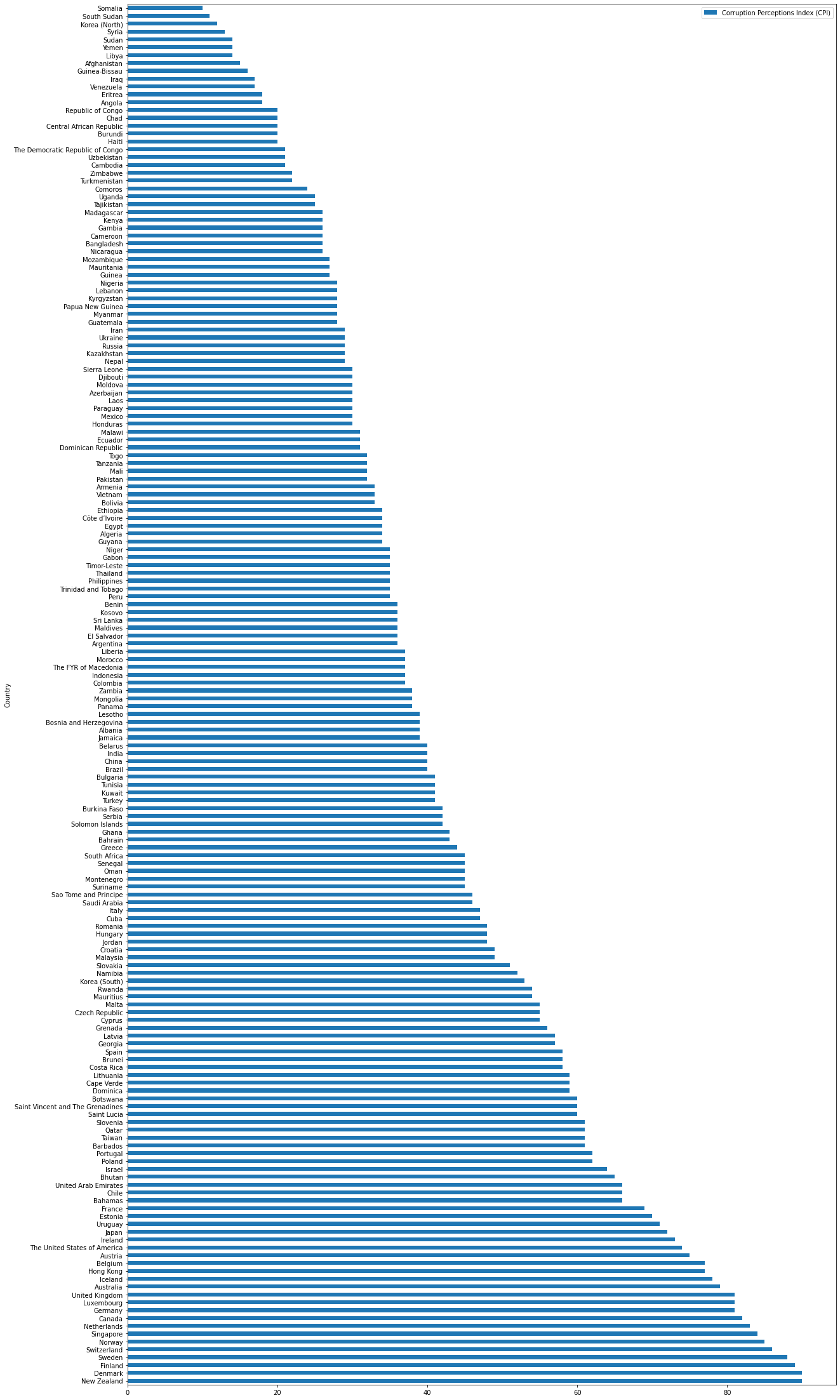
fig\_size[0] = 20

fig\_size[1] = 40

plt.rcParams["figure.figsize"] = fig\_size

#show bar graph

ax\_corruption\_bar\_graph = df\_corruption\_index.plot(x = "Country", y = "Corruption Perceptions Index (CPI)", kind = 'barh')



Gut check[¶](file:///C:\Users\User\Downloads\econ323-final%20(1).html#Gut-check)

The data does not look surprising. Countries that have a reputation as developing countries score poorly, whereas countries that have a reputation as wealthy, developed liberal democracies score highly.

This bar graph helps us get an intuitive feel for what CPI as a variable looks like. It varies from 0 to 100, 0 representing being perceived as most corrupt, 100 being least corrupt (that is, it runs on the "high score = good" intuition). It also serves as a guide, as the CPI varies numerically, which countries that numerical variation actually corresponds to.

In [274]:

# Return the figure size to something more managable for future plotting

fig\_size = plt.rcParams["figure.figsize"]

fig\_size[0] = 12

fig\_size[1] = 10

plt.rcParams["figure.figsize"] = fig\_size

The World Bank's data set[¶](file:///C:\Users\User\Downloads\econ323-final%20(1).html#The-World-Bank's-data-set)

The World Bank's dataset records various economic indicators for multiple regions and countries, with observations recorded by year. However, an observation does not exist for every indicator for every country. Furthermore, we are not necessarily interested in all the indicators in this data set. Because we are interested in seeing the relationship between corruption and inequality, we are most interested in their estimates of countries' Gini index, which is the standard measure of economic inequality. We shall therefore need to slice and clean the data frame.

In [275]:

poverty\_stats\_series\_file = "/kaggle/input/poverty-and-equity-database/povstats-csv-zip-242-kb-/PovStatsSeries.csv"

poverty\_stats\_country\_file = "/kaggle/input/poverty-and-equity-database/povstats-csv-zip-242-kb-/PovStatsCountry.csv"

poverty\_stats\_country\_series\_file = "/kaggle/input/poverty-and-equity-database/povstats-csv-zip-242-kb-/PovStatsCountry-Series.csv"

poverty\_stats\_data\_file = "/kaggle/input/poverty-and-equity-database/povstats-csv-zip-242-kb-/PovStatsData.csv"

df = pd.read\_csv(poverty\_stats\_data\_file)

df.head()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | East Asia & Pacific | EAS | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

5 rows × 50 columns

In [276]:

df\_indexed = df.set\_index(["Country Code", "Indicator Code"])

df\_indexed

Out[276]:

|  |  | Country Name | Indicator Name | 1974 | 1975 | 1976 | 1977 | 1978 | 1979 | 1980 | 1981 | ... | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | Unnamed: 49 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Country Code | Indicator Code |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| EAS | SI.SPR.PC40.ZG | East Asia & Pacific | Annualized growth in per capita real survey me... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| SI.SPR.PT10.ZG | East Asia & Pacific | Annualized growth in per capita real survey me... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| SI.SPR.PT60.ZG | East Asia & Pacific | Annualized growth in per capita real survey me... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| SI.SPR.PCAP.ZG | East Asia & Pacific | Annualized growth in per capita real survey me... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| SI.POV.GINI | East Asia & Pacific | GINI index (World Bank estimate) | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SI.POV.URGP.NC |  | Urban poverty gap at national poverty lines (%... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| SI.POV.URHC | Zimbabwe | Urban poverty headcount ratio at national pove... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| SI.POV.URHC.NC | Zimbabwe | Urban poverty headcount ratio at national pove... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

8140 rows × 48 columns

In [277]:

df\_grouped = df.groupby("Indicator Code")

df\_grouped = df\_grouped.get\_group("SI.POV.GINI")

df\_grouped = df\_grouped.set\_index(["Country Code"])

df\_grouped

Out[277]:

|  | Country Name | Indicator Name | Indicator Code | 1974 | 1975 | 1976 | 1977 | 1978 | 1979 | 1980 | ... | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | Unnamed: 49 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Country Code |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| EAS | East Asia & Pacific | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| ECS | Europe & Central Asia | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| FCS | Fragile and conflict affected situations | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| HIC | High income | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| DFS | IDA countries classified as fragile situations | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| VNM | Vietnam | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 39.3 | NaN | 35.6 | NaN | 34.8 | NaN | 35.3 | NaN | NaN | NaN |
| PSE | West Bank and Gaza | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 35.3 | 34.4 | NaN | NaN | NaN | NaN | 33.7 | NaN | NaN | NaN |
| YEM | Yemen, Rep. | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | 36.7 | NaN | NaN | NaN | NaN | NaN |
| ZMB | Zambia | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 55.6 | NaN | NaN | NaN | NaN | 57.1 | NaN | NaN | NaN | NaN |
| ZWE | Zimbabwe | GINI index (World Bank estimate) | SI.POV.GINI | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | 43.2 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

185 rows × 49 columns

Which number is "the" Gini index for our purposes?[¶](file:///C:\Users\User\Downloads\econ323-final%20(1).html#Which-number-is-"the"-Gini-index-for-our-purposes?)

A glance at the above data frame shows us that, as much as we would have liked to have a Gini index number for every country in the world at every year, the numbers we actually have correspond to irregular observations over the course of 1974 through to 2018. Further, as time marches on, economic, social, and political forces will be acting to change the level of inequality in any given country. The irregularity of the observations may disguise interesting trends within a country, to take one example, or patterns that represent causal forces acting on many countries at once, to take another example.

We need some way to summarize these numbers. For the sake of argument, we will assume that there is no systematic biases effecting when a country could be observed for the sake of this data set, meaning the mean of our observations should be a good estimator of the actual mean Gini index for each country over this period.

While of less intrinsic interest, we also will look at minimum observed Gini indexes and maximum observed Gini indexes. Visualizing these alongside the mean Gini indexes should give a (rough!) intuition of variance.

In [278]:

year\_range = range(1974,2019,1)

country\_codes = df\_grouped.index

minimum\_gini\_series = pd.Series(index=country\_codes, name="Minimum observed GINI index")

maximum\_gini\_series = pd.Series(index=country\_codes, name="Maximum observed GINI index")

mean\_gini\_series = pd.Series(index=country\_codes, name="Mean observed GINI index")

num\_observations\_series = pd.Series(index=country\_codes, name="Number of estimations of GINI index")

for country in country\_codes:

#will be used to compute mean observed gini

successful\_gini\_observations = 0

total\_gini = 0

#conceptually, the Gini index ranges from 0 to 100, these are therefore conceptual extremes of minimum/maximum

minimum\_gini = 100

maximum\_gini = 0

mean\_gini = 0

country\_series = df\_grouped.loc[country]

for year in year\_range:

gini\_this\_year = country\_series.loc[str(year)]

if pd.notna(gini\_this\_year):

successful\_gini\_observations = successful\_gini\_observations + 1

total\_gini = total\_gini + gini\_this\_year

if gini\_this\_year < minimum\_gini:

minimum\_gini = gini\_this\_year

#print(minimum\_gini)

if gini\_this\_year > maximum\_gini:

maximum\_gini = gini\_this\_year

#print(maximum\_gini)

if successful\_gini\_observations > 0:

mean\_gini = total\_gini / successful\_gini\_observations

minimum\_gini\_series.loc[country] = minimum\_gini

maximum\_gini\_series.loc[country] = maximum\_gini

mean\_gini\_series.loc[country] = mean\_gini

num\_observations\_series.loc[country] = int(successful\_gini\_observations)

In [279]:

#simplify the data frame now that we have summary statistics

df\_grouped = df\_grouped.iloc[:,:3]

df\_grouped["Mean GINI"] = mean\_gini\_series

df\_grouped["Min GINI"] = minimum\_gini\_series

df\_grouped["Max GINI"] = maximum\_gini\_series

df\_grouped["Number of observations"] = num\_observations\_series

df\_grouped

Out[279]:

|  | Country Name | Indicator Name | Indicator Code | Mean GINI | Min GINI | Max GINI | Number of observations |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Country Code |  |  |  |  |  |  |  |
| EAS | East Asia & Pacific | GINI index (World Bank estimate) | SI.POV.GINI | 0.000000 | 100.0 | 0.0 | 0.0 |
| ECS | Europe & Central Asia | GINI index (World Bank estimate) | SI.POV.GINI | 0.000000 | 100.0 | 0.0 | 0.0 |
| FCS | Fragile and conflict affected situations | GINI index (World Bank estimate) | SI.POV.GINI | 0.000000 | 100.0 | 0.0 | 0.0 |
| HIC | High income | GINI index (World Bank estimate) | SI.POV.GINI | 0.000000 | 100.0 | 0.0 | 0.0 |
| DFS | IDA countries classified as fragile situations | GINI index (World Bank estimate) | SI.POV.GINI | 0.000000 | 100.0 | 0.0 | 0.0 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

185 rows × 7 columns

Some further data cleaning[¶](file:///C:\Users\User\Downloads\econ323-final%20(1).html#Some-further-data-cleaning)

Some observed countries simply do not have an observed Gini index at any point. These are no good to us for our purposes. We remove these from the data frame.

In [280]:

#For some countries, we simply lack any helpful data about inequality. We can pick these out because Mean GINI is still 0.

for country in country\_codes:

row = df\_grouped.loc[country]

if row.loc["Number of observations"] == 0:

df\_grouped = df\_grouped.drop([country])

df\_grouped

Out[280]:

|  | Country Name | Indicator Name | Indicator Code | Mean GINI | Min GINI | Max GINI | Number of observations |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Country Code |  |  |  |  |  |  |  |
| ALB | Albania | GINI index (World Bank estimate) | SI.POV.GINI | 29.660000 | 27.0 | 31.7 | 5.0 |
| DZA | Algeria | GINI index (World Bank estimate) | SI.POV.GINI | 34.366667 | 27.6 | 40.2 | 3.0 |
| AGO | Angola | GINI index (World Bank estimate) | SI.POV.GINI | 47.350000 | 42.7 | 52.0 | 2.0 |
| ARG | Argentina | GINI index (World Bank estimate) | SI.POV.GINI | 46.141379 | 40.6 | 53.8 | 29.0 |
| ARM | Armenia | GINI index (World Bank estimate) | SI.POV.GINI | 32.255556 | 28.0 | 37.5 | 18.0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| VNM | Vietnam | GINI index (World Bank estimate) | SI.POV.GINI | 36.130000 | 34.8 | 39.3 | 10.0 |
| PSE | West Bank and Gaza | GINI index (World Bank estimate) | SI.POV.GINI | 34.525000 | 33.7 | 35.6 | 8.0 |
| YEM | Yemen, Rep. | GINI index (World Bank estimate) | SI.POV.GINI | 35.466667 | 34.7 | 36.7 | 3.0 |
| ZMB | Zambia | GINI index (World Bank estimate) | SI.POV.GINI | 52.688889 | 42.1 | 60.5 | 9.0 |
| ZWE | Zimbabwe | GINI index (World Bank estimate) | SI.POV.GINI | 43.200000 | 43.2 | 43.2 | 1.0 |

164 rows × 7 columns

Merging the data sets[¶](file:///C:\Users\User\Downloads\econ323-final%20(1).html#Merging-the-data-sets)

'Country Code' is common to both data sets, allowing us to easily perform a merge. We do an inner merge here because that automatically excludes countries (or regions) that are only in one data set or the other. The merged data frame still contains a majority of the countries in the world.

In [281]:

df\_merged = df\_grouped.merge(right=df\_corruption\_index,how='inner',on='Country Code')

df\_merged

Out[281]:

|  | Country Code | Country Name | Indicator Name | Indicator Code | Mean GINI | Min GINI | Max GINI | Number of observations | CPI Rank | Country | Region | Corruption Perceptions Index (CPI) | Standard Error | Lower Confidence Interval | Upper Confidence Interval |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | ALB | Albania | GINI index (World Bank estimate) | SI.POV.GINI | 29.660000 | 27.0 | 31.7 | 5.0 | 83 | Albania | Europe and Central Asia | 39 | 1.99 | 36 | 42 |
| 1 | DZA | Algeria | GINI index (World Bank estimate) | SI.POV.GINI | 34.366667 | 27.6 | 40.2 | 3.0 | 108 | Algeria | Middle East and North Africa | 34 | 2.94 | 29 | 39 |
| 2 | AGO | Angola | GINI index (World Bank estimate) | SI.POV.GINI | 47.350000 | 42.7 | 52.0 | 2.0 | 164 | Angola | Sub-Saharan Africa | 18 | 1.68 | 15 | 21 |
| 3 | ARG | Argentina | GINI index (World Bank estimate) | SI.POV.GINI | 46.141379 | 40.6 | 53.8 | 29.0 | 95 | Argentina | Americas | 36 | 1.76 | 33 | 39 |
| 4 | ARM | Armenia | GINI index (World Bank estimate) | SI.POV.GINI | 32.255556 | 28.0 | 37.5 | 18.0 | 113 | Armenia | Europe and Central Asia | 33 | 4.01 | 26 | 40 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 144 | VEN | Venezuela, RB | GINI index (World Bank estimate) | SI.POV.GINI | 49.307692 | 42.5 | 55.6 | 13.0 | 166 | Venezuela | Americas | 17 | 1.41 | 15 | 20 |
| 145 | VNM | Vietnam | GINI index (World Bank estimate) | SI.POV.GINI | 36.130000 | 34.8 | 39.3 | 10.0 | 113 | Vietnam | Asia Pacific | 33 | 2.46 | 29 | 38 |
| 146 | YEM | Yemen, Rep. | GINI index (World Bank estimate) | SI.POV.GINI | 35.466667 | 34.7 | 36.7 | 3.0 | 170 | Yemen | Middle East and North Africa | 14 | 3.05 | 9 | 19 |
| 147 | ZMB | Zambia | GINI index (World Bank estimate) | SI.POV.GINI | 52.688889 | 42.1 | 60.5 | 9.0 | 87 | Zambia | Sub-Saharan Africa | 38 | 2.91 | 34 | 43 |
| 148 | ZWE | Zimbabwe | GINI index (World Bank estimate) | SI.POV.GINI | 43.200000 | 43.2 | 43.2 | 1.0 | 154 | Zimbabwe | Sub-Saharan Africa | 22 | 2.59 | 18 | 26 |

149 rows × 15 columns

Linear Regressions and Visualizations[¶](file:///C:\Users\User\Downloads\econ323-final%20(1).html#Linear-Regressions-and-Visualizations)

In [282]:

import seaborn as sns

from sklearn import linear\_model

linear\_regressor = LinearRegression()

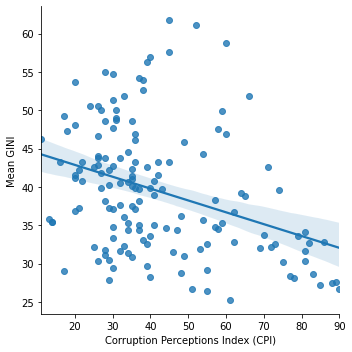
x\_corruption = df\_merged["Corruption Perceptions Index (CPI)"]

y\_gini = df\_merged["Mean GINI"]

sns.lmplot(data = df\_merged, x = "Corruption Perceptions Index (CPI)", y = "Mean GINI")

Out[282]:

<seaborn.axisgrid.FacetGrid at 0x7f671308eac8>

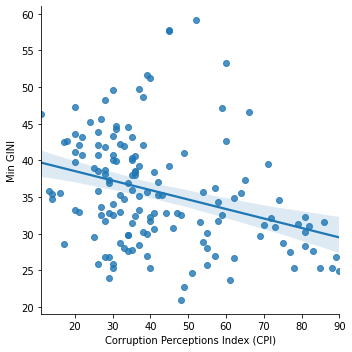


In [283]:

sns.lmplot(data = df\_merged, x = "Corruption Perceptions Index (CPI)", y = "Min GINI")

Out[283]:

<seaborn.axisgrid.FacetGrid at 0x7f671304d908>

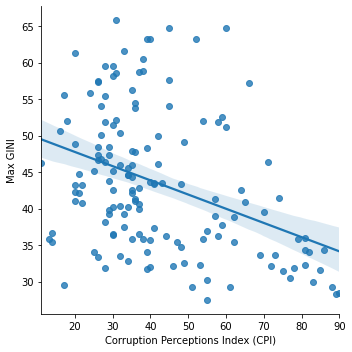


In [284]:

sns.lmplot(data = df\_merged, x = "Corruption Perceptions Index (CPI)", y = "Max GINI")

Out[284]:

<seaborn.axisgrid.FacetGrid at 0x7f6712f73ef0>



What do these graphs mean?[¶](file:///C:\Users\User\Downloads\econ323-final%20(1).html#What-do-these-graphs-mean?)

CONCLUSION

The Gini index is often described as ranging from a score of 0, which represents a perfectly egalitarian economy with the income or wealth of every person in the economy is exactly equal, to a score of 100, which represents an economy where all the income or wealth goes to a single person and none goes to anyone else. Thus, a lower score is indicative of a more egalitarian economy, and a higher score is indicative of a less egalitarian. The Corruption Perception Index, however, works on the "high score is good" intuition. Low scorers are perceived as corrupt, high scorers are perceived as not corrupt.

Therefore, given the empirical theories mentioned in the introduction about corruption and rent seeking causing inequality, we would predict there to be a negative relationship - which is exactly what our linear regression says we do predict. The slope (though not the intercept, obviously) is even roughly the same regardless of which representative Gini index number we use.

This is of minimum value for confirming the causal hypotheses discussed at the beginning of this report, but it is consistent with those hypotheses in such a way so as to suggest to us they are on the right track.

Economic developments and political liberalizations has presented cohesion in the realization of both national and global economic and political realizations. In these instances, it is clear that the emergence of new democracies have been associated with the betterment of the global politics for the sake of different citizens around the world. Changes in the forms of governance have led to variation in political liberations around the world. Despite the attempts to promote better forms of leadership, the absence of liberations have led to a decline in the economic development over time. In this case, it was noted that not opening countries for international business thwarted economic developments over the past years. Furthermore, economic liberalizations is associated with economic diversification, creating more avenues for business growth across the respective nations around the world over the years.

Political instability has been associated with negative economic performance across the global economies. For instance, wars and other civil strife have led to non-conducive environments around the world. These challenges have been experienced in countries such as Somalia, Iraq, Pakistan and Syria among others. During periods of civil unrests, economic liberalization faces greater challenges in terms of failure to promote better economic environments for business, either from the governments or the private sector. Moreover, concerns on the negative economic performance may result in affected countries ending up borrowing more to service their operations from international lenders at the IMF and World Bank. Most these concerns have always raised the debt levels of such countries in a bid to protect and enhance their national operations over time.

Countries with evidence of economic liberalizations and political stability offer better business environments for business to thrive over specified periods. Such cases have been witnessed across developing countries such the US, China, Japan and Germany among others, promoting significant levels of economic developments. These developed countries have benefited significantly from economic liberalizations by opening their economies for international investors. Furthermore, business have also demonstrated the ability to grow faster, with the realization of conducive business environments, improving the economic performance measures of the relevant countries. It is clear that economic development has a positive association with political liberalizations in the context of favorable policies to steer existing businesses to the next levels in terms of profitability. In this case, countries have been advised to adopt both economic and political liberalization to better their economic prospects into the future.

References

Adegboye, F. B., Matthew, O. A., Ejemeyovwi, J., Adesina, O. S., & Osabohien, R. (2020). Assessing the role of trade liberalization in facilitating trade flows and economic expansion: evidence from ECOWAS countries. In *Fostering Trade in Africa* (pp. 97-109). Springer, Cham.

Amodio, F., Baccini, L., Chiovelli, G., & Di Maio, M. (2020). *Trade liberalization and political violence: evidence from North-South preferential trade agreements* (No. 2020/41). European University Institute.

Bjarnegård, E. (2020). Introduction: Development Challenges in Myanmar: Political Development and Politics of Development Intertwined.

Busemeyer, M., & Kemmerling, A. (2020). Dualization, stratification, liberalization, or what? An attempt to clarify the conceptual underpinnings of the dualization debate. *Political Science Research and Methods*, *8*(2), 375-379.

Deniz Genç, H., & Sabah Eryılmaz, G. (2020). Politics of real estate internationalization: contested liberalization in Turkey. *Turkish Studies*, 1-28.

Ding, H., Jin, Y., Koedijk, K. G., & Wang, Y. (2020). Valuation effect of capital account liberalization: Evidence from the Chinese stock market. *Journal of international money and finance*, *107*, 102208.

Estrada, M. A. R. (2020). From World Trade Liberalization to World Trade Emancipation: COVID-19.

Farzanegan, M. R., Hassan, M., & Badreldin, A. M. (2020). Economic liberalization in Egypt: A way to reduce the shadow economy?. *Journal of Policy Modeling*, *42*(2), 307-327.

Haddad, B. S. (2020). 5. The Political Dynamics of Economic Liberalization. In *Business Networks in Syria* (pp. 119-153). Stanford University Press.

Haruna, A. A., & Bakar, A. S. A. (2020). Interest rate liberalization and economic growth nexus: does corruption matter?. *Journal of Financial Crime*.

Hossain, M. (2020). Financial Liberalization, Financial Development and Economic Growth: An Analysis of the Financial Sector of Bangladesh. *Bangladesh's Macroeconomic Policy*, 395-418.

Hummel, S. (2020). Leader age, death, and political liberalization in dictatorships. *The Journal of Politics*, *82*(3), 981-995.

Ikhsan, O. F., Islam, R., Khamis, K. A., & Sunjay, A. (2020). Impact of digital economic liberalization and capitalization in the era of industrial revolution 4.0: case study in Indonesia. *Problems and Perspectives in Management*, *18*(2), 290.

Khayitboy, K., & Ilhom, R. (2020). The impact of liberalization on the development of the social system. *International Engineering Journal For Research & Development*, *5*(3), 4-4.

Li, J., Xi, T., & Yao, Y. (2020). Empowering knowledge: Political leaders, education, and economic liberalization. *European Journal of Political Economy*, *61*, 101823.

Murakami, Y. (2021). Trade liberalization and wage inequality: evidence from Chile. *The Journal of International Trade & Economic Development*, *30*(3), 407-438.

Nguyen, V. C. (2020). Trade liberalization, economic reforms and foreign direct investment–A critical analysis of the political transformation in Vietnam. *Economic Reforms and Foreign Direct Investment–A Critical Analysis of the Political Transformation in Vietnam (April 2, 2020)*.

Simoni, M., & Vlandas, T. (2020). Labour market liberalization and the rise of dualism in Europe as the interplay between governments, trade unions and the economy. *Social Policy & Administration*.

Valeri, M. (2020). 9. State Building, Liberalization from Above, and Political Legitimacy in the Sultanate of Oman. In *Debating Arab Authoritarianism* (pp. 143-158). Stanford University Press.

Zhang, T. H. (2020). Political freedom, education, and value liberalization and deliberalization: A cross-national analysis of the world values survey, 1981-2014. *The Social Science Journal*, 1-18.