Is it possible to test the accuracy of the Human Development Index by testing correlation against other economic indicators?

Adam Curry - 9452

An investigation presented for a non-examined assessment

Computer Science

Godalming College - 64395

United Kingdom

09/09/19

Contents

Table of Figures

Research

Overview of problem

The problem that I will be investigating is: Is it possible to test the accuracy of the Human Development Index (HDI) by testing correlation against other economic indicators?

The problem that I will be investigating seeks to discover if that figure is actually accurate, by comparing it to other data that is available, such as inequality or poverty levels. These comparisons will come in the form of correlations, as if the two factors have a strong correlation it can be concluded that the HDI is an accurate measure.

3rd Party Reviewer

Interview was conducted with Courtney Anderson, a teacher and Head of Economics at Godalming, in order to discuss information existing currently, and what HDI is currently used for. This occurred in college time, at Godalming College premises.

Interview

Are you aware of what the Human Development Index is?

Yes. It's the Human Development Index. It's a measure of economic development in a country. it's a crude shorthand way of creating a hierarchy of the 'best' and 'worst performing' countries.

Is it useful to use recent data to predict the future?

Yes, if you mean that it is possible to take data on total inward investment or aid or some other variable and then expect a correlation between that and a future HDI score.

Yes, if you mean that does an improvement in HDI possibly lead to higher FDI [foreign direct investment- which is where foreign companies by shares in UK companies].

Are there any methods used to analyse data in the economics a-level course?

No, data analysis is mainly done in statistics as a part of A-level maths.

How useful is it to present data graphically to help understanding in economics?

Very- graphical methods can be used to show numerous different relationships between economic factors and are commonly used to show relationships between different factors as a part of the A-level course.

Background Information

United Nations

Founded in 1945, the United Nations is an organization which is made up of 193 member states with the goal of it to aid international cooperation to solve the issues facing the human race, such as climate change, sustainable development and humanitarian emergencies.

As a part of this goal, the organization collects large amounts of data in order to inform its decision making. This data is on a variety of predominantly human factors, categorized by country, as it works on a large scale.

World Bank

The World Bank is an inter-governmental organisation (IGO) that's sole purpose is to end extreme poverty and boost income growth for the most disadvantaged within each of the individual countries in which it works. They provide extremely low interest loans in order to aid developing countries and allow them to improve living conditions. They also extend beyond the ordinary workings of a bank as they offer policy advice, research and analysis for developing nations in order to benefit political decision making, again to aid the poorest 40%.

Human Development Index

The Human development index is a weighted index created by the United Nations in order to make it easier to compare the different socioeconomic levels between countries. It was created to be the ultimate criteria for assessing development of a country, not just economic growth alone. According to the United Nations website "It was created to be the ultimate criteria for assessing development of a country, not just economic growth alone.".

An index is where multiple different indicators are added together and divided by a number in order to calculate a single value. A weighted index is important as is attaches a proportional significance to each individual factor, so more important factors can have a larger effect on the outcome.

The index is created using three different factors; life expectancy, which is the age that the average person is expected to live until at birth; education levels, which is comprised of expected years of schooling and mean years of schooling in the population; and standard of living, measured by gross national income per capita, which is the average salary of a person in that nation.

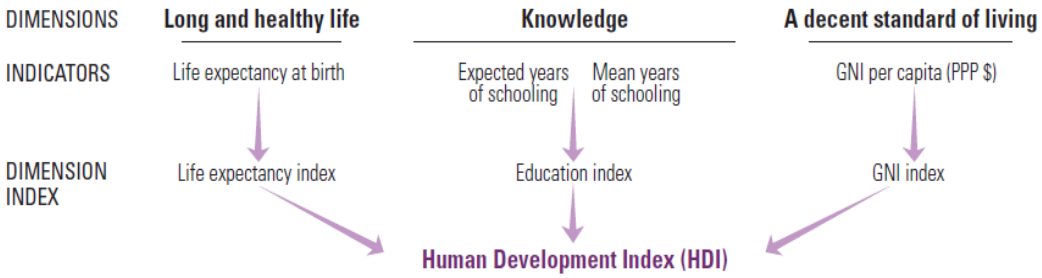


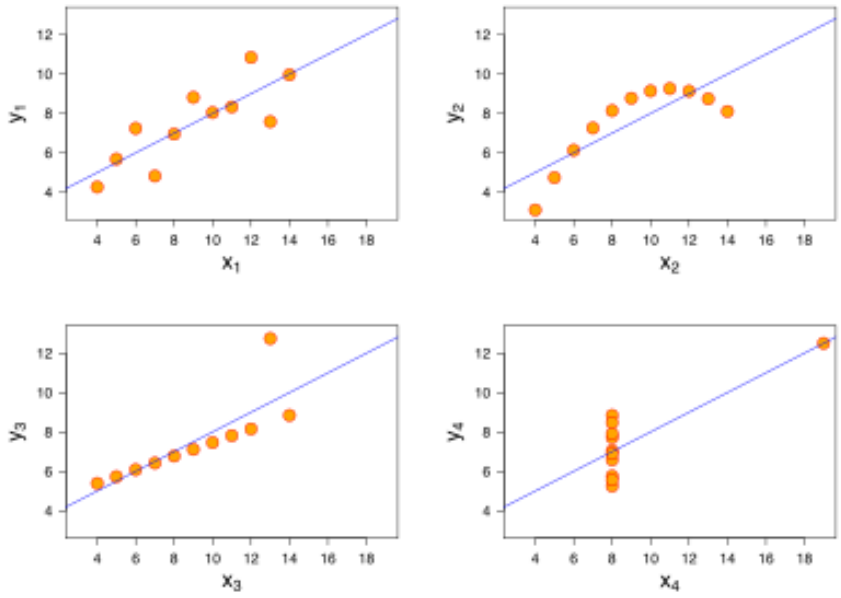
Figure 1- Human Development Index Creation - http://hdr.undp.org/en/content/human-development-index-hdi

Correlation

Correlation is a way to describe a relationship between two or more variables, and the degree to which the variables are associated with one another.

From the research which I have undertaken, there are three primary methods which are used in general cases to try to understand correlation. There are two mathematical methods, which are Pearson's Coefficient and Spearman's Rank; but the graphical method of scatter diagrams can also be used to highlight the degree to which variables are associated. Out of these methods, I would argue that the best one is scatter diagrams, as the other mathematical equations are limited to linear correlations, which is where there is a straight-line relationship, where the variables always change by the same amount. However, although this is the most common form of correlation, other relationships such as polynomial or exponential ones can occur, which are overlooked by the mathematical models.

Figure 2 - Scatter Graphs with different correlations - but same coefficient of 0.816



Scatter graphs are constructed by placing the independent variable on the X axis and the dependent variable on the Y axis, then the data which has been collected is plotted according to its X and Y values. The only issue with scatter graphs is that they rely heavily on subjectivity as the person viewing them makes the decision on whether there is correlation, and how strong it is, which may differ from person to person.

Pearson's Coefficient, which has the full name Pearson Product-Moment Correlation which is why sometimes it as referred to as the product moment, only measures linear association, and is given by the formula below.

Figure 3- Pearson's Product-Moment Correlation Coefficient Formula

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[\, n\sum x^2 - (\sum x)^2 \,][\, n\sum y^2 - (\sum y)^2 \,]}}$$

This formula will always produce a decimal value in the range between -1 and 1 inclusive. The more polarised it is, as in the further away from the zero that the result is, the stronger the correlation between the two variables which were inputted.

Spearman's Rank is the final mathematical way of calculating the strength of a correlation and is commonly used in more practical fields such as geography, whereas Pearson's coefficient tends to be used more in theoretical mathematics.

Figure 4- Spearman's Rank Formula

$$r_s = 1 - \frac{6\sum D^2}{n\left(n^2 - 1\right)}$$

This formula is used to measure non-parametric correlations, therefore although it is less accurate for linear correlation than Pearson's Coefficient, it can at least measure other correlations, such as exponential, but is still not as complete as scatter graphs. Spearman's rank also requires data to be ordered, therefore an algorithm will be needed to sort the data into numerical order. This could be attained through a Bubble Sort or a Merge Sort. Once the data has been ranked, the Pearson's correlation coefficient of these ranked arrays can be found, or the formula above can be used to find the spearman's rank.

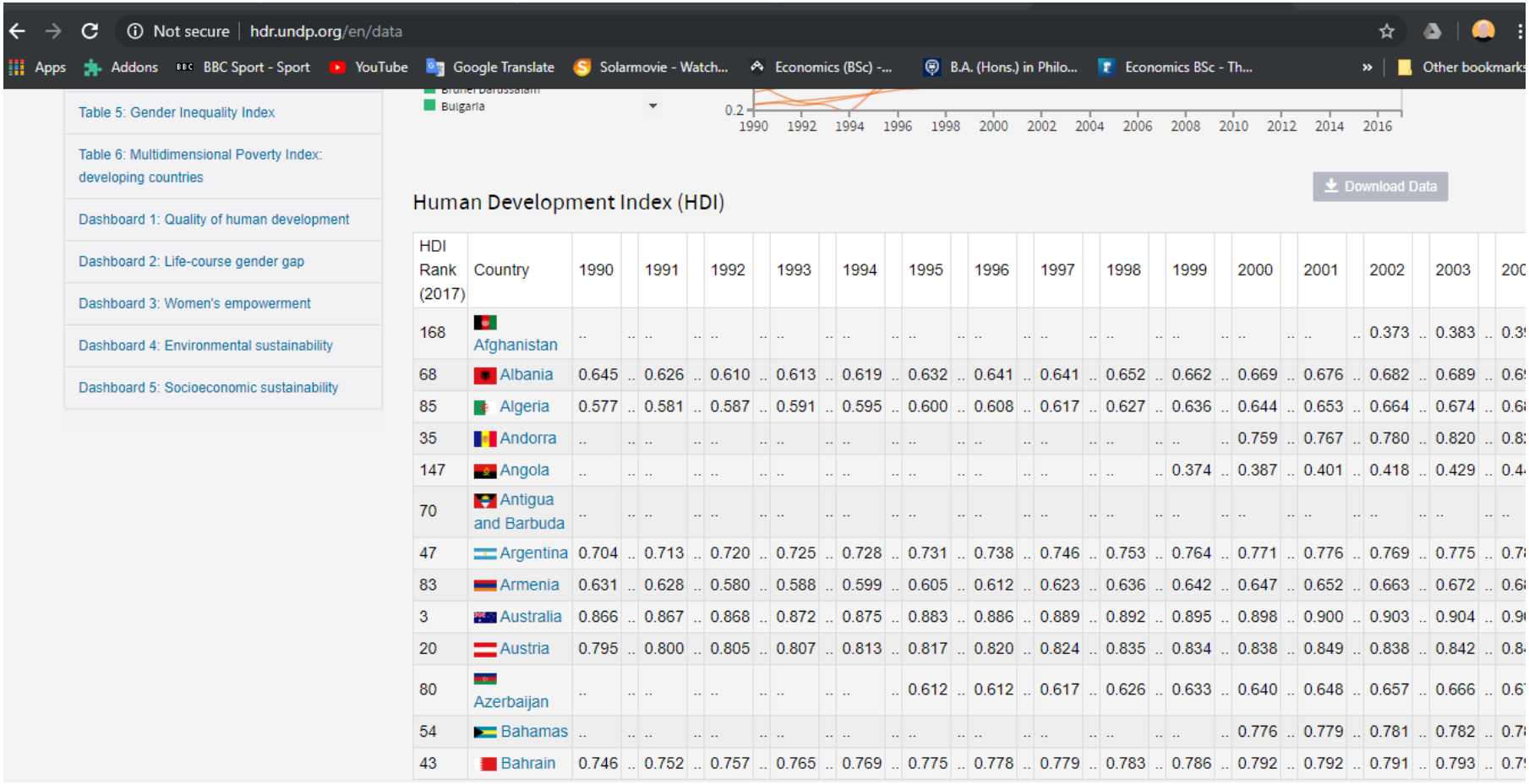
Reasons for Choice

The reason that I chose this investigation is that I am extremely interested in economics as I am studying it at A level at the moment and am planning on continuing to study it at degree level in the future. I spoke to Courtney and he recommended that I create a program like this as manipulating the data is a useful tool to learn and can be really useful in drawing economic conclusions.

The Human Development Index has a large effect on the world, therefore it is interesting to see how accurate it actually is, as it is used to create global macroeconomic policy to improve societal welfare.

Data

Figure 5 - United Nations Website Data



Not secure | hdr.undp.org/en/data

Apps | Addons | BBC Sport - Sport | YouTube | Google Translate | Solarmovie - Watch... | Economics (BSc) -... | B.A. (Hons.) in Philo... | Economics BSc - Th... | Other bookmarks

Table 5: Gender Inequality Index

Table 6: Multidimensional Poverty Index: developing countries

Dashboard 1: Quality of human development

Dashboard 2: Life-course gender gap

Dashboard 3: Women's empowerment

Dashboard 4: Environmental sustainability

Dashboard 5: Socioeconomic sustainability

Brunei Darussalam

Bulgaria

0.2

1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016

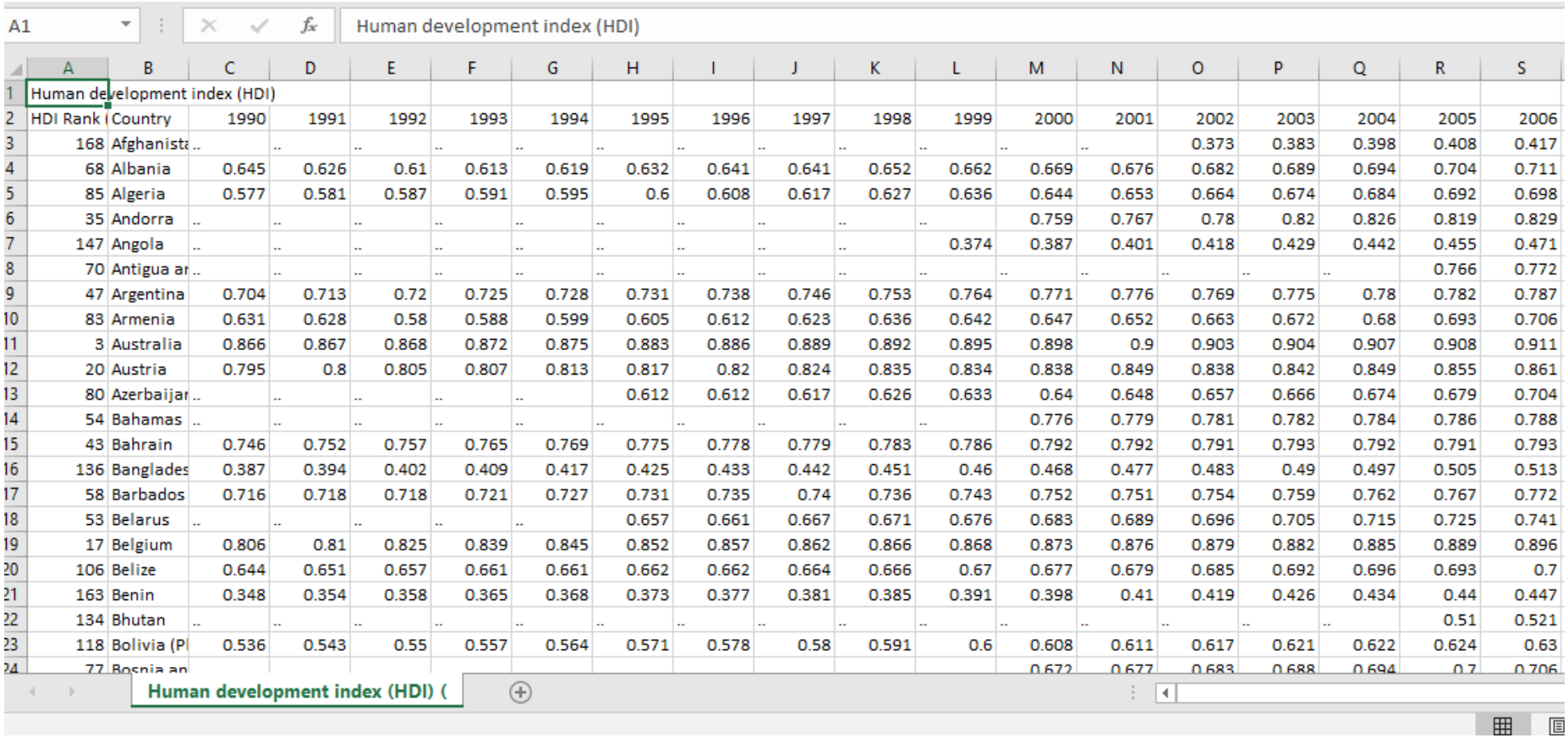
Download Data

Human Development Index (HDI)

HDI Rank (2017)	Country	1990		1991		1992		1993		1994		1995		1996		1997		1998		1999		2000		2001		2002		2003		200
168	Afghanistan	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0.373	..	0.383	..	0.3
68	Albania	0.645	..	0.626	..	0.610	..	0.613	..	0.619	..	0.632	..	0.641	..	0.641	..	0.652	..	0.662	..	0.669	..	0.676	..	0.682	..	0.689	..	0.6
85	Algeria	0.577	..	0.581	..	0.587	..	0.591	..	0.595	..	0.600	..	0.608	..	0.617	..	0.627	..	0.636	..	0.644	..	0.653	..	0.664	..	0.674	..	0.6
35	Andorra	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0.759	..	0.767	..	0.780	..	0.820	..	0.8
147	Angola	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0.374	..	0.387	..	0.401	..	0.418	..	0.429	..	0.4
70	Antigua and Barbuda	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
47	Argentina	0.704	..	0.713	..	0.720	..	0.725	..	0.728	..	0.731	..	0.738	..	0.746	..	0.753	..	0.764	..	0.771	..	0.776	..	0.769	..	0.775	..	0.7
83	Armenia	0.631	..	0.628	..	0.580	..	0.588	..	0.599	..	0.605	..	0.612	..	0.623	..	0.636	..	0.642	..	0.647	..	0.652	..	0.663	..	0.672	..	0.6
3	Australia	0.866	..	0.867	..	0.868	..	0.872	..	0.875	..	0.883	..	0.886	..	0.889	..	0.892	..	0.895	..	0.898	..	0.900	..	0.903	..	0.904	..	0.9
20	Austria	0.795	..	0.800	..	0.805	..	0.807	..	0.813	..	0.817	..	0.820	..	0.824	..	0.835	..	0.834	..	0.838	..	0.849	..	0.838	..	0.842	..	0.8
80	Azerbaijan	..	..	..	..	..	..	..	..	..	..	0.612	..	0.612	..	0.617	..	0.626	..	0.633	..	0.640	..	0.648	..	0.657	..	0.666	..	0.6
54	Bahamas	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0.776	..	0.779	..	0.781	..	0.782	..	0.7
43	Bahrain	0.746	..	0.752	..	0.757	..	0.765	..	0.769	..	0.775	..	0.778	..	0.779	..	0.783	..	0.786	..	0.792	..	0.792	..	0.791	..	0.793	..	0.7

Etc… This is the data presented on the United Nation's website, however, this is not easily accessible, and so will not be that useful, so the download data link in the top left of the image is more useful.

Figure 6- United Nations Data on Human Development Index - CSV



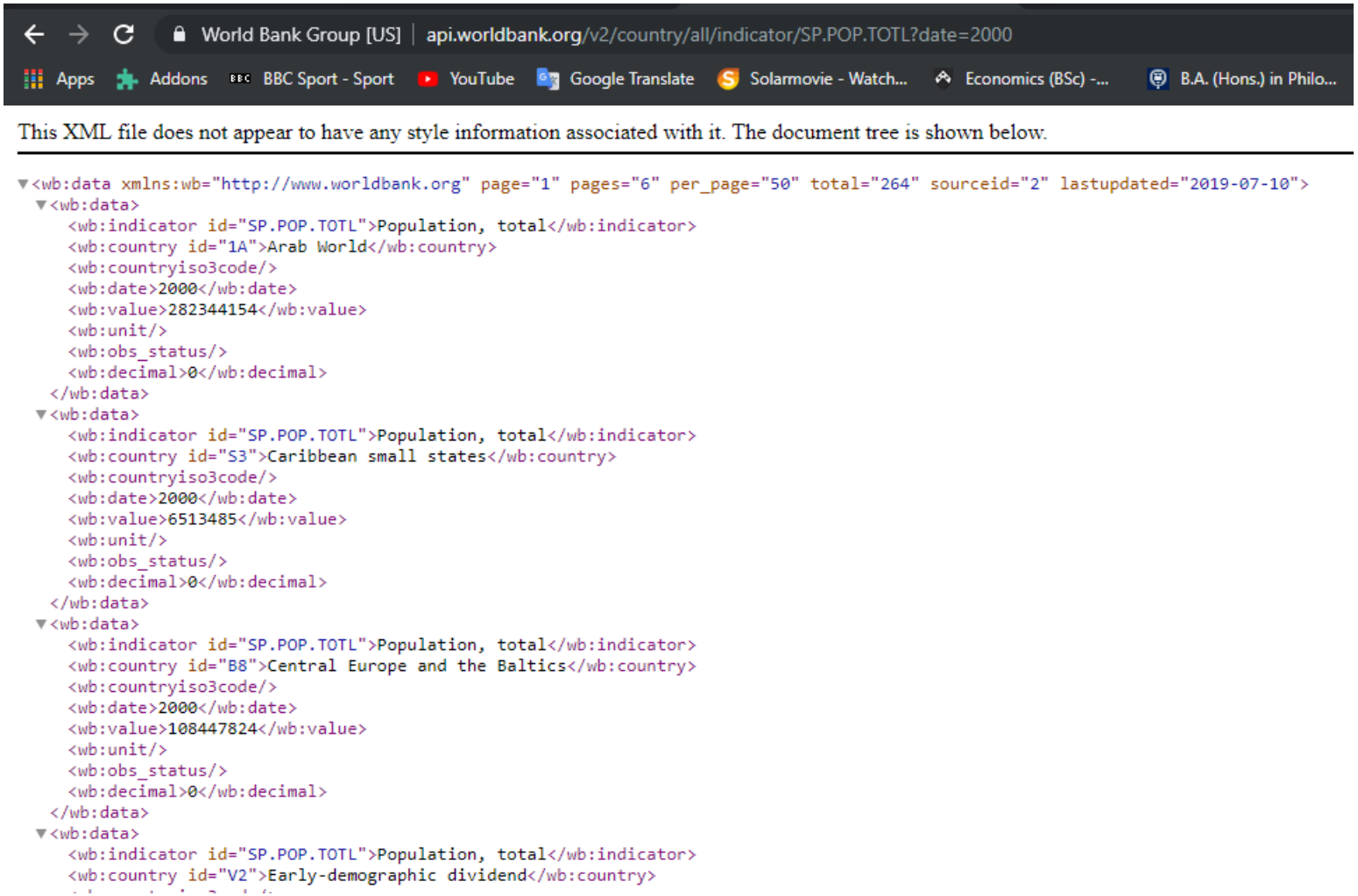
A1 | Human development index (HDI)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Human development index (HDI)																		
2	HDI Rank	Country	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006
3	168	Afghanista	..	..	..	..	..	..	..	..	..	..	..	..	0.373	0.383	0.398	0.408	0.417
4	68	Albania	0.645	0.626	0.61	0.613	0.619	0.632	0.641	0.641	0.652	0.662	0.669	0.676	0.682	0.689	0.694	0.704	0.711
5	85	Algeria	0.577	0.581	0.587	0.591	0.595	0.6	0.608	0.617	0.627	0.636	0.644	0.653	0.664	0.674	0.684	0.692	0.698
6	35	Andorra	..	..	..	..	..	..	..	..	..	..	0.759	0.767	0.78	0.82	0.826	0.819	0.829
7	147	Angola	..	..	..	..	..	..	..	..	..	0.374	0.387	0.401	0.418	0.429	0.442	0.455	0.471
8	70	Antigua ar	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0.766	0.772
9	47	Argentina	0.704	0.713	0.72	0.725	0.728	0.731	0.738	0.746	0.753	0.764	0.771	0.776	0.769	0.775	0.78	0.782	0.787
10	83	Armenia	0.631	0.628	0.58	0.588	0.599	0.605	0.612	0.623	0.636	0.642	0.647	0.652	0.663	0.672	0.68	0.693	0.706
11	3	Australia	0.866	0.867	0.868	0.872	0.875	0.883	0.886	0.889	0.892	0.895	0.898	0.9	0.903	0.904	0.907	0.908	0.911
12	20	Austria	0.795	0.8	0.805	0.807	0.813	0.817	0.82	0.824	0.835	0.834	0.838	0.849	0.838	0.842	0.849	0.855	0.861
13	80	Azerbaijar	..	..	..	..	..	0.612	0.612	0.617	0.626	0.633	0.64	0.648	0.657	0.666	0.674	0.679	0.704
14	54	Bahamas	..	..	..	..	..	..	..	..	..	..	0.776	0.779	0.781	0.782	0.784	0.786	0.788
15	43	Bahrain	0.746	0.752	0.757	0.765	0.769	0.775	0.778	0.779	0.783	0.786	0.792	0.792	0.791	0.793	0.792	0.791	0.793
16	136	Banglades	0.387	0.394	0.402	0.409	0.417	0.425	0.433	0.442	0.451	0.46	0.468	0.477	0.483	0.49	0.497	0.505	0.513
17	58	Barbados	0.716	0.718	0.718	0.721	0.727	0.731	0.735	0.74	0.736	0.743	0.752	0.751	0.754	0.759	0.762	0.767	0.772
18	53	Belarus	..	..	..	..	..	0.657	0.661	0.667	0.671	0.676	0.683	0.689	0.696	0.705	0.715	0.725	0.741
19	17	Belgium	0.806	0.81	0.825	0.839	0.845	0.852	0.857	0.862	0.866	0.868	0.873	0.876	0.879	0.882	0.885	0.889	0.896
20	106	Belize	0.644	0.651	0.657	0.661	0.661	0.662	0.662	0.664	0.666	0.67	0.677	0.679	0.685	0.692	0.696	0.693	0.7
21	163	Benin	0.348	0.354	0.358	0.365	0.368	0.373	0.377	0.381	0.385	0.391	0.398	0.41	0.419	0.426	0.434	0.44	0.447
22	134	Bhutan	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0.51	0.521
23	118	Bolivia (P	0.536	0.543	0.55	0.557	0.564	0.571	0.578	0.58	0.591	0.6	0.608	0.611	0.617	0.621	0.622	0.624	0.63
24	77	Bosnia an											0.672	0.677	0.683	0.688	0.694	0.7	0.706

Human development index (HDI) (

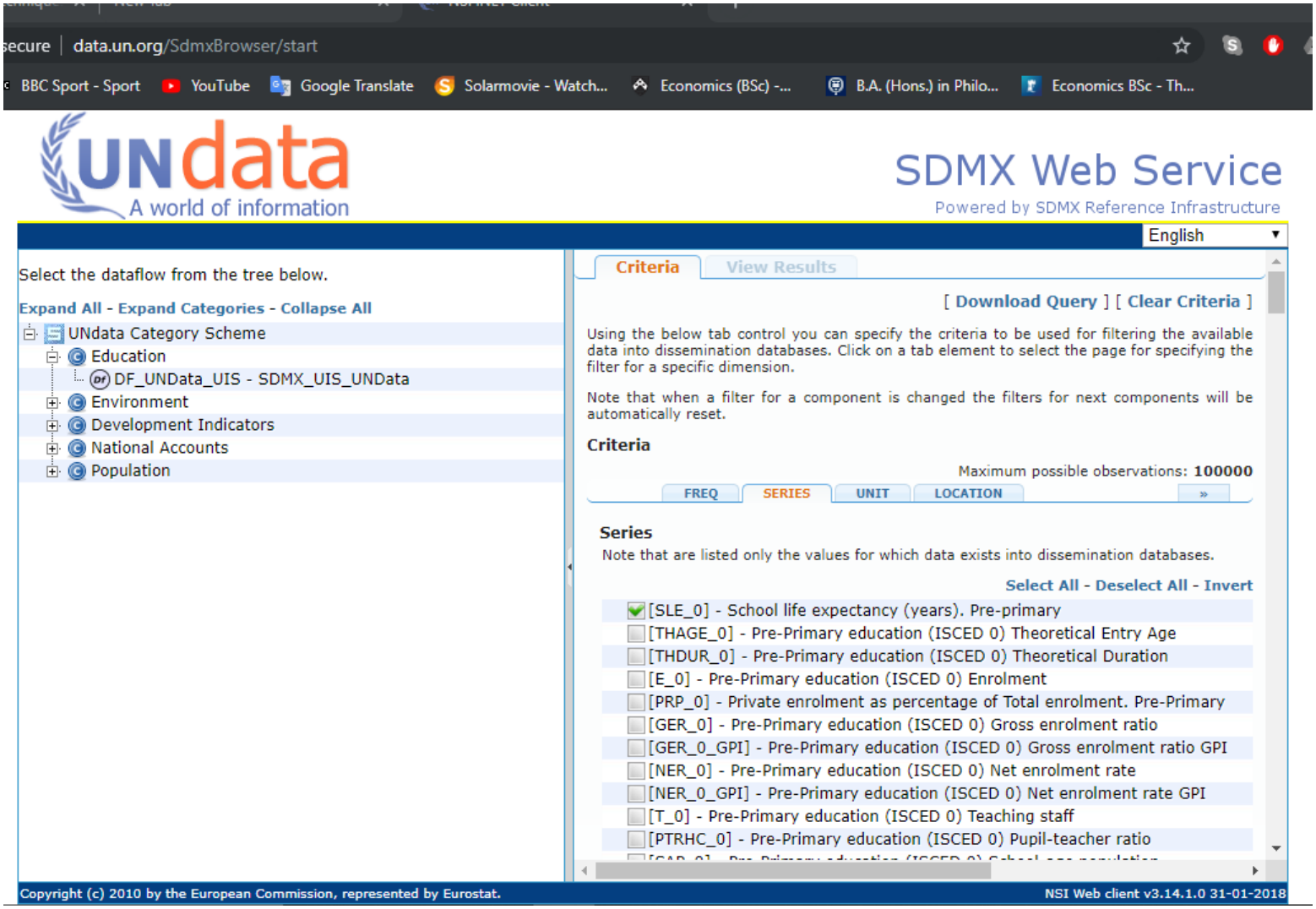
Etc… This file can be downloaded directly from the United Nation's website and is updated regularly. It is a csv file, which means it is stored locally and is easily accessible for a program. The columns contain data arranged by date, and the rows show the data arranged by country. The first column also provides a rank for each country, giving the numbers context in comparison to other nations.

Figure 7- World Bank Data on Economic Indicator – API – Data shown is Population



Etc… This data is presented in an XML file format and uses clear and easy to read tags in order to describe the data stored in the file.

Figure 8- United Nations API - SDMX Web Service



Etc…. these dataflows are presented in a simple way for an individual to access online but have little support or information on how to use the API feature.

Reading Review

Source	Name of Source	Date of Access	Summary of Source	Used	Reliability
Web	VivaxSolutions https://www.vivaxsolutions.com/maths/ul-index-numbers.aspx	11/09/19	This source is an overview of index numbers and explains in mathematical terms how they can be calculated.	Y	This source is just factual mathematics so can be considered reliable.
Web	United Nations http://hdr.undp.org/en/content/human-development-index-hdi	11/09/19	This source explained the fundamentals used in creating the HDI.	Y	This source is directly from the UN so is accurate.
Web	United Nations http://hdr.undp.org/en/data	15/09/19	This source contains data only on the Human Development Index.	N	N/A
Web	World Bank - API https://datahelpdesk.worldbank.org/knowledgebase/articles/898581-api-basic-call-structures	15/09/19	This source contains links to a variety of data sets with an API allowing programs to interact with it.	N	N/A
Web	World Bank https://www.worldbank.org/en/about/what-we-do	15/09/19	This source contains what the world bank's key aims are and how they go about reaching them.	Y	This source is likely to be factual, however, may be slightly bias in portraying the organisation in a positive light.
Web	Businessjargons https://businessjargons.com/methods-of-determining-correlation.html	15/09/19		Y	
Web	Wikiversity https://en.wikiversity.org/wiki/Correlation	15/09/19		Y	

Analysis

Current Systems

Currently, there is no available software which does what my program is aiming to achieve. The existing software which is available is primarily focused on showing individual graphs and trends overtime. For example there are numerous websites available online such as https://www.macrotrends.net/charts/economy , where a variety of graphs can be found showing macroeconomic indicators over time; as shown in the screenshot below.

Figure 9- Screenshot of https://www.macrotrends.net/charts/economy

However, this software does not allow for any personalisation or creation of your own graphs, showing the relationship between variables, they are focused primarily on one indicator, and the time period. These graphs are useful for analysing macroeconomic policy impacts in certain countries, but software like this is not useful for the investigation which I am doing.

There is one alternative program which is currently available, which achieves a similar outcome to my planned software, however the program itself I found to be extremely limiting.

Name: Gapminder Tools

Web Address: https://www.gapminder.org/

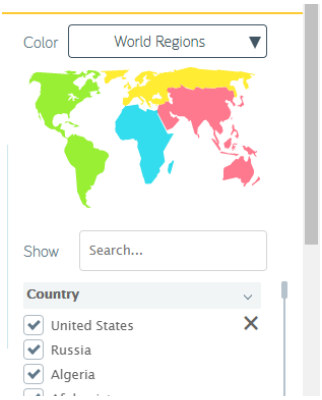


Figure 10 - Screenshot 1 of: https://www.gapminder.org/

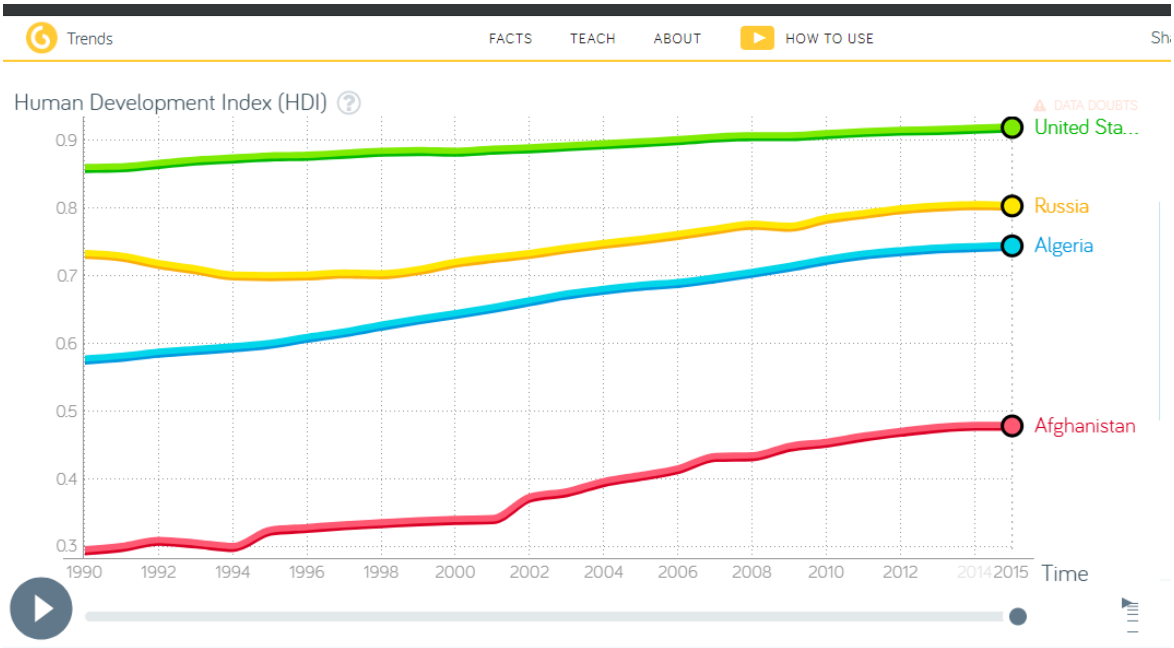


Figure 11 - Screenshot 2 of: https://www.gapminder.org/

As shown by the diagrams, this piece of software is controlled by a simple GUI, and can be used to access United Nations data, on indicators such as the Human Development Index, and map them graphically, presented in a clear way to the user. The program allows as many countries as the user specifies to be put onto the graph and has hundreds of different economic indicators which can be assigned to the Y axis of the graph, which is similar to the software which I am designing.

However, this software is lacking a number of critical elements, which would make it better.

The first glaring issue is the inability to remap the X axis to a variable other than time. This means that the use of this program is limited to comparing different countries, whereas if the X axis could be changed to something such as inequality it would allow individuals to see correlation and areas of improvement, as well as gain a more complete picture of whatever the user is investigating.

Secondly, this program does not contain any mathematical measures to aid the user in coming to a conclusion. Mathematical calculations can be introduced in order to eliminate value judgements and personal bias, which would make any piece of software which handles data more accurate as well as getting more precise results, based in evidence rather than opinion. However, linking back to the last limitation, as comparison is only restricted to time, there is no need for this program to have any calculation features, so it is understandable to see why they have not included them for this individual piece of software.

Below, is a flowchart which maps out the current system that a user would do in order to do what my program seeks to investigate:

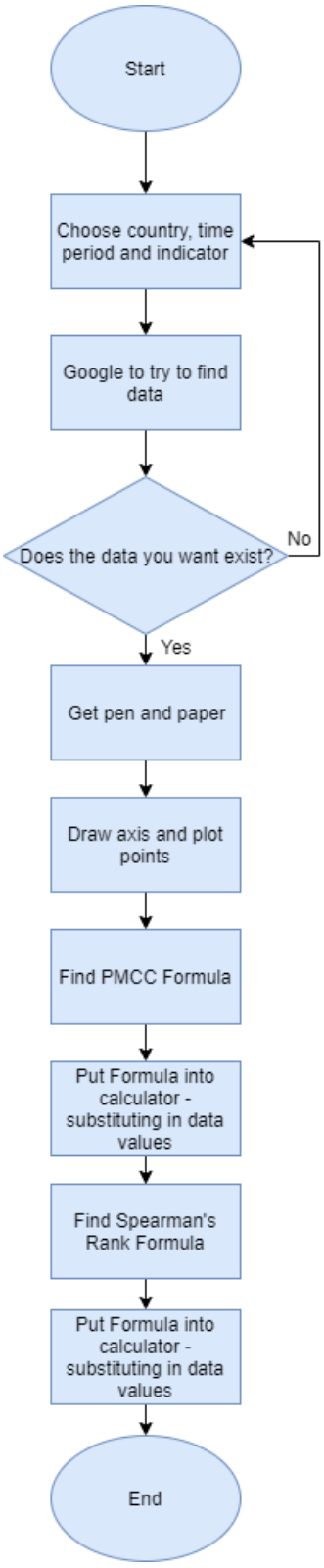


Figure 12- Flow chart of Current System

Issues with the current system

There are clearly numerous issues with the current system that someone undertaking my investigation would encounter. Firstly, there is a large risk of human error. As the user has to research, calculate and draw graphs, all with no assistance, there is a high probability that they will make a mistake somewhere, for example miscopying a number, skewing results and making it a less accurate and reliable investigation.

Secondly an issue is also the fact that it is inefficient. This process takes an extremely long time, which is a large issue. When deciding on which data to use, the user does not get immediate feedback on whether or not it is available, therefore they may spend a large amount of time trying to find data which does not exist, therefore my program will need to reduce the inefficiencies to improve the outcome of the investigation.

IPSO Diagram

An IPSO diagram is key, as it allows us to go over our research and select the most important aspects which will need to be implemented in the program.

Inputs

- Country of data (User)
- Economic indicator of data (User)
- Time period of data (User)

Outputs

- Graph (User)
- Product Moment Correlation Coefficient (User)
- Spearman's Rank Number (User)

Processes

- Fetch requested data from CSV
- Create request for user data
- Obtain data using request from API
- Produce graph from data
- Calculate PMCC from data
- Calculate Spearman's Rank from data

Storage

- Current data options shown
- Previous results stored

UML Data Flow Diagrams

A data flow diagram is a way of representing how data will move, or flow, through a computer system. There are different levels of data-flow diagrams, with higher levels being more detailed, having been abstracted greater than lower levelled diagrams. I also provide information about the outputs and inputs to a system, as well as a brief overview of how the processing in the system will work, which is why it is an effective tool to analyse a project.

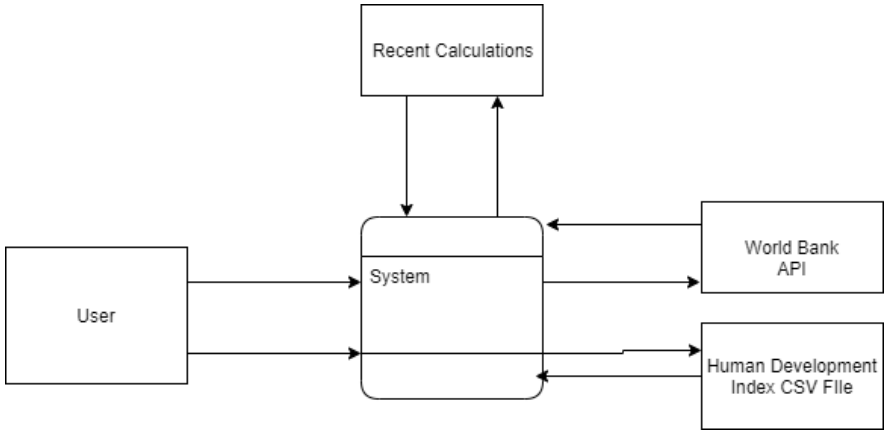


Figure 13 - Data Flow Diagram Level 0

Above is a level zero data flow diagram, showing the system as a whole, it is also known as a context diagram, as it shows only the basic input and outputs, with all processes collected in the System process. It emphasizes the programs interactions with external entities.

A level one data flow diagram, as shown in Figure 9 below is more detailed, including a breakdown of all the transfers of data that the system contains, as well as all data storage points. For the system that I am creating, most data will be dynamic and stored in temporary variables, therefore that data is not represented on the diagram below.

A level two diagram, could be create, but that does not show more flows of data, it recreates the processes occurring in more detail, however, I believe that flowcharts have the ability to show step-by-step algorithms and processes in a clearer, easier to read and create format, therefore, there is no need to produce a data flow diagram beyond a level one diagram.

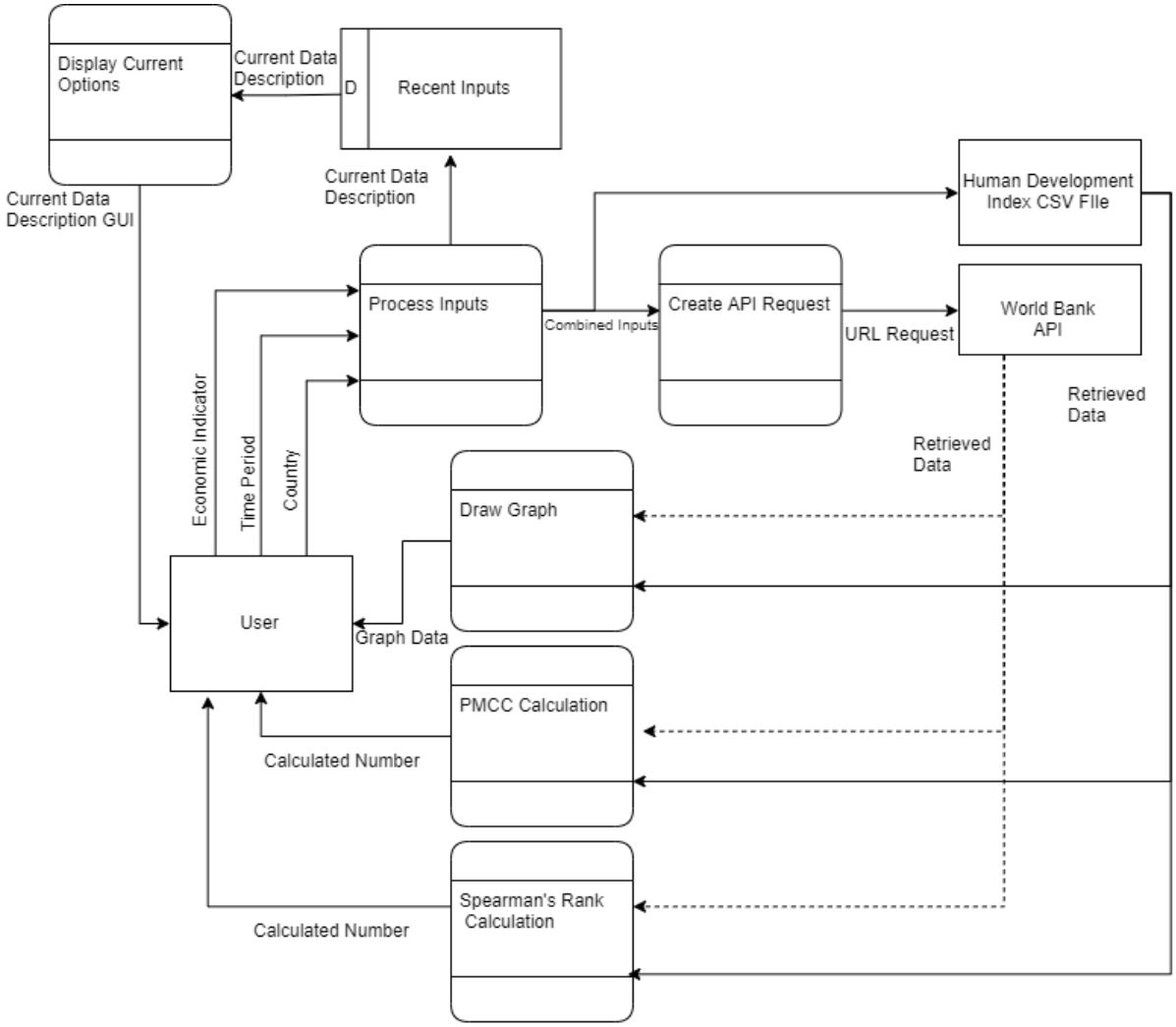


Figure 14 - Data Flow Diagram Level 1

UML Object Diagram

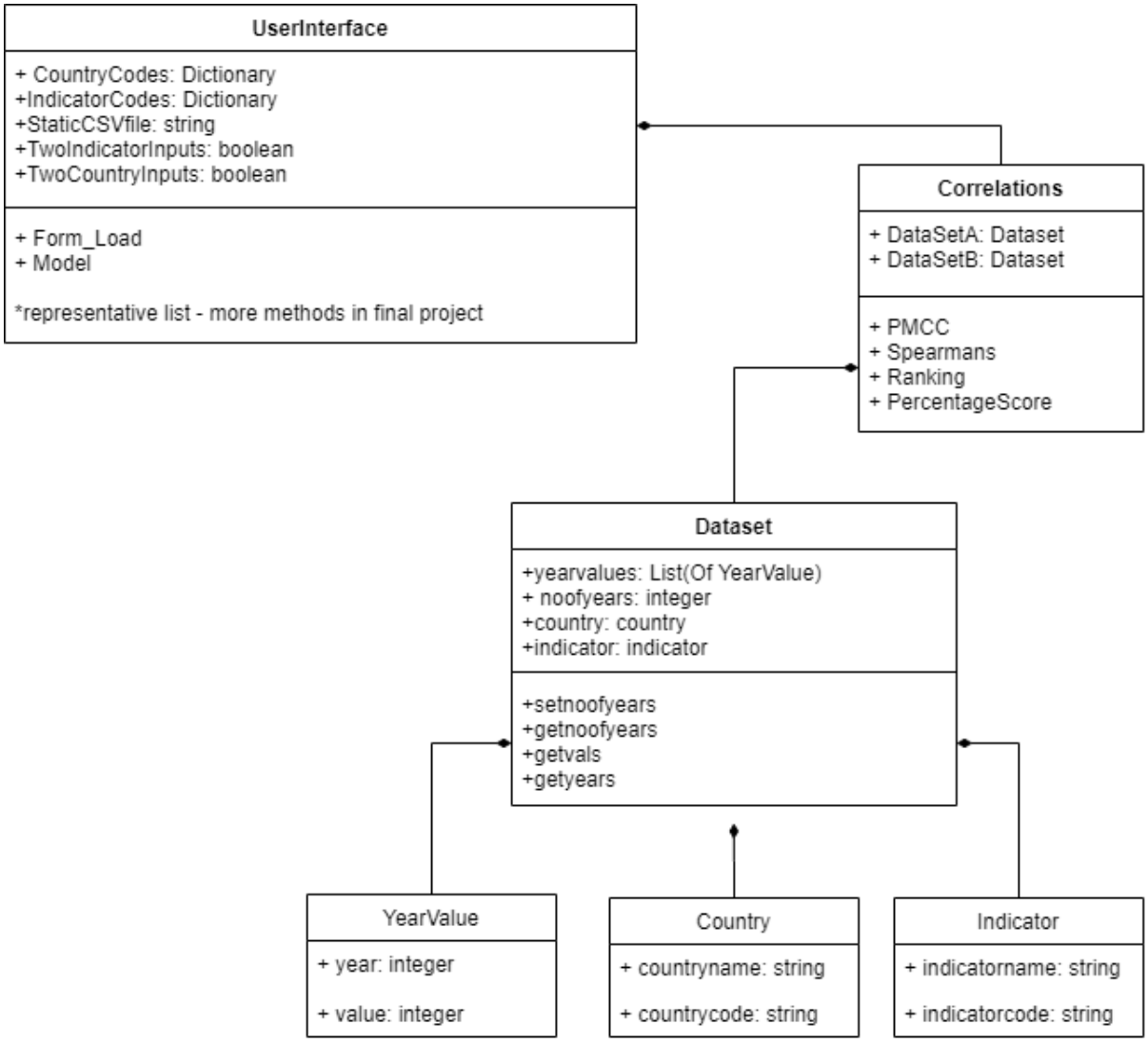


Figure 15- UML Object Diagram

Figure 16- UML Class Diagram

Flowcharts

Flowcharts are a useful tool to analyse complex processes which we identified earlier on in our research would need to take place in our program. Abstracting the problem is critical in order to produce a solution to it. Below is a flowchart documenting the process of creating an API Request.

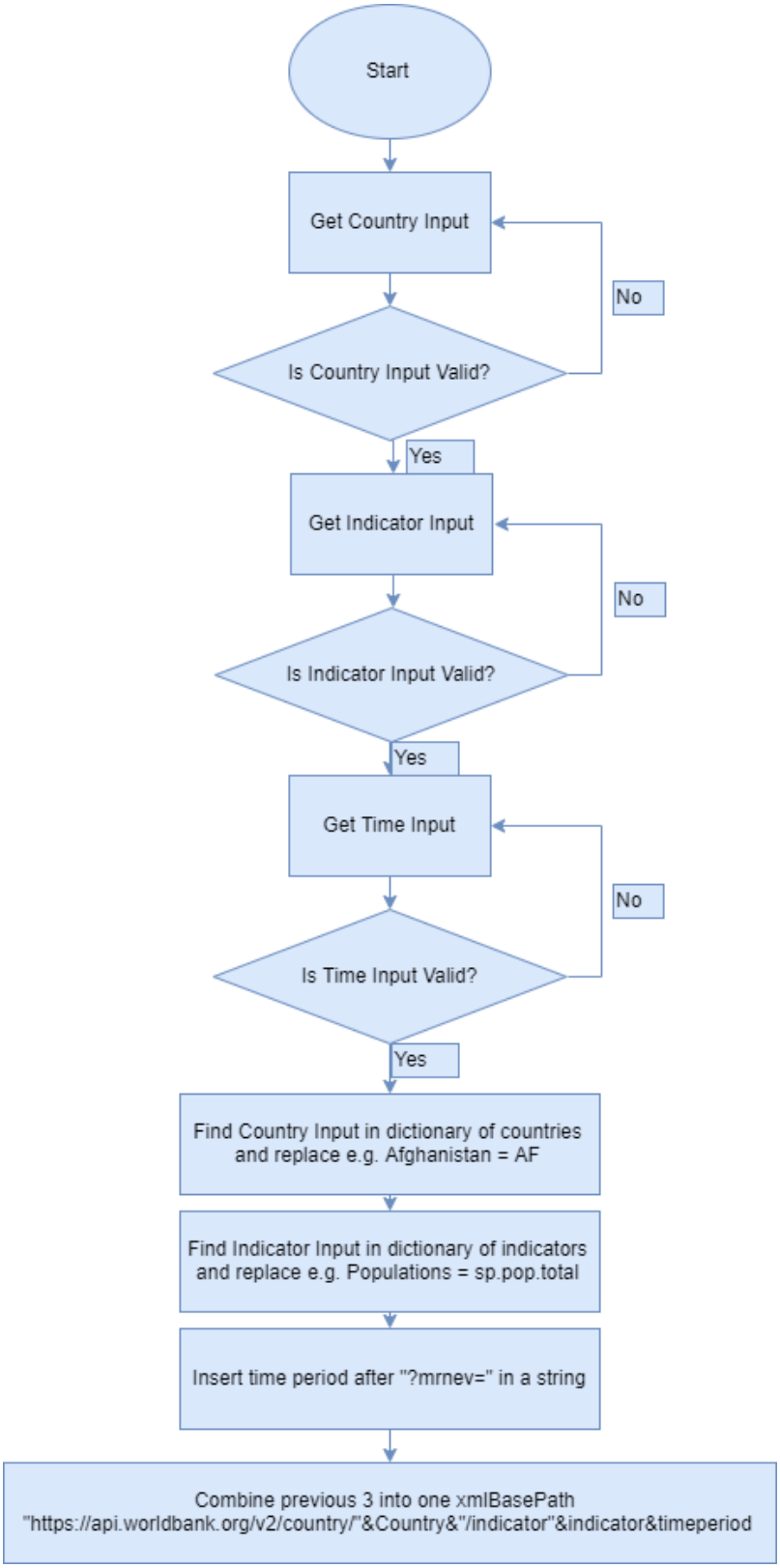


Figure 17- Flowchart showing API request creation

Analysis Data Dictionary

Data will need to be processed in order to retrieve values from a CSV file, and also be used in order to create a request for an API, and this data needs to be in certain forms, and this analysis data dictionary defines critical information about each type of data input.

Attribute Name	Required	Type	Field Length	Default Values	Notes
Country	Yes	String	Unlimited	"Afghanistan"	Default is first alphabetically
Time Period	Yes	Integer	29	10	The field length is variable depending on others, but the standardised max length is 29
Indicator	Yes	String	Unlimited	"Poverty Headcount ($1.90 a day)"	Default is first alphabetically

Dialogue

From my earlier research, the first key problem identified is the process of acquiring the data required, from multiple sources, both the World Bank and the United Nations. In order to obtain this data, there are different options, presented in the data formats shown in the previous chapter, either through CSV files, or API's that both organisations have. Firstly, the most important aspect is the United Nations, as that is the organisation which collects data on the Human Development Index, on which this investigation is about. This should be quick and easy to access as it will be accessed a large number of times. The API for the United Nations I found to be poorly documented with little support and is unnecessarily complex for this project as it contains numerous unneeded data formats. This would mean that the program would lose efficiency and be needlessly complex accessing it, therefore the United Nations data will be stored locally in a CSV file, as that is most appropriate for the data, and can be accessed quickly, and is constantly updated and easily available through following a link, therefore can be downloaded the first time the user runs the program.

For economic indicators to compare it against, there must be a large number of alternatives for comparison to do an accurate investigation, and it would be more difficult to download, organise and access hundreds of CSV files containing data from the World Bank. It would be incredibly inefficient and time consuming, therefore I have come to the conclusion that using the API request system for the World Bank is preferable, as it allows easy access to hundreds of nations data on a plethora of economic indicators, over a large time period.

The other key issues raised by the research is what methods should be used to calculate correlation between two variable factors. The conclusion from this was that no singular method is appropriate for all scenarios, therefore in order for the investigation to be successful, as many of the methods of

data analysis should be applied as possible. Primarily, scatter graphs should be the most visible and easily accessible method as, although it was not perfect, it was the best out of the three methods, due to its graphical presentation being easier to understand, and also any errors being very apparent to the user. The reason it is not the sole method however is due to the degree of subjectivity being too high, and therefore leaving the user with perhaps insufficient evidence to draw accurate conclusions.

Other more obscure methods of measuring correlation were found in research, but they were not generally applicable in most circumstances and tended to be heavily specialised, therefore not a suitable process for this program.

The mathematical techniques of Pearson's Coefficient (PMCC) and Spearman's rank are complicated formula's, however, can be recreated programmatically so therefore are a good choice to use and aid the investigation.

PMCC was chosen primarily because of its universality, being applicable to nearly every linear correlation scenario, and also, as it is more complicated and only taught at A-Level maths it is more likely to produce accurate and in-depth results than other more basic techniques, which are simple and therefore limited in accuracy.

Spearman's Rank may also be difficult as it requires ranking the data which you have, based on how large the integer is in relation to others in the list. In order to do this, both arrays of data must be sorted into either ascending or descending size order. The two methods which could be used for this are a bubble sort and then use positioning for ranking. The other method which I will be using is a ranking algorithm which iterates through each item and does a comparison with each item, which is more efficient for smaller lists, whereas sorting is more appropriate for larger datasets.

Although there is some degree of error with all three methods, together any anomalous results should be spotted, and they should give a comprehensive overview, allowing the user to identify correlation. I will also produce a percentage score combining both mathematical results to make it easy for a new user to understand what the numbers mean in non-mathematical terms.

Finally the program needs to have the ability to handle incomplete datasets, so it will have to be able to account for missing values, using linear interpolation or extrapolation techniques. From research the most effective way of finding these values would be through a regression line, which is also beneficial as it can recycle earlier code used, as Pearson's Coefficient is used to calculate the gradient of the line. This is another statistical technique used in a level mathematics so should add accuracy and validity to the results of my program, as well as making it more robust. However we have to ensure accuracy therefore if there are too many missing values it will just return an error message to the user, as you can't extrapolate large amounts of data as it may become unreliable.

This part of the program is pivotal as it is used to preserve the integrity of the data which is being used.

Programming Language

In order to produce my solution, I need to determine how it should be written, for maximum efficiency and also maximum utility. Due to the large amount of data handling which will occur, a declarative programming paradigm may make the project easier to code, due to its nature of describing what you want to do, not how you want to do it. Declarative programming is more efficient, and a language such as R has many useful built in functions that can be used to create

graphs and handle data. However, theses languages tend to be slightly limited, as you are more restricted with what you can create, as the simple nature means complex problems are harder to solve.

Therefore, a third generation, high-level programming language would be more appropriate for my solution. This is because it can be used to solve more complex solutions than a fourth-generation language, due to its use of the imperative programming paradigm. This means that it can be used to code how to solve specific issues, giving it a greater scope. It also used English like statements making it easy to understand and efficient to code, with a large number of built in functions, making it far better than assembly language or machine code for this solution.

Finally, the decision of which specific language to use was a very simple one. Visual Basic is a language which I have already learnt and am proficient in, and it will also be used for my exams at the end of this year, so improving my skills in it by using it for this solution would be very beneficial. Visual Basic is also a good choice as it is developed by Microsoft so has extensive support and good documentation to help overcome any issues which may arise. It can also make use of Windows Forms, an object-oriented way of creating a graphical user interface, which can be useful in aiding human computer interaction, creating a better, easier experience for the user of the software.

List of Requirements

0 – The system must allow a user to enter inputs.

- *0.1 The system must present the user with a list of valid inputs.*
  - *0.1.1 The system must present a list of regions available to choose from with the ability to check a box next to the one the user wants to select.*
  - *0.1.2 The system must have a textbox for the user to write an integer.*
  - *0.1.3 The system must present a list of economic indicators available to choose from with the ability to check a box next to the one the user wants to select.*
- *0.2 The user must only input valid data, not erroneous data.*
  - *0.2.1 Only one time period must be selected, with a maximum value of 29.*
- *0.3 The system must allow for different combinations of country & indicator inputs.*
  - *0.3.1 The system must allow 1 country, 1 indicator and HDI.*
  - *0.3.2 The system must allow 2 countries and 1 indicator.*
  - *0.3.3 The system must allow 1 country and 2 indicators.*

1 – The system must retrieve data.

- *1.1 The system must retrieve the human development index from the local CSV files.*
  - *1.1.1 The system must open the file.*
  - *1.1.2 The system must read the file.*
  - *1.1.3 The system must do a recursive binary search to find the country listed in alphabetical order.*
  - *1.1.4 The system must store the required data in a local variable.*
  - *1.1.5 The file must be closed by the system.*
- *1.2 The system must retrieve the other data from the World Bank API.*
  - *1.2.1 the system must create an API request.*
  - *1.2.2 The system must send the request.*
  - *1.2.3 The system must store the result of the request in a local variable.*
  - *1.2.4 The system must process the XML file.*
    - *1.2.4.1 The system must search the XML file for the required tags*
    - *1.2.4.2 The data under the required tags must be stored in a local array variable.*
- *1.3 Where data is incomplete this must be handled.*
  - *1.3.1 If data is too incomplete an error message should be returned.*
  - *1.3.2 If data is nearly complete missing values must be interpolated.*
    - *1.3.2.1 A regression line between year and the indicator must be created.*
    - *1.3.2.2 Missing values must be substituted into the regression line to provide a useable, accurate value.*

2 – The system must analyse the data.

2.1 The system must produce a range of graphs from local variables created in 1.2.4.2 and 1.1.3.

2.1.1 There must be 3 graphs generated with content varying depending on number of each input in 0.3

2.1.2 Two-line graphs must be generated

2.1.3 One scatter graph must be generated

2.2 The system must calculate the PMCC of the variables created in 1.2.4.2 and 1.1.3.

2.2.1 The sum of all x values must be found.

2.2.2 The sum of all y values must be found.

2.2.3 The sum of all x^2 values must be found.

2.2.4 The sum of all y^2 values must be found.

2.2.5 The sum of all x*y values must be found.

2.2.6 The components should be put into the formula for PMCC.

2.3 The system must calculate the Spearman's Rank of the variables created in 1.2.4.2 and 1.1.3.

2.3.1 The arrays must be ranked, from smallest to largest.

2.3.2 The PMCC of the ranked arrays must be found

2.4 The system must produce a single output from the results of 2.3 and 2.2.

3 – The system must output the analyses results in an understandable way to the user.

3.1 The system must explain what is currently being shown to the user.

3.2 The system must show the line graph generated in 2.1 to the user.

3.3 The system must output the result of the PMCC calculation (2.2).

3.4 The system must output the result of the Spearman's Rank calculation (2.3).

3.5 The system must save the results to a file to be accessed at a subsequent time by the user.

3.6 The system must output the result of 2.4

Design

Packages and Framework

API Interaction

The Indicators API supports two basic ways to build queries: a URL based structure and an argument-based structure. For example, the following two requests will return the same data, a list of countries with income level classified as low income:

Argument based: http://api.worldbank.org/V2/country?incomeLevel=LIC

URL based: http://api.worldbank.org/V2/incomeLevel/LIC/country

The response from the interaction will be in the xml file format as this format is the default setting used, and to use JSON format just creates unnecessary extra work, as visual basic has an inbuilt xml reader library.

In order to create an argument-based API Query, we will need to create a base string, and fill in user specified parameters to it, in order to retrieve the relevant data. The base URL will be:

```
https://api.worldbank.org/v2/country/XX/indicator/XXXXXXXX?mrv=XX&format=xml
```

The X's represent characters which will need to be replaced by user defined parameters. Firstly, the country code will be replaced, the second blank is the indicator code and the third one is an integer for how far back you want to retrieve data.

The second API interaction that I will be utilising is a flags API used to retrieve the flag of a specific country, based on its two-letter country code, the same codes used in the Indicators API. A similar URL based request will occur, in the format:

```
https://www.countryflags.io/XX
```

The response will need to be downloaded and assigned to a picture box element of the user interface, meaning an object will need to be created to display the picture.

Data Visualisation Package Library

The system that I will use to produce graphs is the inbuilt chart object in visual studio .net framework. To utilise this class, the property Series, along with the object series. A series object will need x and y coordinates added to it, before that series is then used as a parameter for the chart

object. The chart object will need to be of type line graph, as otherwise, it will create the wrong, and an inaccurate data visualisation.

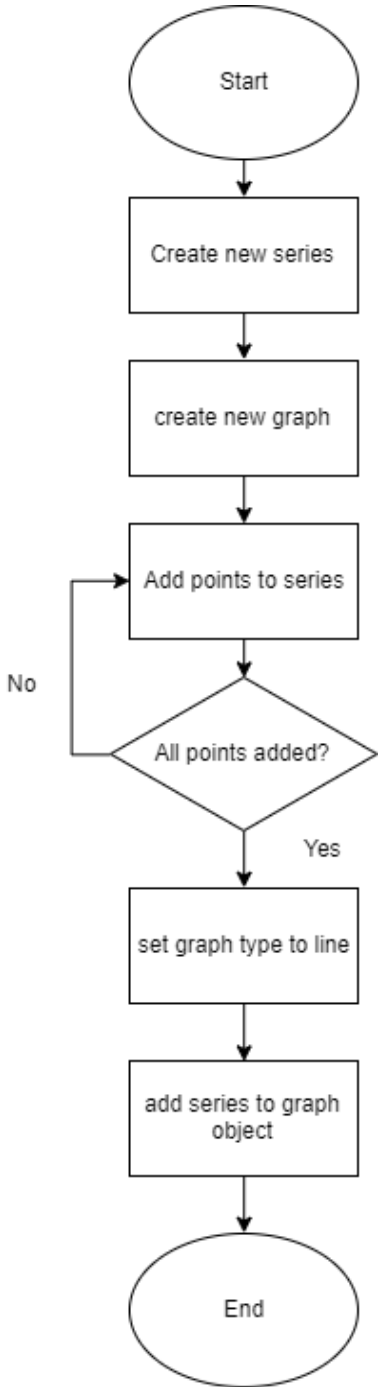


Figure 18- Data Visualisation Flowchart

Data Structures and Files

Dictionaries

Below is a list of the key data structures which will allow my program to retain data and enact its processes:

CountryCodes As New Dictionary(Of String, String)

IndicatorCodes As New Dictionary(Of String, String)

Dim DataSetA As New DataSet

Dim DataSetB As New DataSet

DataSet will be a new class with the following properties:

Public Property country As New Country

Public Property indicator As New Indicator

Public Property yearValues As New List(Of YearValue)

Private Property NoOfYears As Integer

These dictionaries are important to the program as they are used to convert the user inputs into a recognisable form for the word bank API. The use of objects in lists, such as the list of yearvalues will be important as it will be used to store the retrieved data values from the API and the CSV file.

CSV File

A1 COUNTRY

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	COUNTRY	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001
2	Afghanist	0	0	0	0	0	0	0	0	0	0	0	0
3	Albania	0.645	0.626	0.61	0.613	0.619	0.632	0.641	0.641	0.652	0.662	0.669	0.676
4	Algeria	0.577	0.581	0.587	0.591	0.595	0.6	0.608	0.617	0.627	0.636	0.644	0.653
5	Andorra	0	0	0	0	0	0	0	0	0	0	0.759	0.767
6	Angola	0	0	0	0	0	0	0	0	0	0.374	0.387	0.401
7	Antigua ar	0	0	0	0	0	0	0	0	0	0	0	0
8	Argentina	0.704	0.713	0.72	0.725	0.728	0.731	0.738	0.746	0.753	0.764	0.771	0.776
9	Armenia	0.631	0.628	0.58	0.588	0.599	0.605	0.612	0.623	0.636	0.642	0.647	0.652
10	Australia	0.866	0.867	0.868	0.872	0.875	0.883	0.886	0.889	0.892	0.895	0.898	0.9
11	Austria	0.795	0.8	0.805	0.807	0.813	0.817	0.82	0.824	0.835	0.834	0.838	0.849
12	Azerbaijar	0	0	0	0	0	0.612	0.612	0.617	0.626	0.633	0.64	0.648
13	Bahamas	0	0	0	0	0	0	0	0	0	0	0.776	0.779
14	Bahrain	0.746	0.752	0.757	0.765	0.769	0.775	0.778	0.779	0.783	0.786	0.792	0.792
15	Banglades	0.387	0.394	0.402	0.409	0.417	0.425	0.433	0.442	0.451	0.46	0.468	0.477
16	Barbados	0.716	0.718	0.718	0.721	0.727	0.731	0.735	0.74	0.736	0.743	0.752	0.751
17	Belarus	0	0	0	0	0	0.657	0.661	0.667	0.671	0.676	0.683	0.689
18	Belgium	0.806	0.81	0.825	0.839	0.845	0.852	0.857	0.862	0.866	0.868	0.873	0.876
19	Belize	0.644	0.651	0.657	0.661	0.661	0.662	0.662	0.664	0.666	0.67	0.677	0.679
20	Benin	0.348	0.354	0.358	0.365	0.368	0.373	0.377	0.381	0.385	0.391	0.398	0.41
21	Bhutan	0	0	0	0	0	0	0	0	0	0	0	0
22	Bolivia (Pl	0.536	0.543	0.55	0.557	0.564	0.571	0.578	0.58	0.591	0.6	0.608	0.611
23	Bosnia an	0	0	0	0	0	0	0	0	0	0	0.672	0.677

HDI

Figure 19- CSV Input

This CSV file is another input for the program, shown in the diagram above. This file will need to be accessed by the inbuilt reader in visual basic.

Output – Text File

The program will output to a text file, enabling the user to retrieve previous calculations all in one place and make multiple larger comparisons over time. The data will need to be stored in a clear, understandable way so that the user can easily read it. The structure below is how the data will be stored.

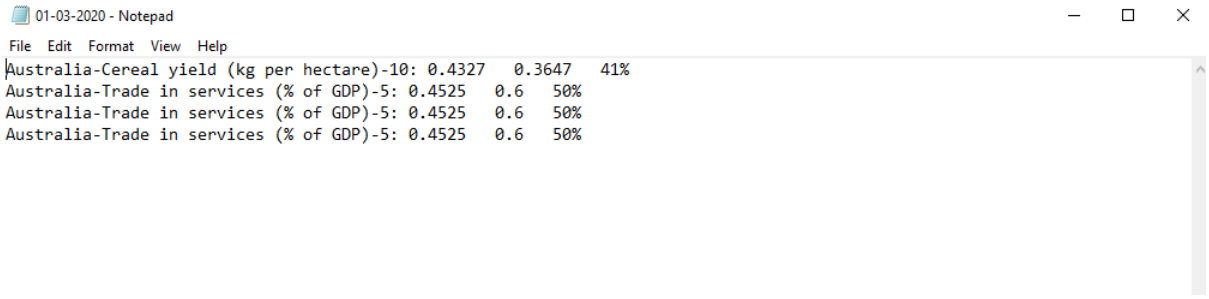


Figure 20 - Text File Storage

The results are sorted by date, shown by the name of the file. It then shows the parameters that were inputted, clearly followed by the calculation results, in the same order that they are given in the user interface, PMCC then spearman's rank followed by the combined score.

Top Level Design

See analysis page 19 for a UML Object Diagram.

See next page for how the user will interact with the program.

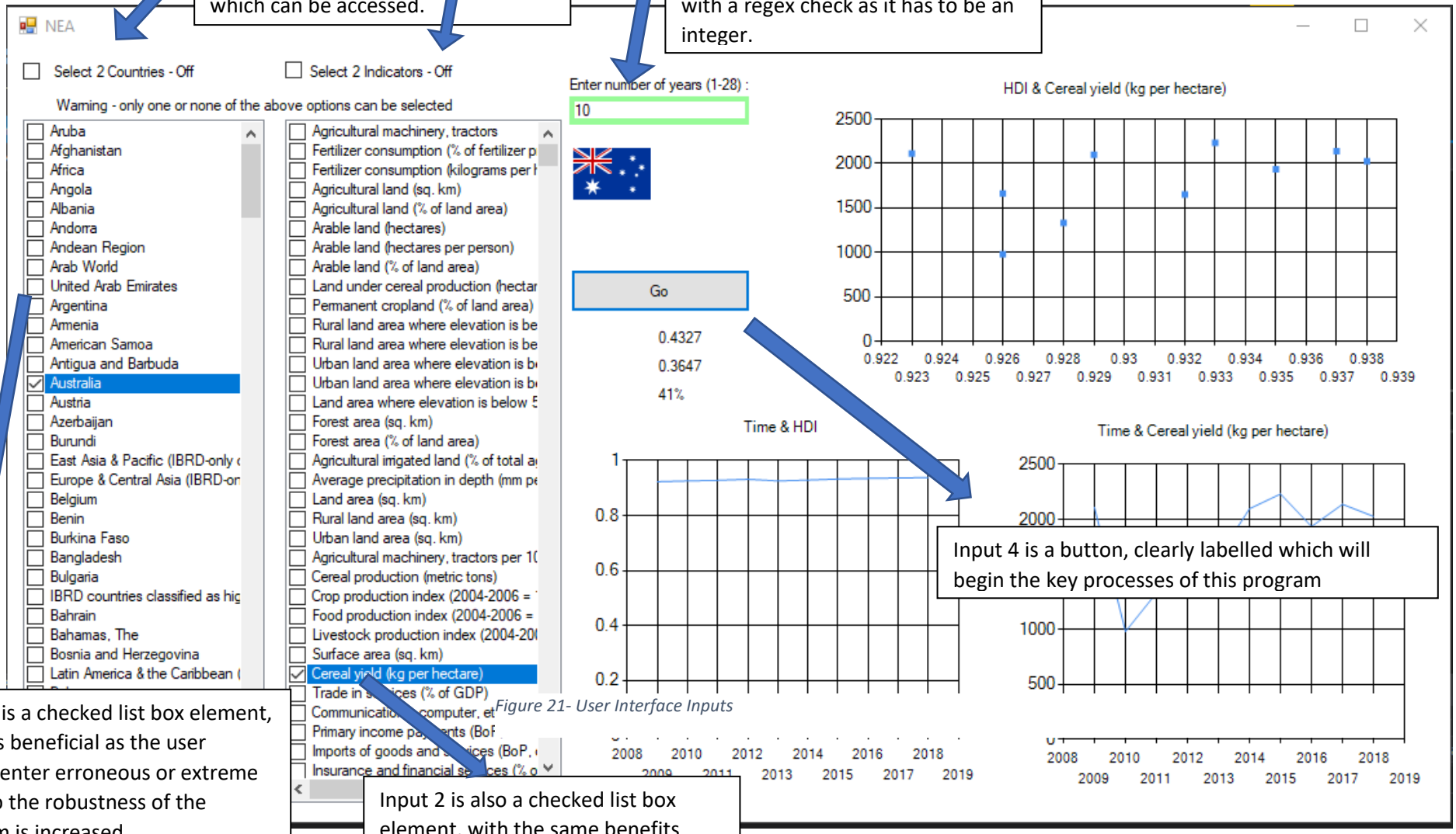
Validation

There is relatively little validation required, as shown in the diagrams below most inputs are set. Either in the checked list box or a button form. I will us RegEx to validate the text box input, as only certain characters can be entered. The RegEx pattern used to match against will be - ^[1-9][0-9]*$. This only allows two digit or single digit numbers to be entered into the program, and will subsequently be followed by a decision statement to ensure that the integer is between 1 and 28.

User Interface

Input 5+6 are checked boxes to change the amount of other inputs which can be accessed.

Input 3 is a text box input, so therefore will need to be validated with a regex check as it has to be an integer.



Input 4 is a button, clearly labelled which will begin the key processes of this program

Input 1 is a checked list box element, which is beneficial as the user cannot enter erroneous or extreme data, so the robustness of the program is increased

Input 2 is also a checked list box element, with the same benefits

Figure 21- User Interface Inputs

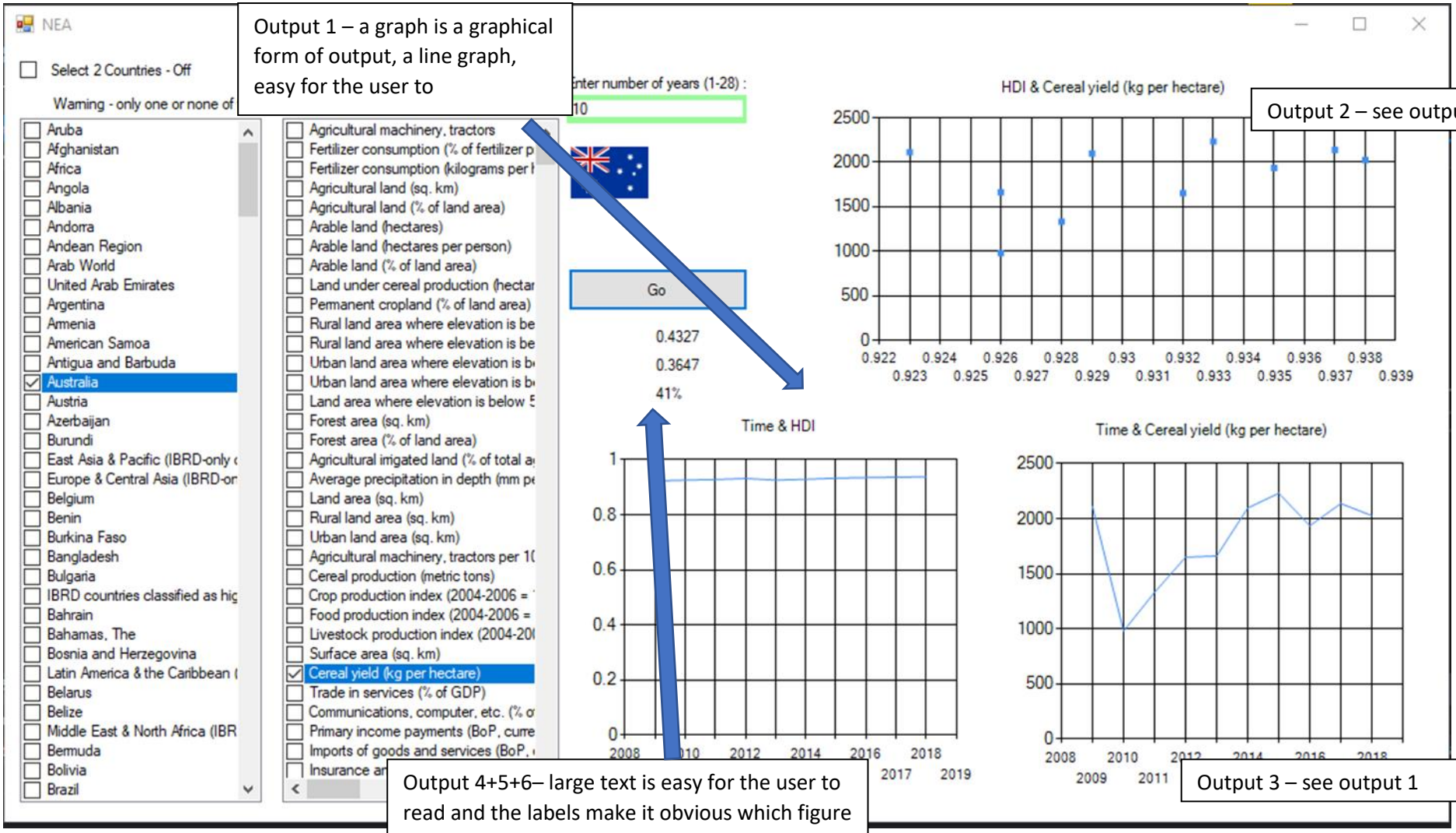


Figure 22- User Interface Outputs

Processes

Pseudocode –

Generating Inputs

This is pseudocode for the process of creating one of the checked list box inputs, which will be a pivotal part in the user interface.

```
Global CountryCodes() as Dictioary

PageNumber = 0

Do

    PageNumber += 1

    Try

      reader = XmlTextReader("https://api.worldbank.org/v2/country?page=" &
PageNumber.ToString)

      While reader.Read()

        If XmlNodeType.Element = reader.NodeType Then

          ElementName = reader.Name

        End If

        If (XmlNodeType.Text = reader.NodeType) And ElementName = "wb:iso2Code" Then

          CountryCode = reader.Value

        ElseIf XmlNodeType.Text = reader.NodeType And ElementName = "wb:name" Then

          CountryName = reader.Value

          CheckedListBox1.Items.Add(CountryName)

          CountryCodes.Add(CountryName, CountryCode)

        End If

      End While

    Catch ex As Exception

      Output("Error")

    End Try

  Loop Until PageNumber = 7
```

API Query

```
Global Years()

Global Values()
```

```
Do
        PageNumber += 1
        Try
            reader = XmlTextReader(xmlFileLocation)
            While reader.Read()
                If XmlNodeType.Element = reader.NodeType Then
                    ElementName = reader.Name
                End If
                If (XmlNodeType.Text = reader.NodeType And ElementName = "wb:date") Then
                    Year = reader.Value
                ElseIf XmlNodeType.Text = reader.NodeType And ElementName = "wb:value" Then
                    Value = reader.Value
                    Years.Add(Year)
                    Values.Add(Value)
                End If
            End While
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    Loop Until PageNumber = TotalPages
    Years.Reverse() ‘ The request returns most recent years first, but to graph it we need it in the reverse order
    Values.Reverse()
    For i = 0 To (Years.Count() - 1)
        s.Points.AddXY(Years.Item(i).ToString(), Values.Item(i))
    Next
```

PMCC

```
For i = 0 To Array1.Length - 1
    SumX += Array1(i)
    SumY += Array2(i)
    SumXY += Array1(i) * Array2(i)
    SumXSquared += Array1(i) ^ 2
```

```
    SumYsquared += Array2(i) ^ 2
Next
Output (Array1.Length * SumXY - SumX * SumY) / (Math.Sqrt((Array1.Length * SumXSquared - SumX ^ 2) * (Array1.Length * SumYsquared - SumY ^ 2)))
```

Ranking algorithm for Spearman's

This ranking algorithm will be used to weight the two arrays in terms of largest to smallest, and when that is completed, the earlier algorithm for Pearson's coefficient can be applied to fin spearman's rank.

```
Dim array1(Array.Length - 1) As Decimal
    For i = 0 To Array.Length - 1 Step 1
        Dim r As Integer = 1 ' rank increases if it's bigger than a value
        Dim s As Integer = 1 ' same number counter
        For j = 0 To i - 1 Step 1
            If Array(j) < Array(i) Then
                r = r + 1
            ElseIf Array(j) = Array(i) Then
                s = s + 1
            End If
        Next
        For j = i + 1 To Array.Length - 1 Step 1
            If Array(j) < Array(i) Then
                r = r + 1
            ElseIf Array(j) = Array(i) Then
                s = s + 1
            End If
        Next
        array1(i) = r + (s - 1) * 0.5
    Next
    Return array1
```

Binary search for strings

This binary search algorithm will be used when fetching data from the csv file, where countries are stored in alphabetical order, and the recursive nature of this algorithm allows it to search the ordered list.

```
Function BinarySearch(ByVal array As List(Of String), ByVal max As Integer, ByVal min As Integer,
ByVal target As String, ByVal DigitNumber As Integer)
    If min > max Then
        Return "-1"
    End If
    Dim mid As Integer = (max + min) / 2
    Dim currentItem As String = array(mid)
    If currentItem.ToLower.ToString = target.ToLower.ToString Then
        Return mid.ToString()
    Else
        Dim temp As Integer = checkLetters(array, max, min, target, mid, DigitNumber)
        If temp = 1 Then
            Return BinarySearch(array, mid - 1, min, target, 0)
        ElseIf temp = 2 Then
            Return BinarySearch(array, max, mid + 1, target, 0)
        Else
            If DigitNumber = target.Length - 1 Then
                Return "-1"
            Else
                Return BinarySearch(array, max, min, target, DigitNumber + 1)
            End If
        End If
    End If
    Return mid
End Function
```

Testing

Input – requirement no. 0's

https://youtu.be/qBM4SJH5mhg

Test No. / Requirement No.	Description of Test	Expected Output	Pass/Fail - Timestamp

1 0.1.1	Click Checked List Box input which contains countries by user.	The item which is clicked has a tick appear next to it, and the previous tick disappears.	0:15
2 0.1.3	Click Checked List Box which contains indicators by user.	The item which is clicked has a tick appear next to it, and the previous tick disappears.	0:26
3 0.1.2	Enter typical data for Number of years input textbox by user.	The number of years is entered into the processing of the program	0:38
4 0.1.2 0.2.1	Enter erroneous data for number of years input textbox by user.	An error message is shown, and the processing section of the program cannot begin running until a correct value is entered	0:50
5 0.1.2	Enter extreme data for number of years input textbox by user.	The number of years is entered into the processing of the program.	1:07
6	API Input - hard list of indicators retrieved from hard coded URL	The XML file is retrieved from the request address, in order to be processed.	1:26
7	API input – hard list of country codes and countries is retrieved from a hard-coded URL.	The XML file is retrieved from the request address, in order to be processed.	1:38
8	Go button is clicked by the user.	The button click registers and the program begins processes.	1:50
9 0.3.2	Enter typical data when the 2 countries button is selected	The inputted data is correctly assigned by the program.	2:09
10 0.3.2	Enter erroneous data when the 2 countries button is selected	An error message is shown to the user	2:26
11 0.3.3	Enter typical data when the 2-indicator button is selected	The inputted data is correctly assigned by the program.	2:38
12 0.3.3	Enter erroneous data when the 2-inidcator button is selected	An error message is shown to the user	2:58

13 0.3.1	Enter typical data when neither extra button is selected.	The inputted data is correctly assigned by the program.	3:09
14 0.3.1	Enter erroneous data when neither extra button is selected	An error message is shown to the user	3:24

Processes

https://youtu.be/bTNXRzpFe4k

Test No. / Requirement No.	Description of Test	Expected Output	Pass/Fail - Timestamp
1	Access API	The xml file is retrieved from the hard-coded URL containing country data.	0:14
2	Create input checked list box countries	The program iterates through the country's column in the CSV file and adds it to the checked list box containing countries.	1:20
3	Create country to country code dictionary	As the program iterates through each country in the XML file, the country name and two-digit reference code are added to a dictionary.	0:14
4	Access API	The xml file is retrieved from the hard-coded URL containing indicators data.	0:44
5	Create Input Checked List box containing indicators	The program iterates through each indicator in the XML file, adding the indicator name to a checked list box.	1:20
6	Create indicator to indicator code dictionary	As the program iterates through each indicator in the XML file, the Indicator Name and Code are put into a dictionary.	0:44
7 1.2.1	Create API request	The inputs are fed through both dictionaries to get data which can then be combined to create	1:53

		a URL request for the API, which is in the correct format.	
8 1.2.2 1.2.3	API request	The XML file is received from the variable URL	2:18
9 1.2.4.1 1.2.4.2	Create list from request	The XML file is iterated through and the data for each year is added to a list, resulting in a list of values.	2:18
10 1.1.1 1.1.2	Access CSV file	The CSV file data is passed into the program	2:57
11 1.1.3 1.1.5	Binary search for position of country	An integer is returned which is the index of the country in the CSV file.	3:26
12 1.1.4	Create list of values from input	The data from the row selected by the chosen country, and over the specified time period is selected and a list is produced containing HDI values.	3:58
13	Check for completeness in both lists.	If either of the two lists is incomplete, with missing or zero values, then it should be checked if it can be interpolated	4:30
14 1.3	Check for interpolation of mostly complete set – using a bitwise comparison	A message appears saying that values have been interpolated and the interpolation function has been called	4:30
15 1.3.2.1 1.3.2.2	Interpolate missing data	Every item in the list where there previously was a 0 is replaced by a value calculated using a regression line analysis.	6:41
16 2.2	PMCC data	The product moment correlation coefficient of the two datasets is found	5:48

17 2.3.1	Rank data	The two datasets are ranked, in order of largest to smallest	Ranking Dry Run
18 2.3 2.3.2	Spearman's rank data	The ranked data is used to calculate an accurate spearman's rank value	6:06
19	Create unique percentage value	Spearman's rank and PMCC results are combined overall using given weightings to give a singular output for the user.	6:13

Storage

https://youtu.be/gqu6DqYiITY

Test No. / Requirement No.	Description of Test	Expected Output	Pass/Fail - Timestamp
1 3.5	Creation of save file	If the file has not already been created, a text file is created in local storage.	0:09
2 3.5	Data is saved correctly	Selected data is permanently appended to the end of the text file.	0:45

Output

https://youtu.be/zX2S3_W2yKs

Test No. / Requirement No.	Description of Test	Expected Output	Pass/Fail - Timestamp
1 2.1.1 2.1.2 3.2	Graph 1 is produced	A graph with one line, typically showing the chosen indicator in the chosen country over the chosen time period is shown on the user interface	0:17
2 2.1.1 2.1.2 3.2	Graph 2 is produced	A graph with one line, typically showing the HDI of the chosen country over the chosen time period is shown on the user interface	0:26
3 2.1.1 2.1.3	Graph 3 is produced	A scatter graph, typically showing the indicator and the HDI	0:35

3.2		of the chosen country is shown on the user interface	
4 3.3	Label containing PMCC calculations results is shown.	A label containing the results of the Pearson's coefficient calculation is shown on the user interface.	0:48
5 3.4	Label containing spearman's rank calculation results is produced.	A label containing the results of the Spearman's Rank calculation is shown on the user interface	0:55
6	Error message box for incomplete data	A message box appears when wrong or incomplete inputs are entered into the program, telling the user something went wrong.	1:06
7	Message box warning that data has been interpolated	Message box warning that data has been interpolated	1:30
8	Message box saying data is incomplete	Message box saying data is incomplete	2:52
9	Country flag shown when it is selected	An image of the country flag shown when it is selected	1:58
10 3.1	It is explained what is currently shown to the user.	All user interface items are labelled clearly and correctly	2:12
11 3.6	Label containing the percentage correlation score is displayed	Label containing the percentage correlation score is visible on the user interface	2:30

Dry Runs

PMCC & Spearman's rank. Any discrepancies are due to rounding to 2.d.p. in the dry run.

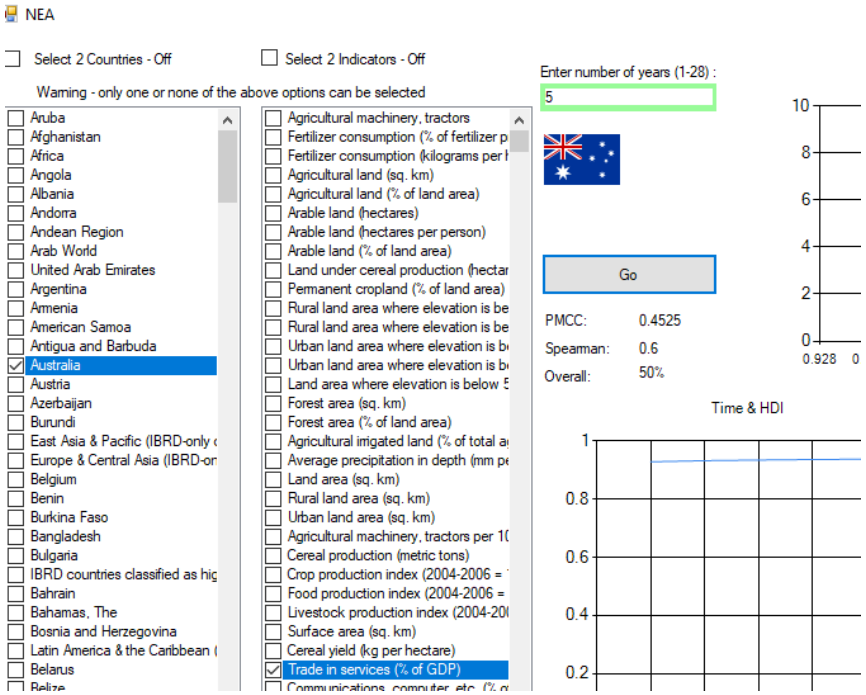
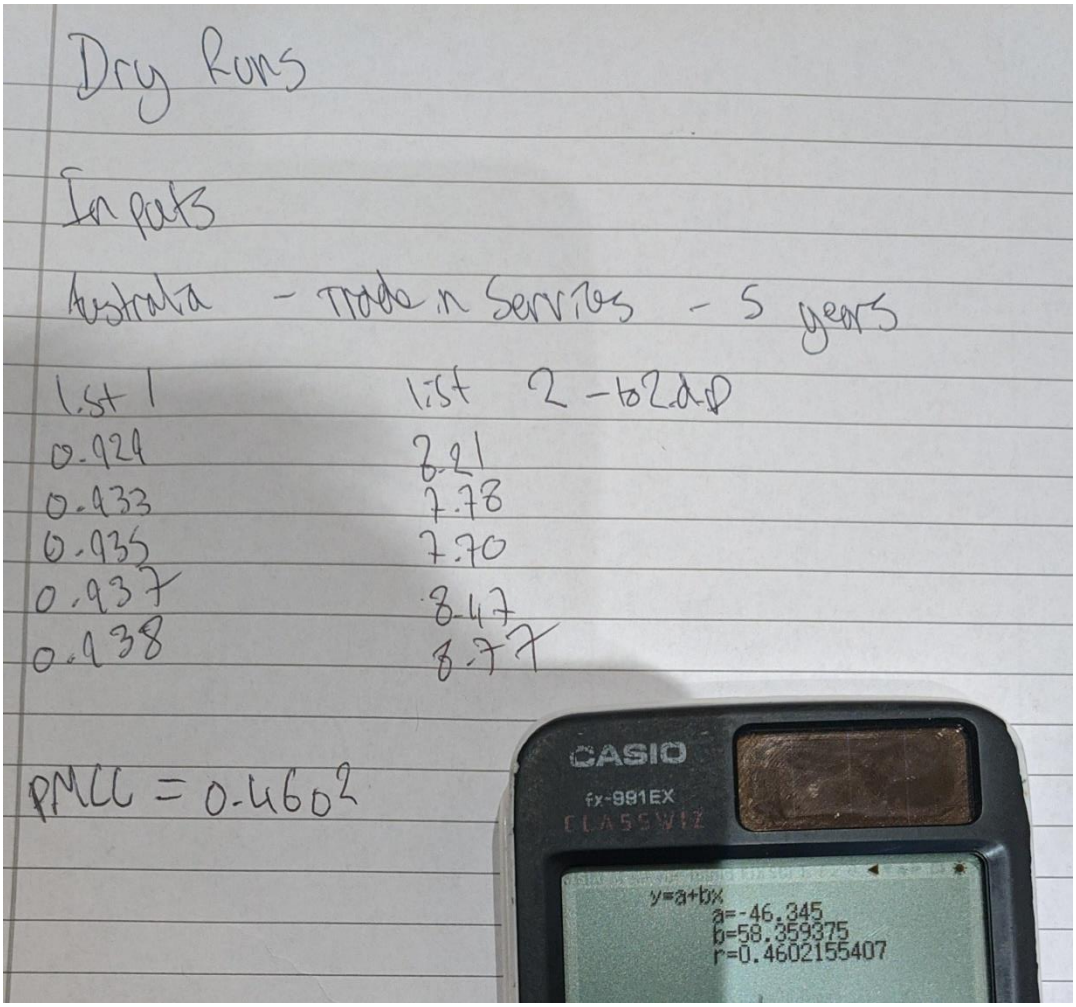
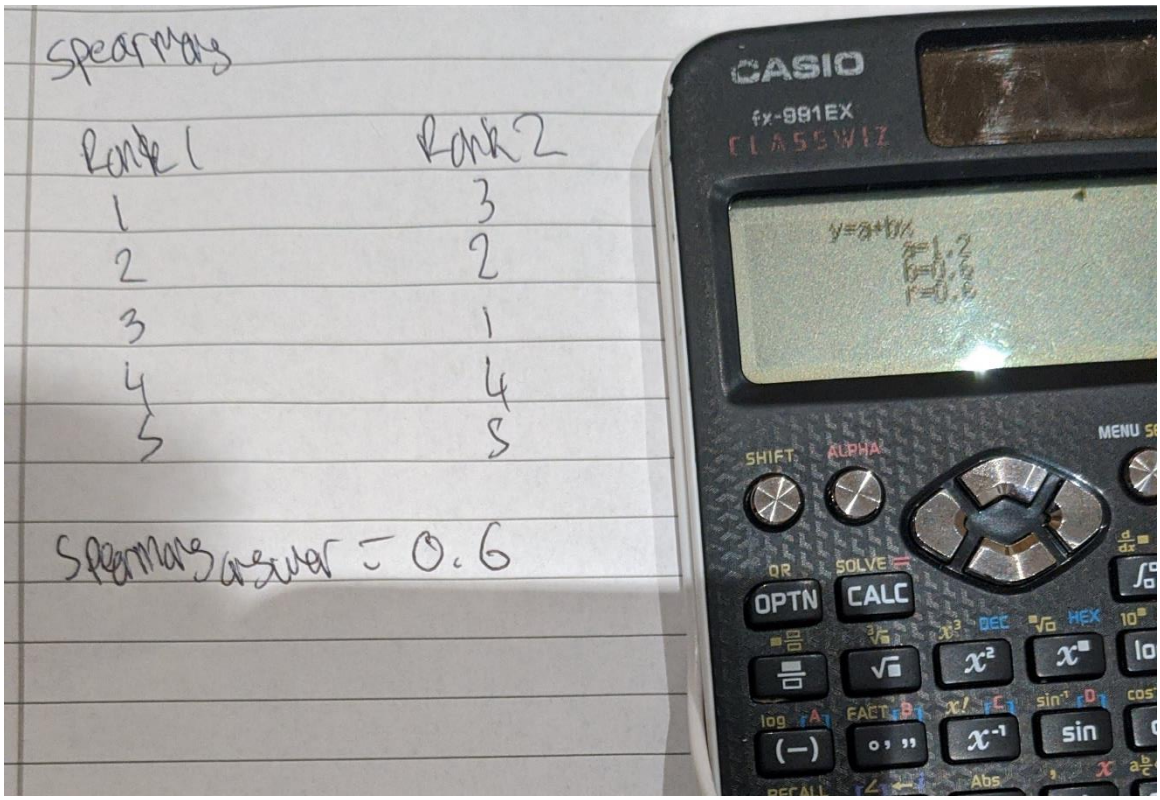
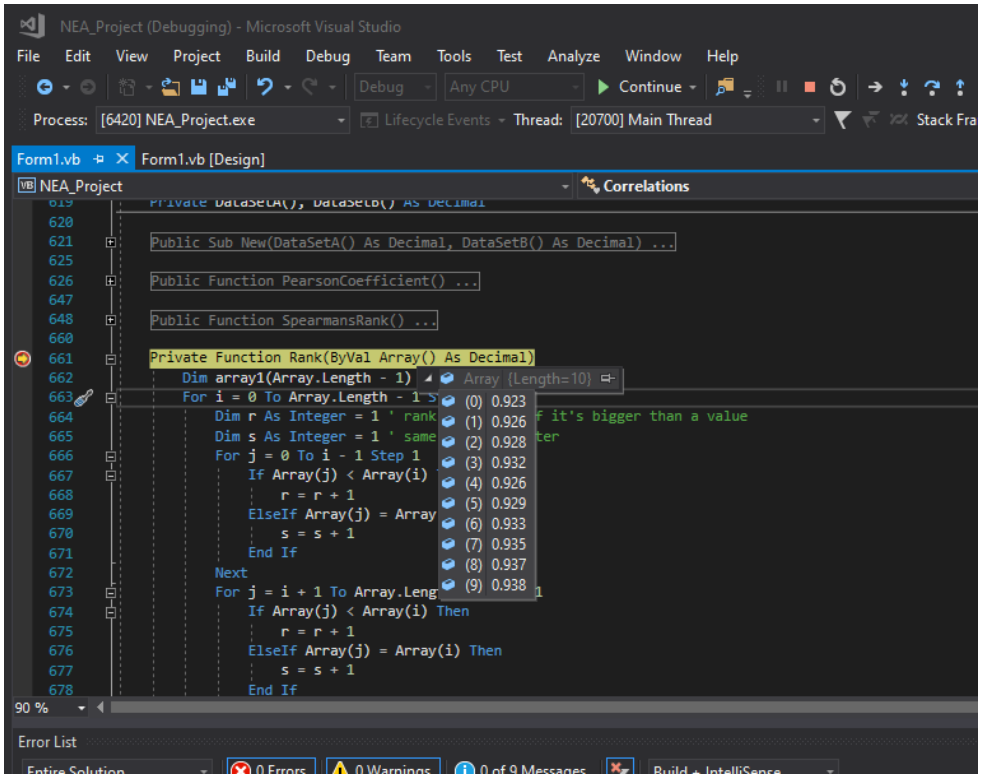


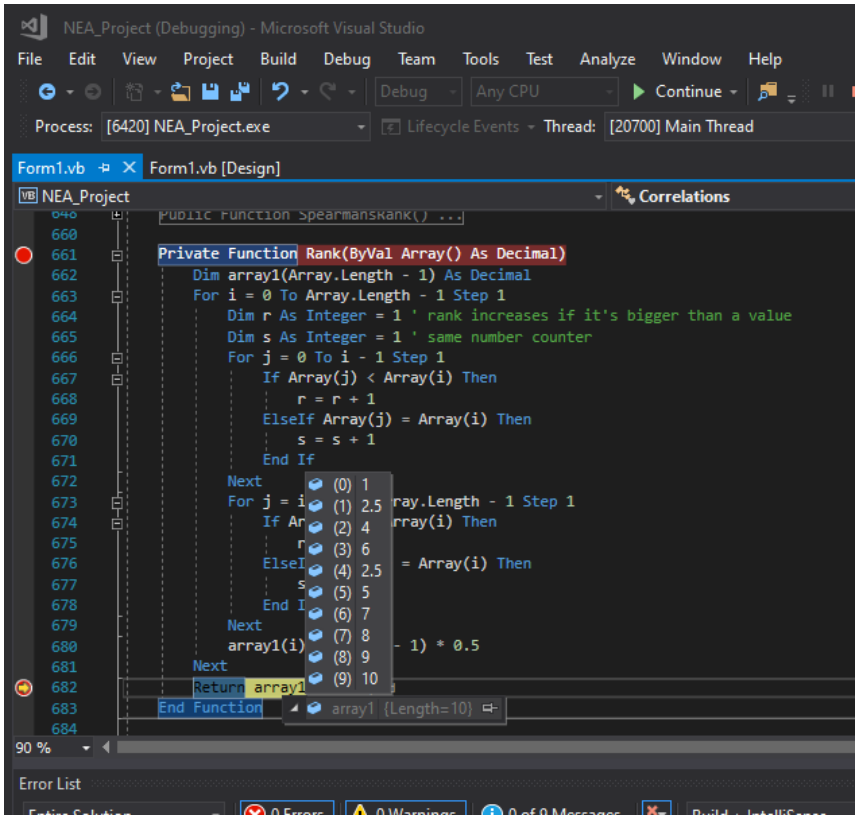
Figure 23- PMCC Dry Run





Ranking algorithm dry run





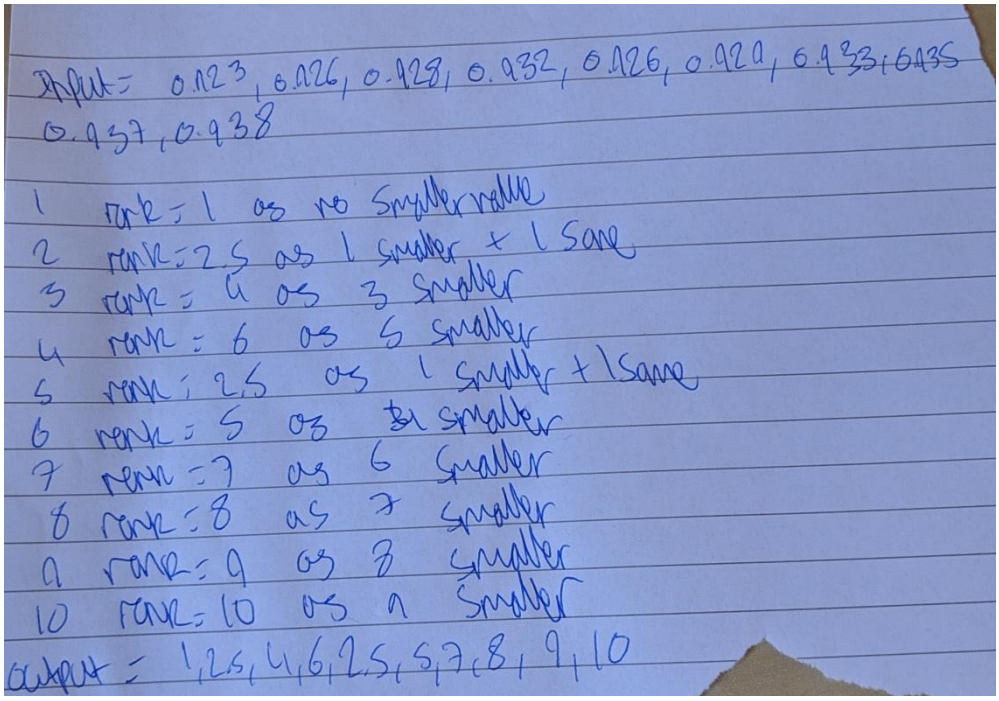


Figure 24- Ranking Dry Run

Evaluation

Improvements

If I was starting a new project, there is one key thing which I would improve about the project. I would alter it so that only inputs that would yield complete data are able to be selected, as currently there are numerous times incomplete datasets are chosen and the program just has to return an error message. To do this the program would have to go through and retrieve every possible input combination given current input restrictions, and the reason I have not implemented it in this project is hardware limitations. The program already takes a reasonable length of time just setting up the user interface as API access is slow on slower wi-fi networks and the laptop that this project was produced on does not have much processing power, so I couldn't implement this change. But going forward it is definitely something that I will look to implement and try to research ways around slow data download speeds over Wi-Fi. One way around this could be to request data in parallel using a programming language such as python.

Another improvement that could be made is the ability to resize graphs by the user in order to create a user interface that is more suited to each individual user. A change like that would allow the user to scale different parts of the user interface to tailor it to their own specific needs, which would improve the overall user experience. It would allow them to make specific graphs easier to view and more prevalent in the display. However this change would be purely aesthetic and is beyond the capabilities of visual basic, meaning I would have to redo the entire program in another language which is why I did not manage to implement this in my project.

One final improvement which could be made to the project were I to do it again is the time delay. When there is no internet connection, the program takes a long time, trying to connect to the API, and keeps trying for a long period of time before returning an error message to the user. During this time the program is frozen and no indication of what is happening is given to the user. It may be useful to implement a timeout function so If it takes longer than say ten seconds, the error is returned earlier. Additionally some kind of feedback during the process would be beneficial, an example of which could be a status message or a loading bar, to improve the user experience, and make it more ergonomic to the average user.

Feedback

Feedback

REPLY REPLY ALL FORWARD

Courtney Anderson
Wed 18/03/2020 16:31

mark as unread

To: Adam Curry

Action Items

Hi Adam

I've had a chance to look through your program and have been impressed by it, here are some responses to the questions you sent:

What are your thoughts on using the program and it's interface?

1. The program is easy to read and easy to navigate
2. Titles are clear and understandable which is good.
3. Large range of inputs is particularly useful as it makes the program versatile for many learning situations.

What are your thoughts on the processes which the program completes?

4. Good graphs, they are especially useful due to the pivotal nature of graphs in the A-level syllabus.
5. Useful, makes students understand causation and correlation easier, and is useful to graphically illustrate economic prosperity
6. Useful for the development topic which is a large macro topic in the a-level

Are there any improvements which could be made?

More colours in the graphs would be good to clearer highlight differences between graphs and trends in the data, so whoever is using it has to do less spotting of patterns.

Tres cordialement

Courtney

Figure 25- Feedback Email

Requirements

Requirement	Met	Test Location
0.1.1	Yes	Input – Test 1
0.1.2	Yes	Input – Test 3
0.1.3	Yes	Input – Test 2
0.2.1	Yes	Input – Test 4
0.3.1	Yes	Input – Test 13&14

0.3.2	Yes	Input – Test 9&10
0.3.3	Yes	Input – Test 11&12
1.1.1	Yes	Processes – Test 10
1.1.2	Yes	Processes – Test 10
1.1.3	Yes	Processes – Test 11
1.1.4	Yes	Processes – Test 12
1.1.5	Yes	Processes – Test 11
1.2.1	Yes	Processes – Test 7
1.2.2	Yes	Processes – Test 8
1.2.3	Yes	Processes – Test 8
1.2.4	Yes	Processes – Test 9
1.3.1	Yes	Processes – Test 14
1.3.2	Yes	Processes – Test 15
2.1.1	Yes	Output – Test 1
2.1.2	Yes	Output – Test 1
2.1.3	Yes	Output – Test 3
2.2.1	Yes	Processes – Test 16
2.2.2	Yes	Processes – Test 16
2.2.3	Yes	Processes – Test 16
2.2.4	Yes	Processes – Test 16
2.2.5	Yes	Processes – Test 16
2.2.6	Yes	Processes – Test 16
2.3.1	Yes	Processes – Test 18
2.3.2	Yes	Processes – Test 18
2.4	Yes	Ranking Dry Run
3.1	Yes	Output – Test 10
3.2	Yes	Output – Test 1
3.3	Yes	Output – Test 4
3.4	Yes	Output Test 5
3.5	Yes	Storage – Test 1 & 2
3.6	Yes	Output – Test 11

Code appendix

```
Imports System.IO
Imports System.Xml
Imports System.Text.RegularExpressions
Imports System.Net
Imports System.Windows.Forms.DataVisualization.Charting

Public Class UserInterface

    'inputs & gui
    Private Property CountryCodes As New Dictionary(Of String, String)
    Private Property IndicatorCodes As New Dictionary(Of String, String)
    Private Property StaticCSVFileAdress As String = ""
    Private Property TwoIndicatorInputs As Boolean = False
    Private Property TwoCountryInputs As Boolean = False

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim StaticCountryAddress As String = ""
        Dim StaticIndicatorAddress As String = ""

        LoadStaticAddress(StaticCountryAddress, StaticIndicatorAddress)
        GetCodes(CountryCodes, StaticCountryAddress, 7, "wb:iso2Code", "wb:name", False)
        GetCodes(IndicatorCodes, StaticIndicatorAddress, 29, "wb:indicator", "wb:name", True)
        CreateListInput(IndicatorCodes, ListOfIndicatorInputs)
        CreateListInput(CountryCodes, ListOfCountryInputs)
        Button_Go.Hide()
    End Sub

    Private Sub LoadStaticAddress(ByRef StaticCountryAddress As String, ByRef StaticIndicatorAddress As String)
        Using reader As StreamReader = New StreamReader("StaticAddressInputs.txt")
            StaticCountryAddress = reader.ReadLine
            StaticIndicatorAddress = reader.ReadLine
            StaticCSVFileAdress = reader.ReadLine
        End Using
    End Sub

    Private Sub GetCodes(ByRef Codes As Dictionary(Of String, String), ByVal StaticAddress As String,
                         ByVal NoOfPages As Integer, ByVal FixedElementName1 As String,
                         ByVal FixedElementName2 As String, ByVal Indicator As Boolean)
```

```
        Dim PageNumber As Integer = 0
        Dim ElementName As String = ""
        Dim Code As String = ""
        Dim Name As String = ""
        Dim HoldAttributeId As String = ""
        Do
            PageNumber += 1
            Try
                Dim reader As New XmlTextReader(StaticAddress & PageNumber.ToString)
                While reader.Read()
                    If XmlNodeType.Element = reader.NodeType Then
                        ElementName = reader.Name
                    End If
                    If Indicator = True Then
                        HoldAttributeId = reader.GetAttribute("id")
                    End If
                    If (XmlNodeType.Text = reader.NodeType) And ElementName = FixedElementName1 And Indicator = False Then
                        Code = reader.Value
                    ElseIf ElementName = FixedElementName1 And HoldAttributeId <> Nothing And Indicator = True Then
                        Code = HoldAttributeId
                    ElseIf XmlNodeType.Text = reader.NodeType And ElementName = FixedElementName2 Then
                        Name = reader.Value
                        Codes.Add(Name, Code)
                    End If
                End While
            Catch ex As Exception
                MessageBox.Show("Connection Error - Try Again")
            End Try
        Loop Until PageNumber = NoOfPages
    End Sub

    Private Sub CreateListInput(ByVal IndicatorDictionary As Dictionary(Of String, String), ByVal ListBox As CheckedListBox)
        For Each kvp As KeyValuePair(Of String, String) In IndicatorDictionary
            ListBox.Items.Add(kvp.Key)
        Next
        ListBox.SelectionMode = SelectionMode.One
        ListBox.CheckOnClick = True
        ListBox.SetItemChecked(0, True)
    End Sub

    Private Sub ListOfIndicatorInputs_SelectedIndexChanged(sender As Object,
```

```
                                                    e As System.EventArgs) Handles ListOfIndicatorInputs.SelectedIndexChanged
        If TwoIndicatorInputs = False Then
            For i = 0 To ListOfIndicatorInputs.Items.Count - 1
                ListOfIndicatorInputs.SetItemCheckState(i, CheckState.Unchecked)
            Next
            ListOfIndicatorInputs.SetItemCheckState(ListOfIndicatorInputs.SelectedIndex, CheckState.Checked)
        Else
            Dim targetNum As Integer = 3
            If ListOfIndicatorInputs.GetItemCheckState(ListOfIndicatorInputs.SelectedIndex) = CheckState.Checked _
                And ListOfIndicatorInputs.CheckedItems.Count + 1 > targetNum Then
                MsgBox("You cannot select more than " & targetNum - 1.ToString() & " items. Please uncheck one to selest another.")
                ListOfIndicatorInputs.SetItemCheckState(ListOfIndicatorInputs.SelectedIndex, CheckState.Unchecked)
            End If
        End If
    End Sub

    Private Sub ListOfCountryInputs_SelectedIndexChanged(sender As Object,
                                                    e As System.EventArgs) Handles ListOfCountryInputs.SelectedIndexChanged
        If TwoCountryInputs = False Then
            For i = 0 To ListOfCountryInputs.Items.Count - 1
                ListOfCountryInputs.SetItemCheckState(i, CheckState.Unchecked)
            Next
            ListOfCountryInputs.SetItemCheckState(ListOfCountryInputs.SelectedIndex, CheckState.Checked)
            Try   ' only gets one country flag
                Dim tClient As WebClient = New WebClient
                Dim temp As String = "https://www.countryflags.io/" & CountryCodes(ListOfCountryInputs.Text).ToLower() & "/flat/64.png"
                Dim tImage As Bitmap = Bitmap.FromStream(New MemoryStream(tClient.DownloadData("https://www.countryflags.io/" _
                                                                                & CountryCodes(ListOfCountryInputs.Text).ToLower() _
                                                                                        & "/flat/64.png")))

                PictureBoxFlag.Image = tImage
            Catch ex As Exception
                Try
                    PictureBoxFlag.Image = Nothing
                    MsgBox("Either the flag does not exist or your internet connection timed out", Title:="Error")
                Catch ex1 As Exception
                    MsgBox("Either the flag does not exist or your internet connection timed out", Title:="Error")
                End Try
            End Try
        Else
            Dim targetNum As Integer = 3
            If ListOfCountryInputs.GetItemCheckState(ListOfCountryInputs.SelectedIndex) = CheckState.Checked _
                And ListOfCountryInputs.CheckedItems.Count + 1 > targetNum Then
```

```
            MsgBox("You cannot select more than " & targetNum - 1.ToString() & " items. Please uncheck one to selest another.")
            ListOfCountryInputs.SetItemCheckState(ListOfCountryInputs.SelectedIndex, CheckState.Unchecked)
        End If
    End If
End Sub

Private Sub NumberOfYearsInput_KeyPress(sender As Object, e As System.Windows.Forms.KeyEventArgs) Handles NumberOfYearsInput.KeyUp
    Dim pattern As String = "^[1-9][0-9]*$"
    Dim rgx As New Regex(pattern)
    Dim match As Match = rgx.Match(NumberOfYearsInput.Text)
    If match.Success Then
        If CInt(NumberOfYearsInput.Text) <= 28 Then
            TextBoxBorderPanel.BackColor = Color.PaleGreen
            Button_Go.Show()
        Else
            TextBoxBorderPanel.BackColor = Color.Tomato
            Button_Go.Hide()
        End If
    Else
        TextBoxBorderPanel.BackColor = Color.Tomato
        Button_Go.Hide()
    End If
End Sub

Private Sub CheckBox_2Indicators_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBox_2Indicators.CheckedChanged
    If CheckBox_2Indicators.Checked = False Then
        Label_2Indicators.Text = "Select 2 Indicators - Off"
        TwoIndicatorInputs = False
    Else
        Label_2Indicators.Text = "Select 2 Indicators - On"
        TwoIndicatorInputs = True
        Label_2Countries.Text = "Select 2 Countries - Off"
        TwoCountryInputs = False
        CheckBox_2Countries.CheckState = CheckState.Unchecked
    End If
End Sub

Private Sub CheckBox_2Countries_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBox_2Countries.CheckedChanged
    If CheckBox_2Countries.Checked = False Then
        Label_2Countries.Text = "Select 2 Countries - Off"
        TwoCountryInputs = False
    Else
```

```
            Label_2Countries.Text = "Select 2 Countries - On"
            TwoCountryInputs = True
            Label_2Indicators.Text = "Select 2 Indicators - Off"
            TwoIndicatorInputs = False
            CheckBox_2Indicators.CheckState = CheckState.Unchecked
        End If
    End Sub

    Private Sub Button_Go_Click(sender As Object, e As EventArgs) Handles Button_Go.Click
        Dim InputsCorrect As Boolean = True
        If ListOfCountryInputs.CheckedItems.Count = 0 Then
            MsgBox("Please enter 1 country input")
            InputsCorrect = False
        End If
        If ListOfIndicatorInputs.CheckedItems.Count <> 2 And TwoIndicatorInputs = True Then
            MsgBox("Please ensure exactly 2 indicators are checked")
            InputsCorrect = False
        End If
        If ListOfIndicatorInputs.CheckedItems.Count <> 1 And TwoIndicatorInputs = False Then
            MsgBox("Please ensure exactly 1 indicator is checked")
            InputsCorrect = False
        End If
        If ListOfCountryInputs.CheckedItems.Count <> 2 And TwoCountryInputs = True Then
            MsgBox("Please ensure exactly 2 countries are checked")
            InputsCorrect = False
        End If
        If ListOfCountryInputs.CheckedItems.Count <> 1 And TwoCountryInputs = False Then
            MsgBox("Please ensure exactly 1 country is checked")
            InputsCorrect = False
        End If
        If InputsCorrect = True Then
            Model()
        End If
    End Sub

    'processes & output
    Private Sub SetParameters(ByRef DataSet As DataSet, ByVal SelectFirstIndicator As Boolean, ByVal SelectFirstCountry As Boolean)
        DataSet.SetNoOfYears(CInt(NumberOfYearsInput.Text))
        Dim CountrySet As Boolean = False
        For Each itemchecked In ListOfCountryInputs.CheckedItems
            If SelectFirstCountry = True And CountrySet = False Then
                DataSet.country.SetCountryName(CStr(itemchecked))
```

```
                CountrySet = True
            End If
            SelectFirstCountry = True
        Next
        Dim IndicatorSet As Boolean = False
        For Each itemchecked In ListOfIndicatorInputs.CheckedItems
            If SelectFirstIndicator = True And IndicatorSet = False Then
                DataSet.indicator.SetIndicatorName(CStr(itemchecked))
                IndicatorSet = True
            End If
            SelectFirstIndicator = True
        Next
        DataSet.country.SetCountryCode(CountryCodes(DataSet.country.GetCountryName))
        DataSet.indicator.SetIndicatorCode(IndicatorCodes(DataSet.indicator.GetIndicatorName))
    End Sub

    Private Sub SetListofDates(ByRef ListOfDates As List(Of Decimal), ByVal NoOfYears As Integer)
        If NoOfYears <> 0 Then
            ListOfDates.Add(2019 - NoOfYears)
            SetListofDates(ListOfDates, NoOfYears - 1)
        End If
    End Sub

    Private Function FetchAPIdata(ByVal CountryCode As String, ByVal IndicatorCode As String, ByVal NoOfYears As Integer)
        Dim xmlfilelocation As String = "https://api.worldbank.org/v2/country/" & CountryCode _
                                        & "/indicator/" & IndicatorCode & "?format=xml"
        Dim FetchedArray As New List(Of Decimal)
        Dim PageNumber As Integer = 0
        Dim ElementName As String = ""
        Dim year As Integer = 0
        Dim value As Decimal = 0
        Dim exitloop As Boolean = False
        Do
            PageNumber += 1
            Try
                Dim reader As New XmlTextReader(xmlfilelocation)
                While reader.Read()
                    If XmlNodeType.Element = reader.NodeType Then
                        ElementName = reader.Name
                    End If
                    If (XmlNodeType.Text = reader.NodeType And ElementName = "wb:date") Then
                        year = reader.Value
```

```
                    If year = 2016 - NoOfYears Then
                        exitloop = True
                    End If
                ElseIf XmlNodeType.Text = reader.NodeType And ElementName = "wb:value" Then
                    value = reader.Value
                    If value = Nothing Then
                        value = 0
                    End If
                    FetchedArray.Add(value)
                    If exitloop Then
                        If FetchedArray.Count() > NoOfYears Then
                            Do
                                FetchedArray.Remove(FetchedArray(0))
                            Loop Until FetchedArray.Count() = NoOfYears
                        End If
                        Exit Do
                    End If
                End If
            End While
        Catch ex As Exception
            MessageBox.Show(ex.Message)
            Exit Do
        End Try
    Loop Until PageNumber = 4
    If FetchedArray.Count < NoOfYears Then
        MessageBox.Show("Complete indicator data currently unavaialble for this country/region - " _
                        & (FetchedArray.Count).ToString _
                        & " is the max number of years available - where data is missing it will be interpolated")
        For i = FetchedArray.Count To NoOfYears - 1
            FetchedArray.Add(0)
        Next
    End If
    FetchedArray.Reverse()
    Return FetchedArray
End Function

Private Sub FetchCSVData(ByVal Country As String, ByVal NoOfYears As Integer, ByRef ListOfHDIorSecondIndicator As List(Of Decimal))
    Dim Lines() As String = File.ReadAllLines("Human Development Index (HDI).csv")
    Dim CSV_2Darray(Lines.Length - 1)() As String
    For I As Integer = 0 To Lines.Length - 1
        CSV_2Darray(I) = Lines(I).Split(","c)
    Next
```

```
        Dim CountryColumn As New List(Of String)
        For i = 2 To 188    ' removes header and footer of the file
            CountryColumn.Add(CSV_2Darray(i)(1))
        Next
        Dim countryindex As Integer
        countryindex = CInt(BinarySearch(CountryColumn, CountryColumn.Count - 1, 0, Country, 0)) + 2 ' +6 acounts for removed header
        If countryindex <> -1 Then
            For i = CSV_2Darray(countryindex).Count - 1 To 2 Step -1 ' removes leading columns
                Try
                    Dim value As String = CSV_2Darray(countryindex)(i)
                    ListOfHDIorSecondIndicator.Add(CDec(value.Replace("""", "")))
                Catch ex As Exception
                    If CSV_2Darray(countryindex)(i) = """.." Then
                        ListOfHDIorSecondIndicator.Add(0) ' where data is unavailable
                    End If
                End Try
            Next
        Else
            MessageBox.Show("HDI currently unavaialble for this country/region")
        End If
        For i = ListOfHDIorSecondIndicator.Count To 29
            ListOfHDIorSecondIndicator.Add(0)
        Next
        ListOfHDIorSecondIndicator.Reverse()
        For j = 0 To 29 - (NoOfYears)
            ListOfHDIorSecondIndicator.Remove(ListOfHDIorSecondIndicator(0))
        Next
    End Sub

    Function BinarySearch(ByVal array As List(Of String), ByVal max As Integer,
                          ByVal min As Integer, ByVal target As String,
                          ByVal DigitNumber As Integer)
        If min > max Then
            Return "-1"
        End If
        Dim mid As Integer = (max + min) / 2
        Dim currentItem As String = array(mid).Replace(("""), "") ' remove leading quote marks
        currentItem = currentItem.Replace(" ", "") ' remove leading space
        target = target.Replace(" ", "") ' match the current item without spaces e.g United Kingdom
        If currentItem.ToLower.ToString = target.ToLower.ToString Then
            Return mid.ToString()
        Else
```

```
        Dim temp As Integer = checkLetters(array, max, min, target, mid, DigitNumber)
        If temp = 1 Then
            Return BinarySearch(array, mid - 1, min, target, 0)
        ElseIf temp = 2 Then
            Return BinarySearch(array, max, mid + 1, target, 0)
        Else
            If DigitNumber = target.Length - 1 Then
                Return "-1"
            Else
                Return BinarySearch(array, max, min, target, DigitNumber + 1)
            End If
        End If
    End If
    Return mid
End Function

Function checkLetters(ByVal array As List(Of String), ByVal max As Integer,
                      ByVal min As Integer, ByVal target As String, ByVal mid As Integer,
                      ByVal DigitNumber As Integer)
    Dim x As Integer = Asc(array(mid).Replace(" ", "").Substring(DigitNumber + 1, 1).ToLower)
    Dim y As Integer = Asc(target.Substring(DigitNumber, 1).ToLower)
    If y < x Then
        Return "1"
    ElseIf y > x Then
        Return "2"
    Else
        Return "3"
    End If
End Function

Private Sub ShowCalculations(ByVal ListOfHDIorSecondIndicator As List(Of Decimal), ByVal ListOfSelectedIndicator As List(Of Decimal))
    Dim calculations As New Correlations(ListOfHDIorSecondIndicator.ToArray, ListOfSelectedIndicator.ToArray)
    Label_PMCC.Text = calculations.PearsonCoefficient().ToString
    Label_Spearmans.Text = calculations.SpearmansRank().ToString
    Label_Percentage.Text = calculations.PercentageScore().ToString & "%"
End Sub

Private Sub ProduceGraphs(ByVal Country As String, ByVal Indicator As String, ByRef DatasetA As DataSet, ByRef DatasetB As DataSet)
    Dim title As String = ""
    If TwoIndicatorInputs = False Then    'long decsion statements are too ensure titles are all corectly labelled to aid user
        If TwoCountryInputs = False Then
            title = "HDI & " & DatasetA.indicator.GetIndicatorName
```

```
            GraphGenerator(Chart_Indicator_Indicator2, DatasetB.getVals, DatasetA.getVals, "Scatter", title)
            title = "Time & HDI"
            GraphGenerator(Chart_Time_Indicator2, DatasetB.GetYears, DatasetB.getVals, "Line", title)
            title = "Time & " & DatasetA.indicator.GetIndicatorName
            GraphGenerator(Chart_Time_Indicator, DatasetA.GetYears, DatasetA.getVals, "Line", title)
        Else
            title = DatasetB.country.GetCountryName & " & " & DatasetA.country.GetCountryName _
                & " - " & DatasetA.indicator.GetIndicatorName
            GraphGenerator(Chart_Indicator_Indicator2, DatasetB.getVals, DatasetA.getVals, "Scatter", title)
            title = "Time & " & DatasetB.country.GetCountryName
            GraphGenerator(Chart_Time_Indicator2, DatasetB.GetYears, DatasetB.getVals, "Line", title)
            title = "Time &" & DatasetA.country.GetCountryName
            GraphGenerator(Chart_Time_Indicator, DatasetA.GetYears, DatasetA.getVals, "Line", title)
        End If
    Else
        title = DatasetB.indicator.GetIndicatorName & " & " & DatasetA.indicator.GetIndicatorName
        GraphGenerator(Chart_Indicator_Indicator2, DatasetB.getVals, DatasetA.getVals, "Scatter", title)
        title = "Time & " & DatasetB.indicator.GetIndicatorName
        GraphGenerator(Chart_Time_Indicator2, DatasetB.GetYears, DatasetB.getVals, "Line", title)
        title = "Time &" & DatasetA.indicator.GetIndicatorName
        GraphGenerator(Chart_Time_Indicator, DatasetA.GetYears, DatasetA.getVals, "Line", title)
    End If
End Sub

Private Sub GraphGenerator(ByRef Chart As Object, ByVal List1 As List(Of Decimal),
                           ByVal List2 As List(Of Decimal), ByVal ChartType As String,
                           ByVal Title As String)
    Try
        Chart.series(0).points.clear()
        Chart.series.clear()
    Catch ex As Exception
        Chart.series.clear()
    End Try
    Chart.Titles.clear()
    Dim series As New Series
    If ChartType = "Scatter" Then
        series.ChartType = SeriesChartType.Point
    Else
        series.ChartType = SeriesChartType.Line
    End If
    For i = 0 To List1.Count() - 1
        series.Points.AddXY(List1.Item(i), List2.Item(i))
```

```
        Next
        Chart.Series.Add(series)
        Chart.titles.add(Title)
    End Sub

    Private Sub SaveResults(ByVal ListOfHDIorSecondIndicator As List(Of Decimal), ByVal ListOfSelectedIndicator As List(Of Decimal),
                            ByVal Country As String, ByVal Indicator As String, ByVal NoOfYears As Integer)
        Dim calculations As New Correlations(ListOfHDIorSecondIndicator.ToArray, ListOfSelectedIndicator.ToArray)
        Dim SaveLine As String = Country & "-" & Indicator & "-" & NoOfYears & ": " _
            & calculations.PearsonCoefficient().ToString & "   " & calculations.SpearmansRank().ToString _
            & "   " & calculations.PercentageScore().ToString & "%"
        Dim dt As Date = Date.Parse(DateTime.Now())
        Dim dateString = dt.ToShortDateString()
        dateString = dateString.Replace("/", "-")
        Dim SaveLocation As String = dateString & ".txt"
        Using writer As New StreamWriter(SaveLocation, True)
            writer.WriteLine(SaveLine)
        End Using
    End Sub

    Function BitwiseCompareData(ByVal List1 As List(Of Decimal), ByVal List2 As List(Of Decimal))
        'bitwise comparision returns binary string
        Dim ByteCode1 As String = ""
        Dim ByteCode2 As String = ""
        For Each item In List1
            If item = 0 Then
                ByteCode1 = ByteCode1 & "0"
            Else
                ByteCode1 = ByteCode1 & "1"
            End If
        Next
        For Each item In List2
            If item = 0 Then
                ByteCode2 = ByteCode2 & "0"
            Else
                ByteCode2 = ByteCode2 & "1"
            End If
        Next
        ByteCode1 = BinaryStringToInteger(ByteCode1)
        ByteCode2 = BinaryStringToInteger(ByteCode2)
        Dim AndResult As Integer = ByteCode1 And ByteCode2 'logical AND bitwise operation
        Dim BinaryAndResult As String = DecimalToBinary(AndResult)
```

```
        If Len(DecimalToBinary(ByteCode1)) > Len(BinaryAndResult) Then
            For i = Len(BinaryAndResult) To Len(DecimalToBinary(ByteCode1)) - 1
                BinaryAndResult = "0" & BinaryAndResult
            Next
        End If
        Return BinaryAndResult
    End Function

    Function BinaryStringToInteger(ByVal binary As String)
        Dim Answer As Integer = 0
        Dim Power2 As Integer = 0
        For i = Len(binary) - 1 To 0 Step -1
            If binary(i) = "1" Then
                Answer += 2 ^ Power2
            End If
            Power2 += 1
        Next
        Return Answer
    End Function

    Private Function DecimalToBinary(dec As Integer) As String
        Dim binary As Integer
        Dim output As String = ""
        While dec <> 0
            If dec Mod 2 = 0 Then
                binary = 0
            Else
                binary = 1
            End If
            dec = dec \ 2
            output = Convert.ToString(binary) & output
        End While
        If output Is Nothing Then
            Return "0"
        Else
            Return output
        End If
    End Function

    Function IsInterpolateable(ByVal ComparisonOutput As String)
        Dim count As Integer = 0
        Dim CountSequential As Integer = 0
```

```
        Dim returnedvalue As Boolean = False
        For Each c As Char In ComparisonOutput
            If c = "0" Then
                count += 1
                CountSequential += 1
                If CountSequential > 4 Then
                    Return False
                    returnedvalue = True
                End If
            Else
                CountSequential = 0
            End If
        Next
        If count > Len(ComparisonOutput) / 2 Then
            Return False
        Else
            If returnedvalue = False Then
                Return True
            Else
                Return False
            End If
        End If
    End Function

    Sub InterpolateData(ByVal List1 As List(Of Decimal), ByVal List2 As List(Of Decimal))
        'interpolate missing values in each of the arrays - works by comparing date to value of each array
        ' shorten array to correct size - so regression doesnt include 0 values
        Dim Looplength As Integer = List2.Count - 1
        Dim NewList1 As New List(Of Decimal)
        Dim NewList2 As New List(Of Decimal)
        For i = 0 To Looplength
            If List2(i) <> 0 Then
                NewList1.Add(List1(i))
                NewList2.Add(List2(i))
            End If
        Next
        Dim calculations As New Correlations(NewList1.ToArray, NewList2.ToArray)
        Dim R As Decimal = calculations.PearsonCoefficient()
        Dim MeanX As Decimal = calculations.MeanX()
        Dim MeanY As Decimal = calculations.MeanY()
        Dim StandardDeviationX As Decimal = calculations.StandardDeviationX
        Dim StandardDeviationY As Decimal = calculations.StandardDeviationY
```

```
        Dim Gradient As Decimal = R * (StandardDeviationY / StandardDeviationX)
        Dim Intercept As Decimal = MeanY - Gradient * MeanX
        For i = 0 To List2.Count - 1
            If List2(i) = 0 Then
                List2(i) = List1(i) * Gradient + Intercept
            End If
        Next
    End Sub

    Private Sub Model()
        Dim ListOfDates As New List(Of Decimal)
        Dim ListOfSelectedIndicator As New List(Of Decimal)
        Dim ListOfHDIorSecondIndicator As New List(Of Decimal)
        Dim DataSetA As New DataSet
        Dim DataSetB As New DataSet
        ListOfDates.Clear()
        ListOfSelectedIndicator.Clear()
        ListOfHDIorSecondIndicator.Clear()
        SetParameters(DataSetA, True, True)
        If TwoIndicatorInputs = True Then
            SetParameters(DataSetB, False, True)
        Else
            If TwoCountryInputs = True Then
                SetParameters(DataSetB, True, False)
            Else
                SetParameters(DataSetB, True, True)
            End If
        End If
        SetListofDates(ListOfDates, DataSetA.GetNoOfYears)
        Try
            ListOfSelectedIndicator = FetchAPIdata(DataSetA.country.GetCountryCode _
                                                   , DataSetA.indicator.GetIndicatorCode _
                                                   , DataSetA.GetNoOfYears)
        Catch ex As Exception
            MessageBox.Show("Error - Incomplete dataset or internet issues - Please try again.")
        End Try
        If TwoIndicatorInputs = True Or TwoCountryInputs = True Then
            Try
                ListOfHDIorSecondIndicator = FetchAPIdata(DataSetB.country.GetCountryCode _
                                                          , DataSetB.indicator.GetIndicatorCode _
                                                          , DataSetB.GetNoOfYears)
            Catch ex As Exception
```

```
                MessageBox.Show("Error - Incomplete dataset or internet issues - Please try again.")
            End Try
        Else
            FetchCSVData(DataSetA.country.GetCountryName, DataSetA.GetNoOfYears, ListOfHDIorSecondIndicator)
        End If
        If IsInterpolateable(BitwiseCompareData(ListOfHDIorSecondIndicator, ListOfSelectedIndicator)) = True Then
            InterpolateData(ListOfDates, ListOfHDIorSecondIndicator)
            InterpolateData(ListOfDates, ListOfSelectedIndicator)
            Dim counter As Integer = 0
            For Each item In ListOfDates
                Dim x As New YearValue
                x.year = item
                x.value = ListOfSelectedIndicator(counter)
                DataSetA.yearValues.Add(x)
                Dim y As New YearValue
                y.year = item
                y.value = ListOfHDIorSecondIndicator(counter)
                DataSetB.yearValues.Add(y)
                counter += 1
            Next
            ShowCalculations(ListOfHDIorSecondIndicator, ListOfSelectedIndicator)
            ProduceGraphs(DataSetA.country.GetCountryName, DataSetA.indicator.GetIndicatorName, DataSetA, DataSetB)
            SaveResults(DataSetB.getVals, DataSetA.getVals, DataSetA.country.GetCountryName _
                        , DataSetA.indicator.GetIndicatorName, DataSetA.GetNoOfYears)
        Else
            MessageBox.Show("Not enough comparable data to interpolate effectively - please alter the restraints")
        End If

    End Sub

End Class


Class Correlations
    Private DataSetA(), DataSetB() As Decimal

    Public Sub New(DataSetA() As Decimal, DataSetB() As Decimal)
        Me.DataSetA = DataSetA
        Me.DataSetB = DataSetB
    End Sub

    Public Function PearsonCoefficient()
```

```
        Dim SumXY As Decimal = 0
        Dim SumX As Decimal = 0
        Dim SumY As Decimal = 0
        Dim SumXSquared As Decimal = 0
        Dim SumYsquared As Decimal = 0
        Dim array2item As Decimal
        For i = 0 To DataSetB.Length - 1
            If DataSetB(i) > 100000 Then
                array2item = Math.Round(DataSetB(i) / 10000) ' for large values so it doesn't crash
            Else
                array2item = DataSetB(i)
            End If
            SumX += DataSetA(i)
            SumY += array2item
            SumXY += DataSetA(i) * array2item
            SumXSquared += DataSetA(i) ^ 2
            SumYsquared += array2item ^ 2
        Next
        Return Math.Round((DataSetA.Length * SumXY - SumX * SumY) / (Math.Sqrt((DataSetA.Length * SumXSquared - SumX ^ 2) * (DataSetA.Lengt
h * SumYsquared - SumY ^ 2))), 4)
    End Function

    Public Function SpearmansRank()
        Dim DataSetHoldA(DataSetA.Length - 1) As Decimal
        DataSetHoldA = DataSetA
        Dim DataSetHoldB(DataSetB.Length - 1) As Decimal
        DataSetHoldB = DataSetB                          ' these holding datasets prevent the ranked array being used in later calcualtion
        DataSetA = Rank(DataSetA)
        DataSetB = Rank(DataSetB)
        Dim answer As Double = PearsonCoefficient()
        DataSetA = DataSetHoldA
        DataSetB = DataSetHoldB
        Return answer
    End Function

    Private Function Rank(ByVal Array() As Decimal)
        Dim array1(Array.Length - 1) As Decimal
        For i = 0 To Array.Length - 1 Step 1
            Dim r As Integer = 1 ' rank increases if it's bigger than a value
            Dim s As Integer = 1 ' same number counter
            For j = 0 To i - 1 Step 1
                If Array(j) < Array(i) Then
```

```
                r = r + 1
            ElseIf Array(j) = Array(i) Then
                s = s + 1
            End If
        Next
        For j = i + 1 To Array.Length - 1 Step 1
            If Array(j) < Array(i) Then
                r = r + 1
            ElseIf Array(j) = Array(i) Then
                s = s + 1
            End If
        Next
        array1(i) = r + (s - 1) * 0.5
    Next
    Return array1
End Function

Public Function PercentageScore()
    Return Math.Round((70 * Math.Abs(PearsonCoefficient())) + (30 * Math.Abs(SpearmansRank())))
End Function

Public Function MeanX()
    Dim total As Decimal = 0
    For Each item In DataSetA
        total += item
    Next
    Return total / DataSetA.Count
End Function

Public Function MeanY()
    Dim total As Decimal = 0
    For Each item In DataSetB
        total += item
    Next
    Return total / DataSetB.Count
End Function

Private Function StandardDeviation(ByVal Array() As Decimal, ByVal mean As Decimal)
    Dim SumOfXMinusMean As Decimal = 0
    For Each item In Array
        SumOfXMinusMean += (item - mean) ^ 2
    Next
```

```
        Return Math.Sqrt(SumOfXMinusMean / (Array.Count - 1))
    End Function

    Public Function StandardDeviationX()
        Return StandardDeviation(DataSetA, MeanX())
    End Function

    Public Function StandardDeviationY()
        Return StandardDeviation(DataSetB, MeanY())
    End Function
End Class

Class Country
    Private Property CountryName As String
    Private Property CountryCode As String

    Public Sub SetCountryName(ByVal input As String)
        CountryName = input
    End Sub

    Public Sub SetCountryCode(ByVal input As String)
        CountryCode = input
    End Sub

    Public Function GetCountryCode()
        Return CountryCode
    End Function

    Public Function GetCountryName()
        Return CountryName
    End Function
End Class

Class Indicator
    Private Property IndicatorName As String
    Private Property IndicatorCode As String

    Public Sub SetIndicatorName(ByVal input As String)
        IndicatorName = input
    End Sub

    Public Sub SetIndicatorCode(ByVal input As String)
```

```
        IndicatorCode = input
    End Sub

    Public Function GetIndicatorName()
        Return IndicatorName
    End Function

    Public Function GetIndicatorCode()
        Return IndicatorCode
    End Function
End Class

Class DataSet
    Public Property country As New Country
    Public Property indicator As New Indicator
    Public Property yearValues As New List(Of YearValue)
    Private Property NoOfYears As Integer

    Public Sub SetNoOfYears(ByVal input As Integer)
        NoOfYears = input
    End Sub

    Public Function GetNoOfYears()
        Return NoOfYears
    End Function

    Public Function getVals()
        Dim x As New List(Of Decimal)
        For Each item In yearValues
            x.Add(item.value())
        Next
        Return x
    End Function

    Public Function GetYears()
        Dim x As New List(Of Decimal)
        For Each item In yearValues
            x.Add(item.year())
        Next
        Return x
    End Function
```

```
End Class

Class YearValue
    Public Property year As Integer
    Public Property value As Decimal
End Class
```