Visual Analytics Exploration of Income Inequality

Arshad Ahmed, *MSc Data Science 2015/16, City University London*

Abstract—We explore the trends in income inequality across the globe using data from the World Bank from the period 1980-2010. The indicator we use to assess Income Inequality is the Percentage of Income held by the top 10% of the population in a country. We find that for this type of spatiotemporal data Visual Analytics is a useful analysis methodology. Here we illustrate the benefits of combining the computational and visual methods of analysis. Our application of the Visual Analytics methodology leads us to some surprising findings regarding the Income Inequality and our derived attributes also allow us to pick out the countries which have much greater Income Inequality compared to other nations.

Index Terms—Visual Analytics, Maps, Clustering,Computational Statistics

INTRODUCTION

The aim of this study is to investigate the income inequality trends through the World Bank data from the period 1980 – 2010. This range is chosen because the target indicator, which is the percentage of income held by the top 10% of the population of a country is not available prior to this period. To reduce the sparsity in the data this range is deemed most useful.

To conduct this analysis we use computational and visual methods. The Visual Analytics approach is found to be particularly useful due to the spatiotemporal nature of the data. The computations allow us to derive meaningful attributes from the data but it is only when presented on a map that the usefulness of the data is appreciable. We highlight this with examples.

The analysis is conducted in through a variety of tools such as IPython, Matlab and Tableau. This paper is organised as follows: in section 1 we conduct an extensive literature review to inform us of the clustering techniques and visual methods that would support the application of the visual analytics methodology on this data. In section 2, we detail the analysis steps on the data, from the acquisition, processing and exploratory analysis and present the results through a combination of visual methods. In section 3, we present our discussion and finish with our conclusions.

1 LITERATURE REVIEW

Clustering is an unsupervised learning technique that aims to find structures in data utilising either partitional, hierarchical, density, grid or model based methods [1]. Intuitively they can be thought of as grouping different objects within a dataset to groups of objects with similar properties. Therefore after clustering the properties of the items in a cluster should be more similar than those of items in a different cluster [2], [3].

Clustering is called an unsupervised learning technique because in contrast to supervised methods like classification there do not exist any labels in the data. Thus the structure of the data must be learnt from the data. [3]

Clustering is difficult due to the many factors that must be considered and addressed before a successful clustering algorithm can be implemented. These factors are derivation of effective similarity measures, criterion functions and initial conditions. Also, the authors in [3] note that no clustering method is able to handle all aspects of a cluster structure such as different shapes, sizes and densities successfully. Inherently, some methods are better at handling some of these structural properties than others. We will review such methods

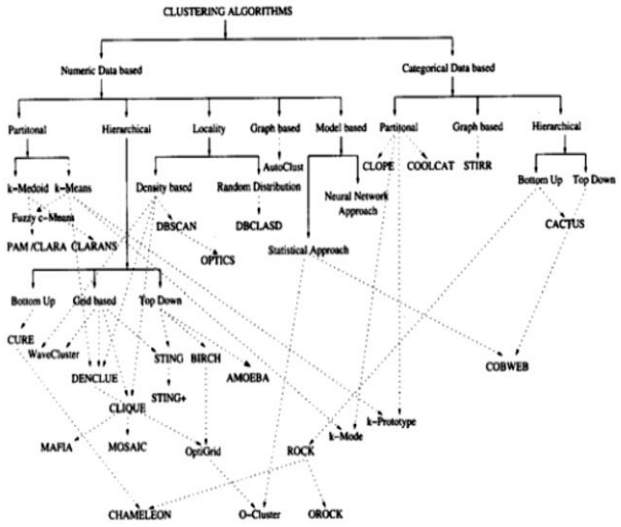

Figure 1: Categorization of clustering algorithms [3]

and explore their strengths and weaknesses which will inform our choice of technique to use on this dataset.

The data type that a clustering algorithm operates on can be used for its taxonomic classification. The primary data types of interest are numerical and categorical. In our analysis of the World Bank Data we are dealing with numerical data only. These algorithms that operate on such data types can then be divided into the categories shown in Fig 1. These can then be further subdivided for example hierarchical methods are classed as being bottom up, grid based or top down. Locality based algorithms are classed as being if density or random distribution based. Model based algorithms can be classed as being statistical or neural network based. Also some algorithms combine mix and match concepts from these different approaches and some can also handle mixed data. [1]–[5]

1.1 Partitional Clustering

Partitional clustering algorithms finds divisions in the data rather than an agglomerative structure produced by hierarchical methods. These methods have the benefit of being scalable to large datasets and hence are widely used. Dendograms produced by agglomerative or hierarchical methods can become prohibitive computationally when dealing with large datasets. The partitions are produced my optimizing some criterion function defined either globally or locally. Since the

potential combinations of labels within the datasets are large in practice these algorithms are run multiple times with different starting states to identify the best clustering output. Partitional methods are divided into centroid and medoid algorithms. The centroid methods represent a cluster by the center of gravity of the instance while the medoids methods represent each cluster by the means of instances that are closest to the center of gravity. [2], [3]

The K-means algorithm is a centroid based method which partitions data into k disjoint partitions by minimizing the objective function in Eq. 1.

$$E = \sum_{i=1}^{K} \sum_{O \in C_i} |O - \mu_i|^2.$$

Equation 1: K-means objective function

Here, the individual points are represented by O in a cluster, C_i where μ, represents the mean of the objects or the centroid of C_i. The squared distances between the objects within a cluster are minimized from their cluster centres. The K-means is a fast and simple algorithm which has complexity of *O* (*I*k*n*) where *I* is the number of iterations and *k* is the number of clusters. Some examples of k-medoids methods are CLARA and CLARANS. The CLARA method determines the medoids of the dataset through an iterative optimization and classifies samples of the dataset into partitions. Assuming a random sampling the medoids inferred from the samples are taken to represent the medoids of the dataset. The CLARANS methods uses a randomized graph search for the medoids in contrast to CLARA. [3]

1.2 Hierarchical Clustering

Hierarchical clustering algorithms typically fall into the single-link, complete link and minimum variance approaches. The difference between these methods are their similarity measures for computing clusters. In the single link case the distance between two clusters is defined as the minimum distance between all pairs of points drawn from the two cluster i.e. one point from cluster 1 and one point from cluster 2. In the complete link case, the distance between two clusters is defined as the maximum of all pairwise distances between the points in two clusters. [2]

These algorithms perform a sequence of partitioning operations either in a bottom up way which is like performing repeated consolidations of data points until a user defined threshold is reached. This can also be done top down where recursive partitions of the data are computed until a threshold is reached. [3]

Some examples of these methods are CURE, ROCK, and BIRCH. The CURE algorithm is bottom up method which use well formed group of points to define intra cluster distance rather than a centroid based approach. CURE begins by choosing a constant number of well scattered points from a cluster and then shrinks the selected points towards the centroid of the cluster using some predetermined fraction. ROCK uses concept of links to measure similarity of a pair of points. The number of links is the number of common neighbors of these points. The merging of clusters also utilizes links instead of distances which allow this method to be extended to non-metric similarity measures. BIRCH is an integrated hierarchical clustering algorithm that uses clustering feature and clustering feature tree to summarize clusters. This allows the BIRCH method to be scalable to large datasets, while being fast and is also suitable for clustering of incremental and/or dynamic inputs. This method applies multiple phases of clustering, where the first phase produces a basic clustering and further iterations can be applied to refine its output. [3]

1.3 Density Based Clustering

Density based clustering methods make use of the notion of a nearest neighbor distance for cluster assignment. A cluster is grown till the density of points in a neighborhood exceeds a predefined threshold. This means that for each point within a given cluster the neighborhood of a given radius (Eps) has to contain a minimum of number of defined points (MinPts). These methods can discover arbitrary cluster shapes and can be used to filter noise. [2], [3]

Some common density based clustering algorithms are DBSCAN and OPTICS. The DBSCAN algorithm separates data points into three classes consisting of core points, border points and noise points. The core points are those that are inside the cluster. A point can be considered an interior point if there it has enough neighborhood points. The border points are those that do not have sufficient neighborhood points to be classed as a core point but are within the neighborhood of a core point. Noise points are those that do not fall within the previous two classes. DBSCAN finds clusters from a dataset by starting with an arbitrary instance and then retrieving all instances that fall within the Eps and MinPts. OPTICS is an extension to DBSCAN that computes an augmented clustering ordering that allows for data clustering. This ordering of the data represents a density based clusters of the data that is equivalent to those obtained by varying the range of parameters. It does this by considering the minimum radius that would make a neighborhood valid for consideration given the MinPts and then extends them to the maximum value. [3]

1.4 Grid Based Clustering

Grid based approaches divide the data space into a finite number of cells to form a grid and then all clustering operations are performed on this grid. This makes grid based methods fast and dependent only on the number of cells in the grid. An example grid based methods are WAVECLUSTER. [3]

WAVECLUSTER maps the input data to a higher dimension feature space and then applies a wavelet transform to the feature space. The algorithm then identifies dense regions within this transformed feature space by looking for connected components. Treating the transformed space as a signal, a rapid change in the distribution of objects corresponds to high frequency areas which are used to identify clusters whereas low frequency areas are those that are outside the clusters. [3]

1.5 Model Based Clustering

Model based clustering assumes that the data are generated by a mixture of probability distributions. The statistical approach to model based clustering often assumes a Gaussian mixture model which then allows for maximum likelihood type approaches such as Expectation Maximization to be applied iteratively to arrive at parameter vectors of the component densities. However, non-parametric density estimation methods such as those based on the Parzen window have been used to search for bins with large counts in a multi-dimensional histogram of the input data. The other approach is to use neural networks. Both approaches attempt to improve the fit of the data to an underlying model. The Self Organizing Map (SOM) is the best known method in this category. [2], [3]

An SOM can be thought of as a two layer neural network, where each neuron represents an n-dimensional weight vector. The dimensions of the weight vector corresponds to the dimensions of the input data. The SOM is trained iteratively and the neurons act like the centers of the cluster. At each training step a vector is chosen at random from the input and then the distance between it and all the weight vectors are calculated using some distance measure. After this step the neuron with the weight vector which most closely matches the weight of the input vector are moved closer to the input vector. The topological neighbors of these matching units are treated in a similar manner. The SOM is a very robust technique that can be used for outlier detection and can deal with missing values. [2], [3]

1.6 Graph Based Clustering

Graph based methods are noted by the authors [3] as still maturing. In [2] the hierarchical clustering methods are explained in terms of subgraphs of the input data. One such method described is AUTOCLUST which extracts boundaries based on Voronoi modelling and Delauney Diagrams. Instead users specifying parameters, the algorithm calculates them from the proximity of structures during the Vornoi modelling phase and from the Delauney Diagram. This approach has multiple benefits as it reduces exploration time, allows for detection of clusters of arbitrary densities in addition to sparse clusters that are near dense ones. [2], [3]

1.7 Review of Cartographic Visualization and Color

We conduct a brief search of the literature to get a sense of what type of map visualizations have been proposed and what their relative strengths and weaknesses are. Since the data relates to countries a Visual representation through maps would be very effective.

With regards to map visualisations we identify a few type below from [6]:

- ⊕ Chloropleth Maps
- ⊕ Cartogram
- ⊕ Dot distribution Maps
- ⊕ Proportional Symbol Maps
- ⊕ Dasymetric Maps

Chloropleth maps are thematic maps in which areas are shaded by color using some attribute. The colors can be inferred from the values being visualised and depending on the nature of the attribute the colors chosen can be harmonic, contrastive or clashing.

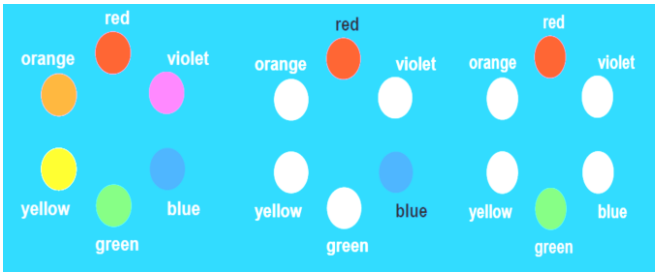

Figure 2: (Left) A color wheel showing harmonic colors. Harmonic colors are those that are next to each other on the wheel such as yellow and green. (Middle) Shows contrastive colors which are colors with one neighbor skipped such as red and blue. (Right) Shows clashing colors which are two neighbors removed.

Cartograms are thematic maps that deform the area of the map using the variable that is to be represented. Dot distribution maps represent density of an attibute by using scatter to shown spatial patterns. The value of the attribute is represented as dots. The proportional symbol techniques uses symbols of different sizes to represent data associated with different locations.the dasymetric map is an extension or alternative to the chloropleth map where the attributes are represented by enumeration units on the map so the regions appear uniform but additional information is used to model the internal distribution of the variable. So it has an additional distribution layer on top of the chloropleth map.

For our analysis we will use mainly chlorpleth maps with a divergent color scheme. The reason for ths being that even though we have shown some of the other visualisations that are available they are not appropriate given the data. The data from the World Bank is it a country level and at the year level. So it is a highly aggregated time series and a lot of the data at the year for our time range is missing for a lot of countries. Therefore in the analysis we resort to using further aggregations such as percentage change in the attributes over 10 years and yearly averages for all countries to get a sense of global trends. Therefore chloropleth maps represent the simplest and easiest representaion of our data. Using a area deforming approach would not be ideal because smaller countries next to big countries with similar trends would not be as visible. Also as the data is at the country level it would make no sense to use proportional symbols or daysmetric methods as there is insufficient granularity in the data.

In the next section we will explain in detail the analysis and analytical steps that we took with the data starting from the acquisition, processing, exploratory analysis and final clustering. With regards to the choice of clustering methods, we have presented a wide variety of techniques from the literature. However, upon closer inspection of the implementations available in open source tools such as IPython is has been found that most of the newer techniques do not have stable implementations hence they will be excluded from our analytical methods.

The most widely implemented clustering techniques we found were for partitional, hierarchical and density based methods. Of these techniques the partitional and density based methods are preferred as they can be interpretted as clustering countries with high and low income inequaliuty together. The interpretation of the hierarchical clusters are more vague hence we exclude these methods for this analysis.

2 DATA ANALYSIS

As mentioned in this analysis we use datasets available freely from the World Bank (WB). The indicators that we use along with their WB code as follows:

- ⊕ Gross Domestic Product (GDP) : NY.GDP.MKTP.KN
- ⊕ Total Population: SP.POP.TOTL
- ⊕ Income share held by highest 10%: SI.DST.10TH.10

The Income share held by highest 10% is our variable of interest which we will take to represent a measure of income inequality. The other variables are used to derive ratios of Income

Inequality to GDP and Population to help us understand trends in Income Inequality (II) better.
This analysis is conducted using IPython [7] and the PANDAS[8], MATPLOTLIB[9], SEABORN[10], SCIPY[11] and SCIKIT-LEARN[12] packages. For the mapping we use Tableau[13].

Data Acquisition and Preprocessing

The first step in our analysis was to source the data. We did so using the WB API built into the PANDAS library in IPython. We used the codes for the different variables and saved them into a Dataframe object for the time ranges between the years 1980-2010.

After the initial data loading stage upon inspecting the data for missing values we noticed that large parts of the data in the earlier time ranges were missing for a lot of countries and also that in addition to the countries there additional rows with aggregates like the totals for High, Low, Medium Income countries etc. So in addition to the missing data we had to delete these entries as well from the data as it would bias our calculations. Therefore we filled all missing values with zeros and kept only the records pertaining to actual country names. This eventually yielded 7688 points for all the countries over the time span considered.

Exploratory Analysis

The initial analysis of our data gave a good sense of what to expect but relying purely on analytical tools was found to be limiting and it also highlights the utility of the Visual Analytics methodology. Initially, the assumption was that strong II would be observed in mainly low to medium income countries which had recently undergone periods of great economic growth. But when the results are presented on a map we found surprises in places we did not expect such as Europe. This is a good example of how the Visual Analytics method can support and complement analytical methods.

Once, we acquired and filtered the raw data we plotted a correlation matrix shown in Fig 2 to see the relationships among the variables in our dataset.

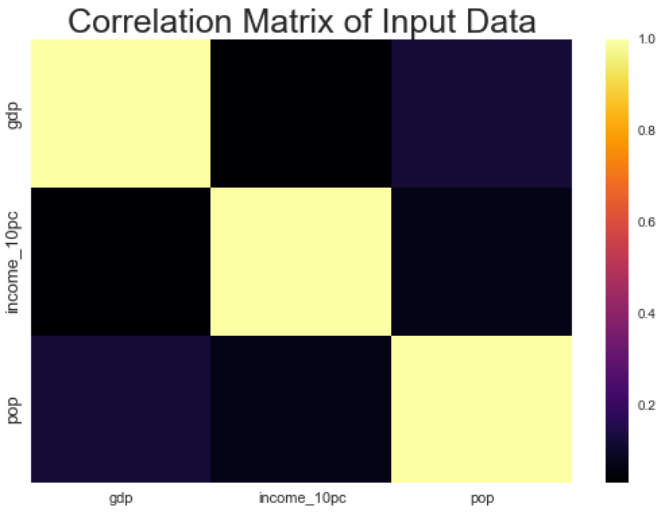

Figure 3: Pearson Correlation Coefficient among indicators in the input data.

From this plot we already start to see the correlations among our variables and get a sense of what to expect. We observe that all the variables are negatively correlated none of the variables show a positive correlation among them. This means that based on this data we can expect to see that II increases even if GDP and population decrease or it can mean that wealth concentration at the top is much faster than it is for the other 90% of the population so we don't get a linear mapping of the variables. This would be an illustration of the rich get richer principle.

The next step is to visualize the individual indicators and get a sense of their scale.

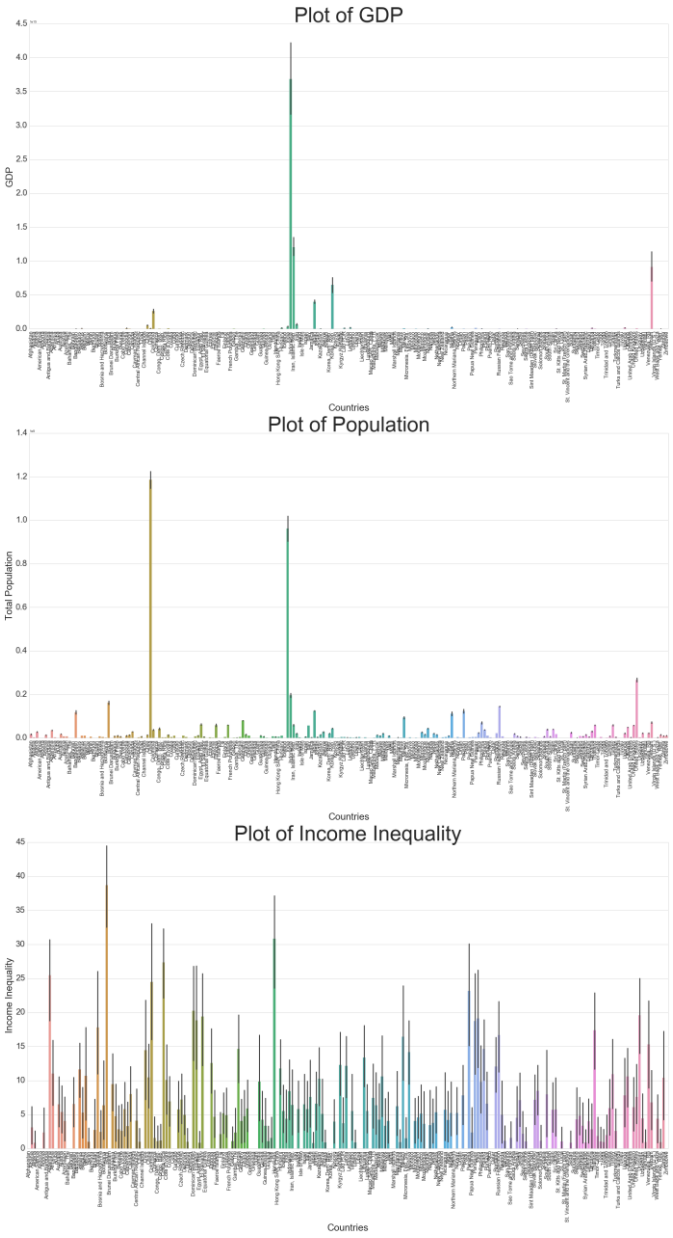

Figure 4: (top) Plot of GDP of countries, (mid) plot of total population and (bottom) of II. Since our time series has missing data some countries have more points than others.

From fig 4, we see that traditional graphical visualization is inadequate for this dataset. Since the time series has missing values and scales are varied, there is a need to scale, aggregate and then map these attributes for meaningful

analysis. This highlights one the ways in which **computation informs the visualization.** The density of points requires better visualization than charts.

As we have already identified a need for data scaling and aggregation. The next step was to perform the scaling. For this we choose scaling to [0, 1] interval. This has the effect of bringing all the indicators to scale that allows for meaningful comparison.

After the scaling, we calculate additional attributes based on these indicators. These are as follows:

- ⊕ II_GDP = II / GDP
- ⊕ II_Pop = II / Pop
- ⊕ II_GDP_Pop = SQRT(ABS(II / (GDP+Pop)))
- ⊕ Percentage Change over 30 years for GDP, Pop and II

The motivation for deriving these attributes are to incorporate the different variables into one metric that can be easily visualized and interpreted.

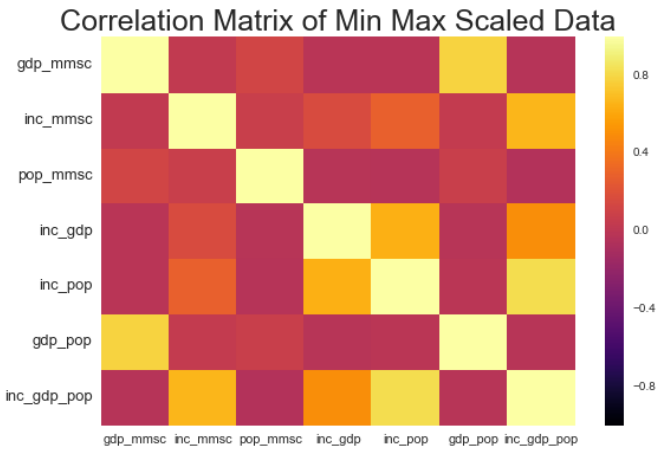

Figure 5: Pearson Correlation Coefficient Matrix of scaled data with additional attributes

We observe from Fig 5, that the scaling has not altered the correlation relationships in the data. But what is interesting to see is that the addition of our attributes are showing some positive correlations with some of our original indicators. The II, GDP and Population indicators exhibits a weakly positive correlation to the II_GDP_Pop variable possibly suggesting that relationship between the underlying variables are captured well in this metric.

Since it would be impractical to present maps of these attributes at different time periods which is more suitable for interactive analysis. The data is aggregated for this analysis so we get an overall understanding of the general trend in II across globe over this time period.

The first aggregation we perform is by countries and by compressing the attributes over the time series by their mean. This has the beneficial effect of us not having to deal with missing values and we end up with values for all the countries over the time period for a more representative contribution. Even though other data imputation techniques are available the reason this is not used is because of the asymmetry in the nature of the missing data. Some countries have more data missing than others so filling values would give more weight to existing values which may or may not be representative of the past. Therefore filling with zeros and aggregating the variables by their means is preferred approach in this analysis.

The next step was to visualize the attributes in the form of a choropleth map shown in Fig 6.

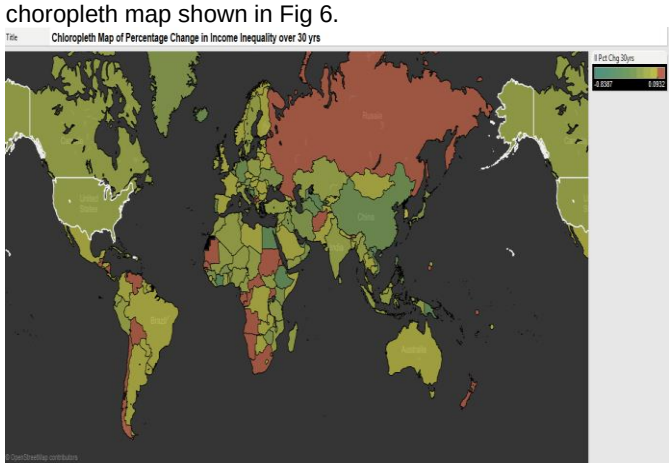

Figure 6: Choropleth map of Percentage Change in II over 30 years

Fig 6, shows the global trends in the II over the time period. Since the attribute represented is the percentage change both positive and negative values are of interest and this is mapped using a divergent color scheme. The dark greens represent positive values while the lighter shades towards the red represent negative values. From this we note that Russia, experienced the most positive change in the attribute the other countries are close to zero hence show up as dark colors. Most of the world experienced a negative change in the percentage of II over the 30 year period with Canada and US experiencing the greatest change.

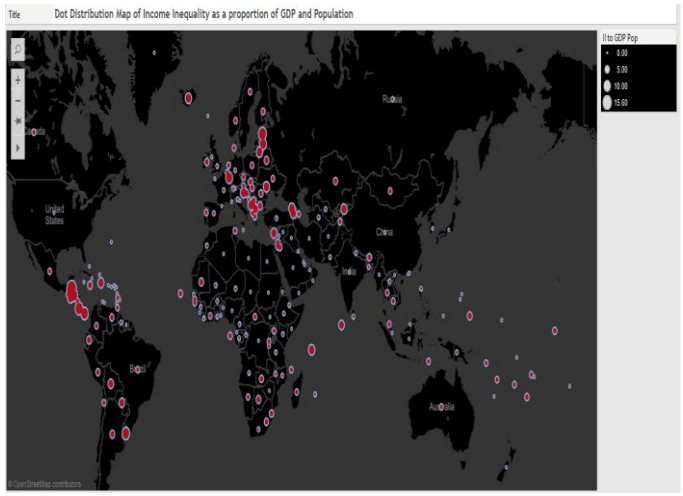

Figure 7: Dot distribution map of II as a proportion of GDP and Population.

Although fig 7, gives us an idea of the spatial distribution of the highest II regions as a proportion of GDP and Population. Here the visual mark used is a circle and the attribute is encoded into the size of the circle. This is not very helpful when we want to know which countries they are specifically. This is the benefit of the Visual Analytics methodology that we can get an overview through a map and then get more details on demand. To find

the country names we present the map as a tile map sized by the attribute.

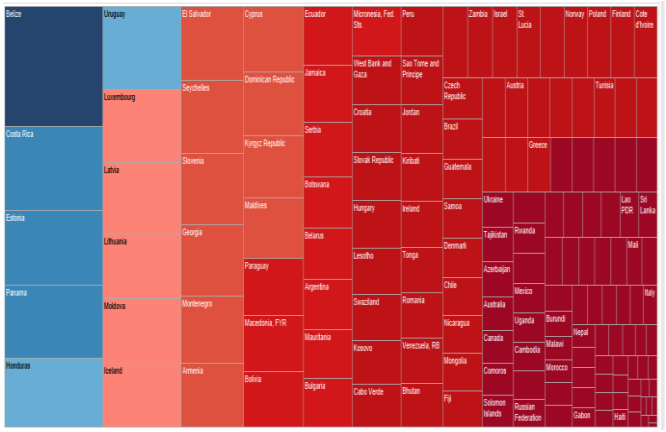

Figure 8: Tile map of II as a proportion of GDP and Population

Fig 8, is more helpful as we can see that Belize has an anomalously large value for this metric this is due to the fact that it has a relatively small population and GDP so when this is scaled the values are close to zero but the II variable scales to around 15 giving such as large value. Some of the other countries are not surprising given their reputation as tax havens such as Panama and Luxembourg. Iceland's appearance in this list alongside Moldova and El Salvador is interesting and unexpected.

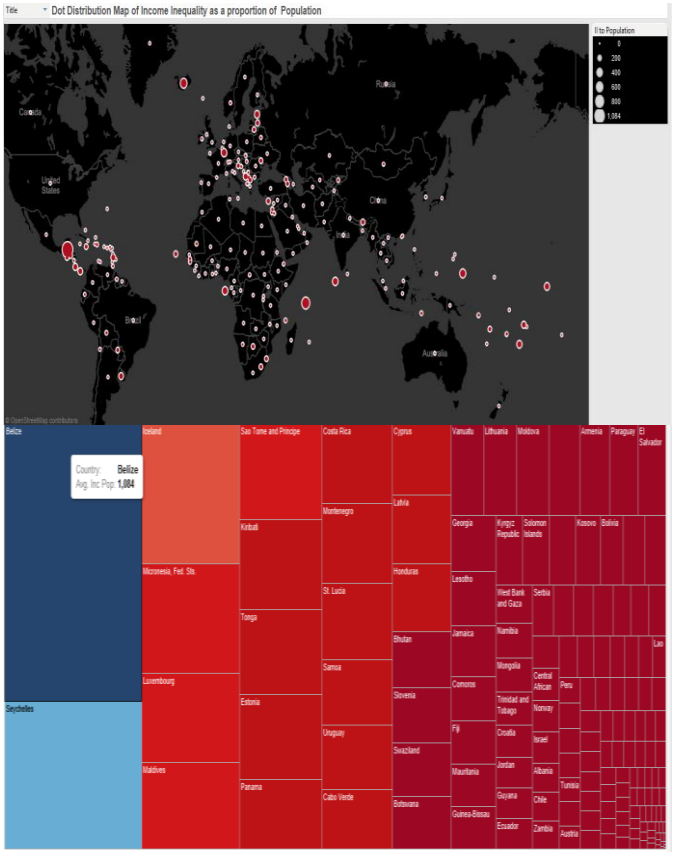

Figure 9: (Top) Dot distribution and (bottom) tile map of II as a proportion of Population.

From Fig 9, we see that when we segment our data in terms of II as a proportion of Population, Belize again comes at the top followed by some island nations such as Seychelles, Maldives and Iceland and we see Luxembourg again. Given its small population and being generally wealthy it is perhaps unsurprising but it does show that even though Iceland and Luxembourg are fairly wealthy nations but they are very unequal in their wealth distributions. This is a surprising finding.

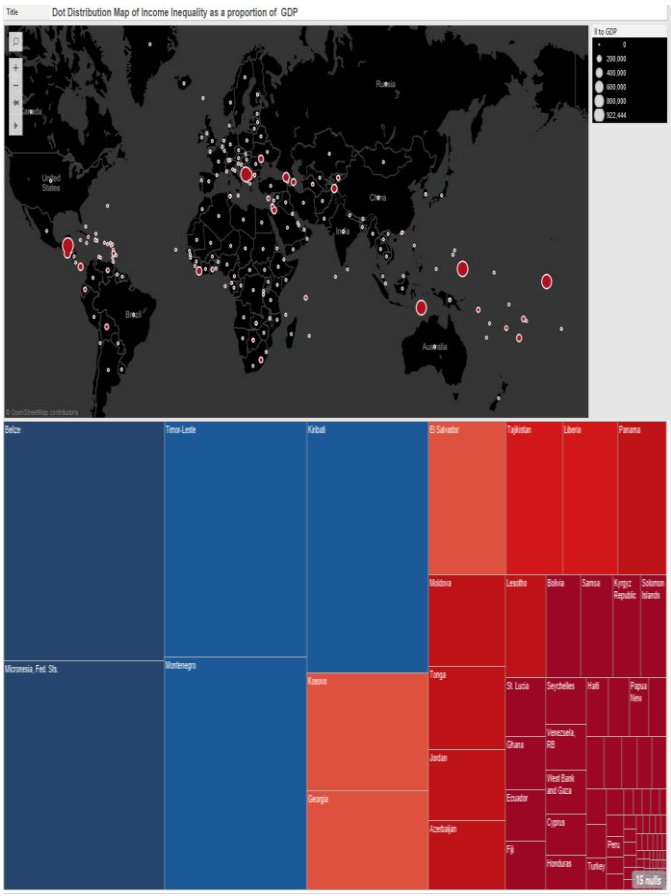

Figure 10: (Top) Dot distribution and (bottom) tile map of II as a proportion of GDP.

We see a slightly different picture emerge when we consider II as a proportion of GDP. Belize is still the winner here but we see other nations such as Kosovo and Montenegro emerge closer to the top. This could be the result of the wars in the region in the 90's and asymmetric wealth distribution after that. Georgia and Jordan also make it here on the list.

At this stage we can draw some initial conclusions that a traditional graphic approach to representation would have made these insights more difficult to get at but with Visual Analytics we are able to get an overview and then get details on demand. Also the value of the attributes calculated are obvious given the figures presented. The metrics allow for the different indicators to be combined in an effective way that makes the route to insights shorter. Depending on the metric we see that II manifests across nations differently based on how we choose to analyze it.

So far we have presented the results of our analysis aggregated by country but what are the trends that we observe when we aggregate by year? We also wanted to explore the global trends year of year. Therefore we present briefly this analysis in Fig 11.

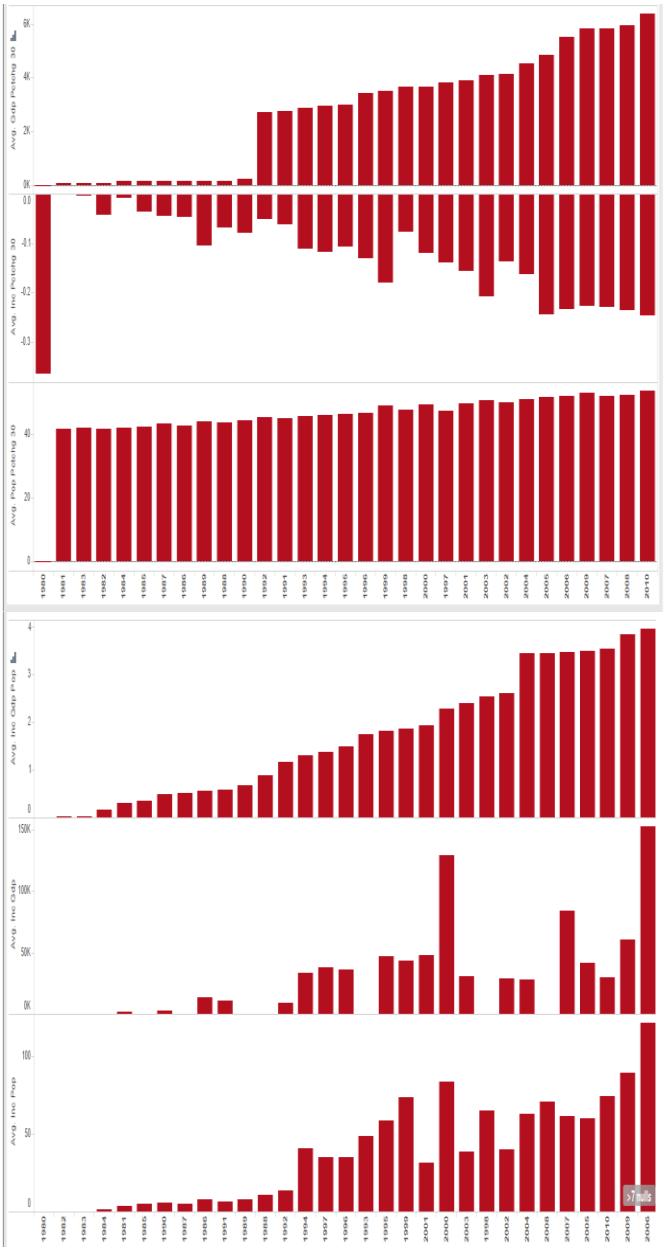

Figure 11: (Top) Yearly Trends in Percentage Change of Indicators from 1980-2010. (Bottom) Trends in attributes over the same time period.

K-Means Clustering

We will now present the results of clustering on this dataset. We use K-means clustering to find the optimal number of clusters in the data. This clustering is performed in Matlab. We use the dataset aggregated by country using the Min Max Scaling. We keep only the numerical features and then evaluate the Silhouette Value for the clustering solution with different values of K. We evaluate the k values of [2, 3, 5, 7, 9, 11] on this data evaluate the silhouette value for each solution and then plot the mean silhouette value against increasing number of clusters.

The Silhouette value measures the similarity of inter cluster objects to that of intra cluster objects. It can be thought of as a measure of cohesion between objects in a cluster compared to the separation of clusters. This value ranges from -1 to 1. A high value indicates a good match to its current cluster and a low value indicates a poor match to neighboring clusters. If the objects derived from the clustering solution have mostly high values then the configuration can be thought of as being optimal while if there are many negative points than this indicates that we have defined an insufficient number of clusters for the data.

The results are shown in Fig 12 and 13. We observe that the optimal number of clusters for our data is 2 and that increasing the number of clusters hardly improves separation of the clusters in the silhouette plots but the silhouette score declines relatively fast as we increase the cluster number. This gives us confidence in the solution derived.

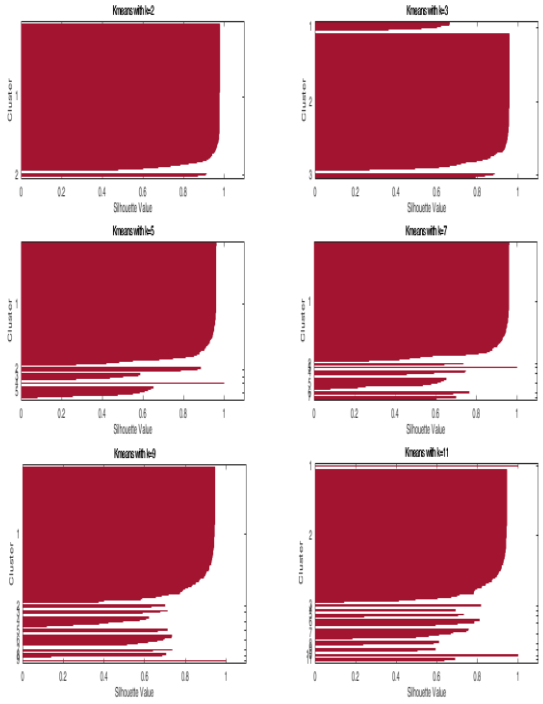

Figure 12: Silhouette value for different values of K

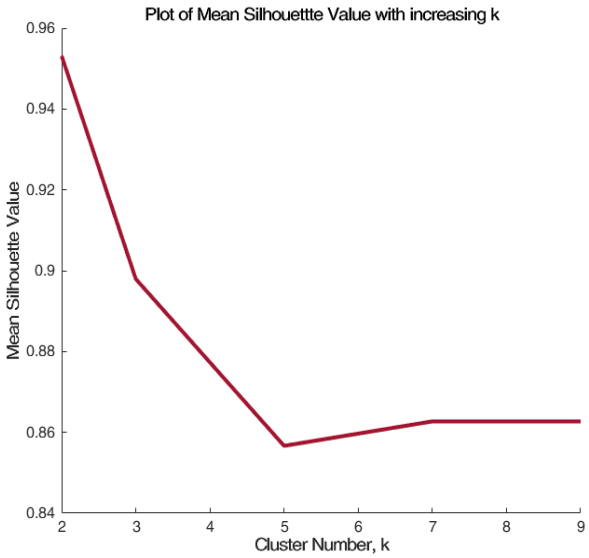

Figure 13: Mean Silhouette Value plot with increasing K values

Self-Organizing Map (SOM)

As an additional verification of the clustering solution derived with K-means we try another clustering method to see if we get the same results. We try the neural network analogue of K-means the SOM. We cluster the data with an SOM using a [5, 5], [10, 10] and [15, 15] map. Since the SOM network takes a square form the dimensions translate to 25, 100 and 225 neurons in the network. The input data has a weight vector of 214. So the first two set ups cluster and reduce dimensionality of the data. For the SOM we present the U-Matrix which is a two dimensional representation of the higher dimensional luster centers. The blue hexagons in these plots represent the neurons and the red lines connect the neighboring neurons. The colors in the regions that contain the red lines indicate the distances between the neurons in the SOM network. The darker colors represent greater distances, while lighter colors represent smaller distances.

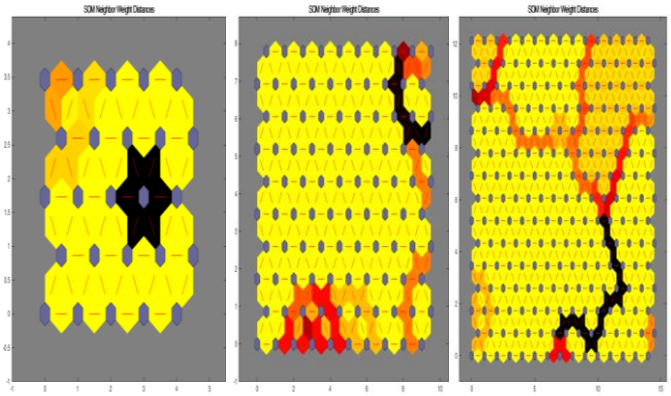

Figure 14: SOM Clustering of data with (left) 25 (mid) 100 and (right) 225 neurons.

From Fig 14, we clearly see that 25 neurons is insufficient for the input data but 100 and 225 and more than sufficient for this data. In both cases with 100 and 225 neurons we see a dark patch in the U-matrix indicating that the network has clustered the data into 2 clusters. This agrees with the results from the K-means.

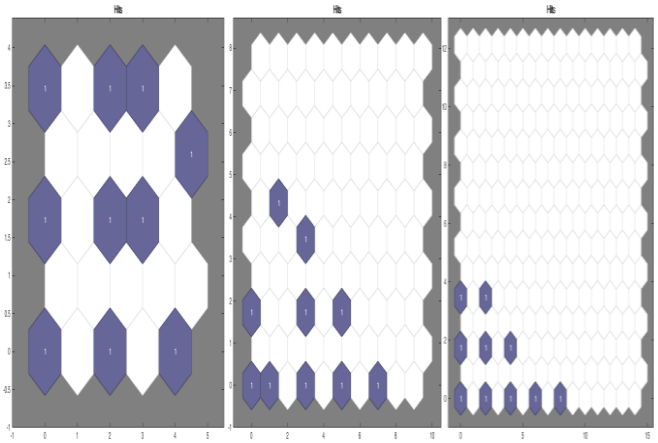

Figure 15 Hits plot for the SOM networks showing each neuron showing the number of input vectors that it classifies. The relative number of vectors for each neuron is shown via the size of a colored patch.

3 Discussion

In this paper we have shown that using open data from the World Bank we are able to derive meaningful patterns in Income Inequality throughout the globe. Even though there is a lot of data available there is big issue with missing values in the time series of the indicators of interest in this study therefore after initial analysis it was deemed that aggregating the data would be necessary.

So after initial data scaling to unit interval the data was aggregated by country and year for the time series from 1980 to 2010. The final data that was used for the analysis was the country aggregated data and the yearly aggregated data is used to present general trends. In the country level data we have calculated a number of attributes based on our initial indicators. Thus the final variables in our dataset were:

- ⊕ Scaled GDP values
- ⊕ Scaled Population values
- ⊕ Scaled II values
- ⊕ 30 Year percentage change in GDP values
- ⊕ 30 Year percentage change in Population values
- ⊕ 30 Year percentage change in II values
- ⊕ II to GDP ratio
- ⊕ II to Population ratio
- ⊕ II to GDP and Population metric

This results in a 214 by 11 matrix for analysis. We make some surprising findings through the application of the Visual Analytics methodology complemented by the custom metrics derived here. We observe that the most severely affected country in our data is Belize and is followed by surprisingly Luxembourg and Iceland in two cases as well as Panama. In the current context this is perhaps less surprising given the role of Panama and Iceland in offshore tax havens but it is interesting to note that our metrics and visual methods are able to allow us to get to this insight which are not possible from static charts. Hence, this represents a good instance of the success of the Visual Analytics methodology.

Initially the expectation was that II would be most observed in countries such as India and China which have undergone rapid economic and societal change over this time period. But taking into account II as a proportion of GDP and Population these countries are not the most affected. Rather we find Russia to have experienced the largest positive change in II over this time period while USA and Canada have experienced large negative changes. This could be due to the fact that 2010 could include the effects of the recession which have biased the figures. Still this is an interesting finding.

As an extension to the maps presented we can see that the data hints at an underlying structure in the data. We see from Fig 6-10 that there a large collection of countries that dominate and the rest where by virtue of missing data or other factors they do not show up as being significant in our analysis. To confirm whether such structure exists we choose the K-Means and SOM clustering techniques based on our literature review.

We find that both the K-Means and SOM confirm the existence of two large clusters. The Silhouette value for K-means shows relatively little change as the number of clusters are increased from 2 to 11 but rather shows a decline in the mean value. The dark patches indicating high distance in the SOM U-matrix also confirms the two clusters in the data.

4 CONCLUSIONS

We show that the Visual Analytics methodology is a very useful methodology in analysis of data with spatio temporal component. We also show that using this methodology it is possible to reach surprising insights not otherwise obvious. We have demonstrated these practically through our analysis.

REFERENCES

[1] P. M. Latha, “A Review on Clustering Techniques 1,” vol. 11, no. 5, pp. 14–17, 2014.

[2] a. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[3] S. Roy and D. K. Bhattacharyya, “Data Clustering Techniques – A Review,” *Rev. Lit. Arts Am.*, no. April, 2016.

[4] S. K. Popat and M. Emmanuel, “Review and Comparative Study of Clustering Techniques,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 1, pp. 805–812, 2014.

[5] A. K. Mann and N. Kaur, “Review Paper on Clustering Techniques,” vol. 13, no. 5, pp. 803–806, 2013.

[6] A. Zoss, “LibGuides: Introduction to Data Visualization: Visualization Types.”

[7] F. Pérez and B. E. Granger, “IPython: a System for Interactive Scientific Computing,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 21–29, May 2007.

[8] W. McKinney, “Data Structures for Statistical Computing in Python,” in *Proceedings of the 9th Python in Science Conference*, 2010, pp. 51–56.

[9] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.

[10] M. Waskom, “Seaborn: statistical data visualization,” 2012. [Online]. Available: http://stanford.edu/~mwaskom/software/seaborn/. [Accessed: 06-Apr-2016].

[11] E. Jones, T. Oliphant, and P. Peterson, “SciPy: Open Source Scientific Tools for Python. Version: 0.16.0.” 2015.

[12] F. Pedregosa and G. Varoquaux, “Scikit-learn: Machine Learning in Python,” *J. Mach. …*, vol. 12, pp. 2825–2830, 2011.

[13] Tableau, “Business Intelligence and Analytics | Tableau Software.” [Online]. Available: http://www.tableau.com/. [Accessed: 06-Apr-2016].