# WORLD BANK COUNTRY DATA ANALYSIS

## Part 2

Date 31-05-2018

Teresa Riedl, University of Cape Town

## A) Introduction

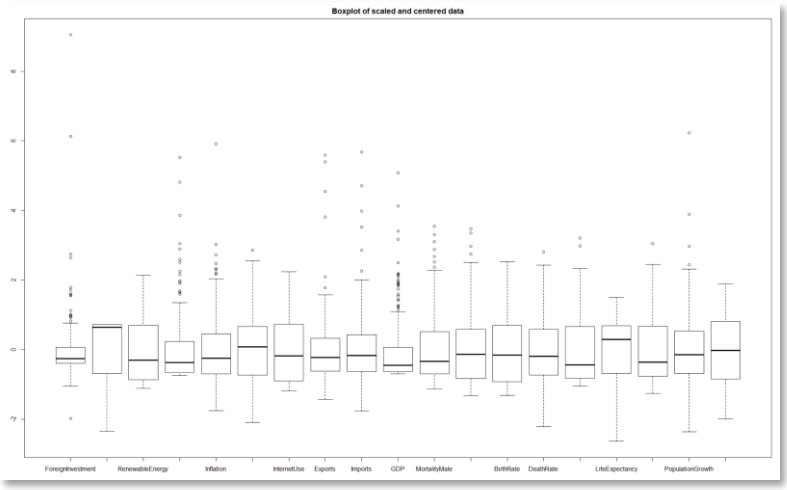
The following report will explore world bank country data. It allows for comparison of countries and answers to the question how different variables such as GDP per capita, energy consumption, mortality rates and various others are linked.

Firstly, it is about understanding the data set, its structure and its gaps. For further processing we identify missing values and either remove the whole row or choose an appropriate approximation as a replacement for the missing value. In order to perform the algorithms, the data needs to be scaled and centred around zero.

## B) Data Exploration

Originally the country data has 192 observations and 19 variables. The first processing to be done is to name the variables with key-words to make sure that we can easily interpret the data at a later stage. Then we identify the fields with missing values which are 63 in total. We remove rows with more than three missing values and use the Country Classification Data from World Bank to estimate the remaining 21 missing values according to their country groups.

We now have the data “original” in a clean format that we can use for the first PCA analysis. But for later use we are looking at the boxplot of the data and observe that the scales of the variables vary heavily. We have to scale the data and centre it around zero for further processing. Scaling means that we divide the column entries by their column standard deviations. We call the data “variables” and boxplot the data to make sure the pre-processing was successful.




#For unsupervised learning methods we use distance as the key feature to distinguish variables and clusters. The distance formula is highly dependent on how variables are measured. In the country data set GDP for instance had a much larger range than all other variables and therefore strongly dominated in the first PCA we pursued on the original data set. So far, we have scaled our data and centred it around zero.

## C) Analysis

For unsupervised learning methods we use distance as the key feature to distinguish variables and clusters. In Part 1 I already outlined why I believe that the correlation based pearson distance would be best for our purposes. I will though process the Multi-dimensional scaling with both Euclidean and Pearson distance to make sure it was the better choice.

## 1. Multidimensional scaling

Multidimensional scaling is a multivariate data analysis approach used to visualize the similarity or dissimilarity between observations by mapping data points into a two-dimensional space. The number of dimensions is pre-specified. It is possible to use a more dimensional space to map the observations, but two dimensions are usually chosen to enhance interpretability. To perform MDS we input the dissimilarity matrix of the data which contains the distances between pairs of objects. As opposed to SOM, MDS only needs the dissimilarities and not the data points themselves. The MDS uses a gradient descent algorithm that seeks to minimize the so-called stress function with $z_1, z_2, \dots, z_N \in \mathbb{R}^k$ and $d_{ij} = ||\, x_i - x_j\,||$ as distance between observation $x_i$ and $x_j$. The following formula shows the stress function for least squares or Kruskal-Shephard scaling.

$$S_M(z_1, z_2, \dots, z_N) = \sum_{i \neq i'} (d_{ii'} - ||z_i - z_{i'}||)^2$$

There are several variations for instance classical scaling where similarities instead of dissimilarities are used as a starting point and Sammon mapping where more emphasis is put on the keeping the smaller pairwise distances. Least square and classical scaling are known as metric scaling methods and Shephard-Kruskal as nonmetric scaling.

Metric Multidimensional Scaling

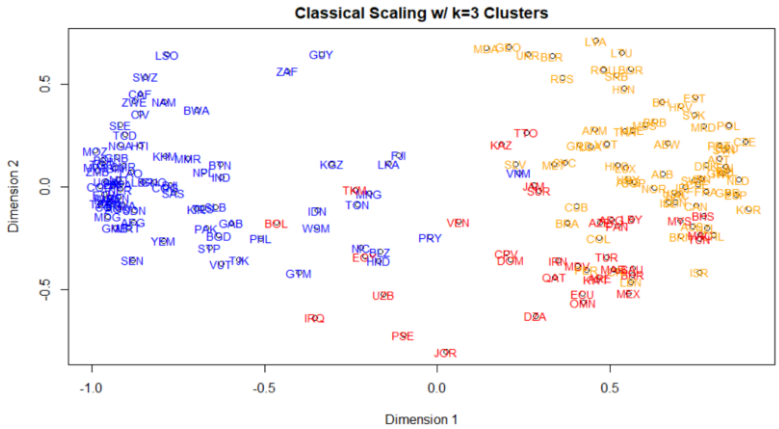
This method also known as principal coordinates analysis is suited for quantitative data. It keeps - as well as possible - the original distance between points. The fitted distances on the MDS map and the original distances are in the same metric.

Non-metric Multidimensional Scaling

Instead of using the distance value non-metric MDS or ordinal MDS uses the value in relation to the distances between other pairs of observations. It is suited for qualitative data and creates fitted distances such that they are in the same rank order as the original distances.

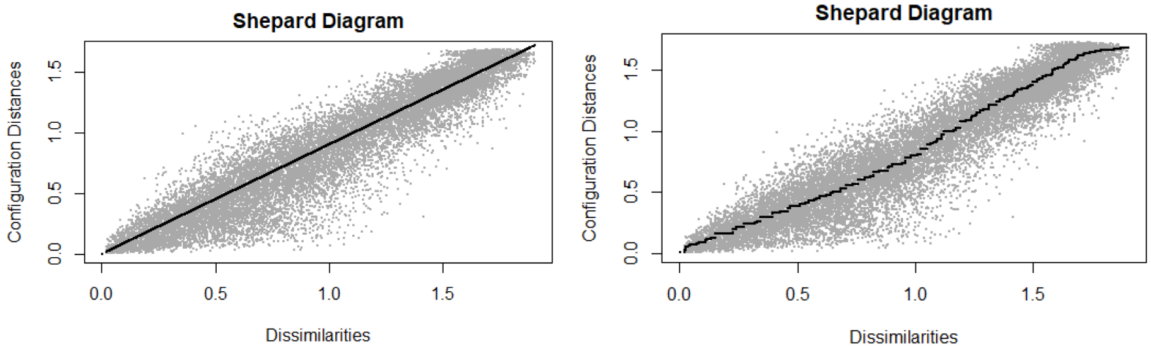
For the countries data set I ran various models both for Euclidean and Pearson distance. As you will see later in the stress value table the Pearson distance returned better results. For the number of clusters, I specified three clusters which was the result from the clustering methods in Part 1 using Pearson distance and ward2 linkage.

Classical Scaling returned the following plot. The cluster seem already fairly evenly distributed. When we look at the countries per cluster we can categorize that the blue cluster is representing underdeveloped countries such as Lesoto or Yemen, the red cluster emerging markets with Mexico and Turkey in it and the yellow cluster developed countries with countries like Korea or Israel. What is outstanding is that outliers like Luxemburg and Hong Kong are included in the clusters and not far off as they were in previous clustering methods which is one benefit of multi-dimensional scaling.

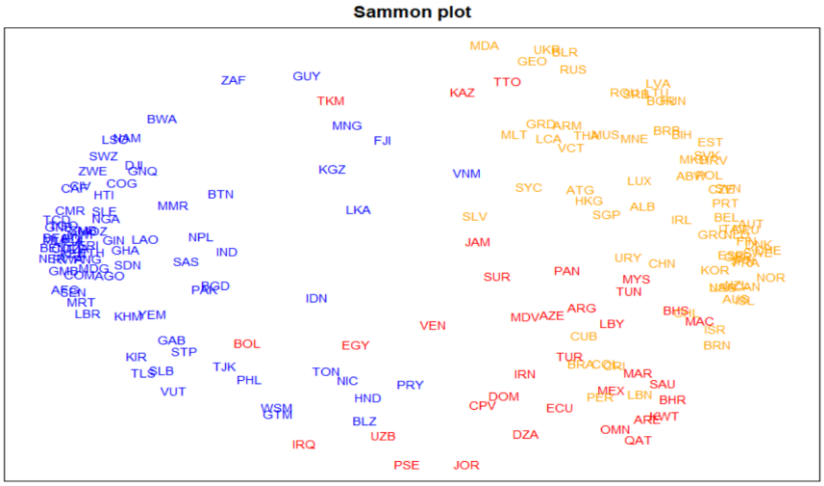



I then used the SMACOF package to compute metric and non-metric scaling with it. For the metric scaling the configuration distances where still quite fat of the middle line as you can see in the following plot of dissimilarities and configuration distance and the stress value lies at 17%. Non-metric SMACOF improved the value only slightly to just over 16%. From the graph we can tell that the non-metric configuration distances curve is bend whereas in the metric one it is linear.

**Shepard Diagrams Metric vs Non-Metric Scaling**




Similarly, Kruskal's Method performed. The algorithm started at a stress value of 17.88% and improved it to 16.86%. I achieved the best value using the Sammon algorithm, the initial stress value is already beating all other methods at a value of 6.2% and it keeps improving up until 70 iterations and then returns a stress value of 4,3%.




The plot above shows the Sammon plot and although the algorithm improved the overall categorizaiton says the same as mentioned in the section for classical scaling. The following table shows the stress values per

**Stress Values**

| Distance Measure & Method | Pearson | Euclidean |
|---|---|---|
| **Classical Scaling** | | |
| **Metric SMACOF** | 17.00% | 17.74% |
| **Non-metric SMACOF** | 16.18% | 11.00% |
| **Sammon** | **4.34%** | 8.02% |
| **Kruskal's** | 16.86% | 11.02% |

Method and distance measure. The pearson distance and Sammon algorithm allow for the best results. Other possible ways to evaluate the models are the goodness of fit, the R-squared values between the actual distances from the two-dimensional maps as well as the eigenvalues. Unlike the stress values those two measured would be the better the higher they are. I calculated the values for the Sammon method and got the following not too promising results:

| | |
|---|---|
| Eigenvalue | 47.19% |
| Goodness of Fit | 61,46% |
| R^2 | • |

## 2. Self-Organizing Maps

We already used several clustering methods in the first part of the report. Self-organizing maps is similar to k-means clustering. Again, the goal is to project the observations into a low-dimensional feature space. The method results in a so-called “constrained topological map”. The algorithm initialized so-called buttons or nodes onto a two-dimensional principal component plane - the gird - before it tries to bend the plane so that the points on the principal component plane approximate the observations. Then from the bended plane, points get projected onto a two-dimensional grid. This process works for one observation at a time by updating the following formula where m are the initialized points, x the observations and $\alpha$ the learning rate.

$$m_k \leftarrow m_k + \alpha(x_i - m_k)$$

As in most of the other methods we use distance measures to define which observations are similar. The algorithm uses a threshold r to determine whether the distance is small or large. With each update the nodes get moved closer to the data whilst maintaining a two-dimensional relationship between them. The learning rate $\alpha$ gets decrease from 1.0 to 0.0 over the iterations, similarly r gets reduced from a pre-defined value R to 1. The following formula shows another term to be updated by the algorithm when a more complex method is chosen.

$$m_k \leftarrow m_k + \alpha h(\left|\left|l_j - l_k\right|\right|)(x - m_k),$$

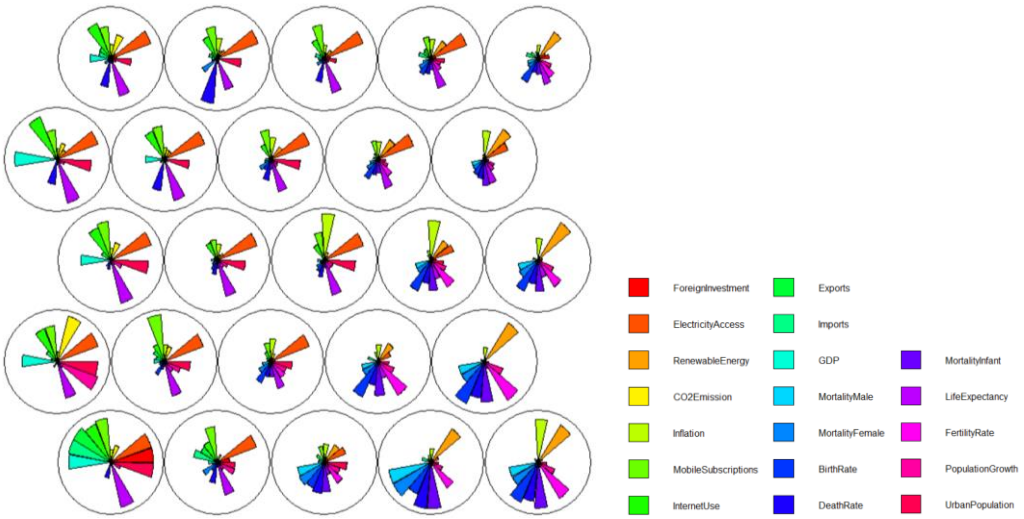
Here $l$ is an integer coordinate pair parametrizing the buttons K and h represents a neighbourhood function that gives more weight to $m_k$ with indices $l_k$ closer to $l_j$ than to those further away.

Before we are implementing the SOM algorithm given the variables matrix, we are initializing the SOM grid. I specified a “hexagonal” grid with dimensions 5x5. Then we input the grid into the SOM model and specify the number of iterations and different values for the learning rate alpha (I choose 0.05 and 0.01 as per the examples in class). One of the key features is that the initiated nodes of the original input data are kept on the map. This means that input samples that are similar in terms of variables are placed close together on the SOM grid.

Codes Diagram

Now there are several different graphical outputs for the SOM model. At first, let's have a look at the codes diagram below. I specified codeRendering="segments" as this seems easier to interpret than the line graph. The legend shows that each variable get's one colour from the rainbow palette which I specified in the code. We can already start thinking about some clusters.

**Codes Plot**



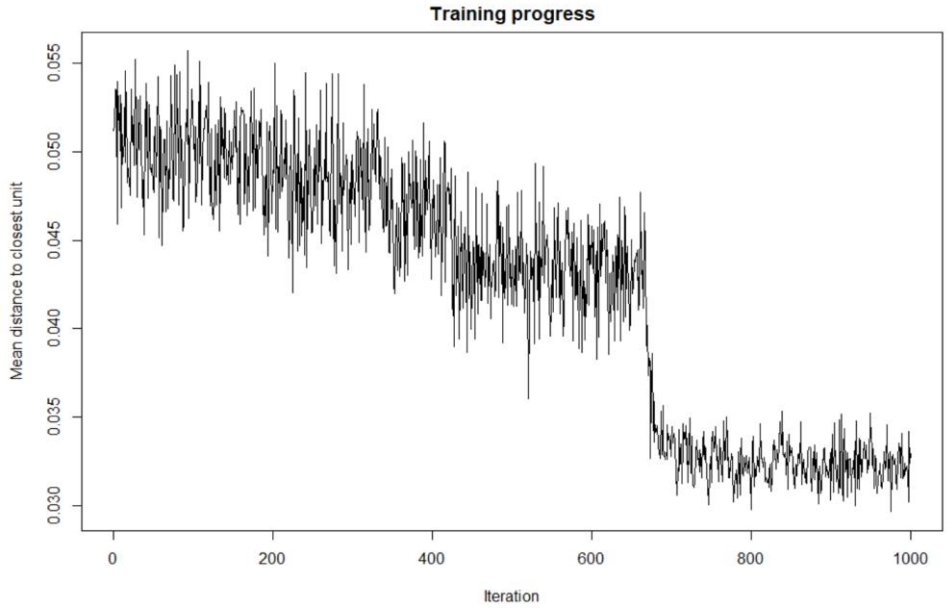

What stands out are the following relationships:

- On the (bottom) right the predominant features are the green, orange and purple segments. Looking at the legend those can be identified as Mobile Subscriptions, Internet Use, Import and Export, Electricity Access and Life Expectancy. We can categorize those nodes as well-developed countries with international trade relationships.
- On the bottom left there is another dominant appearing cluster. The outstanding features are blue, orange, purple and pink and seem to be the inverse of bottom right. The features include all variables related to population growth: Male, female and infant mortality rates as well as fertility, birth and death rates. Moreover, there is an outstandingly high inflation and a high use of renewable energy which in relation with the little developed economic values seems to be due to a lack of infrastructure. These 3-4 bottom left clusters most likely represent developing countries.

We could find other clusters in the plot which would be less dominant than the specified ones.

Training Progress

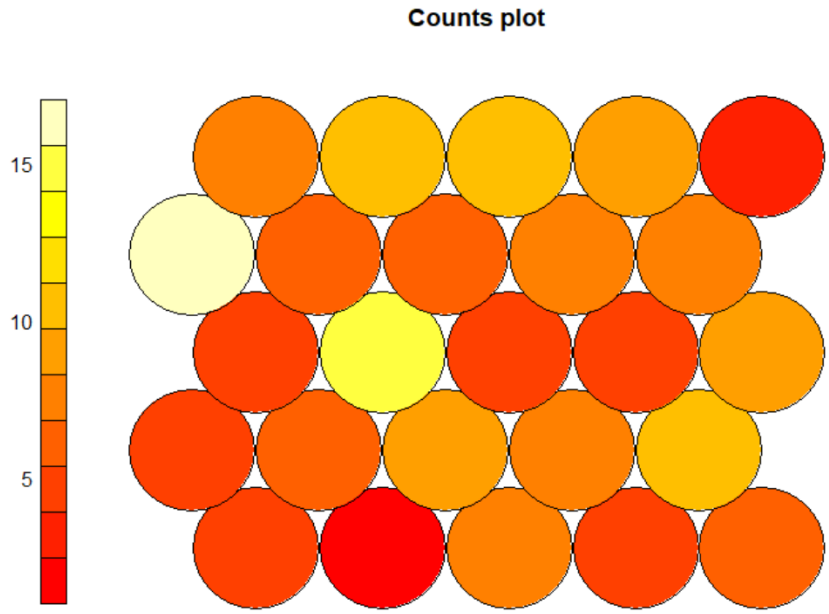
Next, we have a look at the training progress of the SOM model. Again, we can use the plot function to visualize the training progress of the SOM algorithm. The graph below maps out the mean distances to the closest units based on the number of iterations.




The SOM model is perceived to progress when the distance from each node's weights to the samples represented by that node is reduced iteration by iteration. In my model the training progress stagnates at a mean distance to the closest unit between 3% and 3.5%.

Node Counts

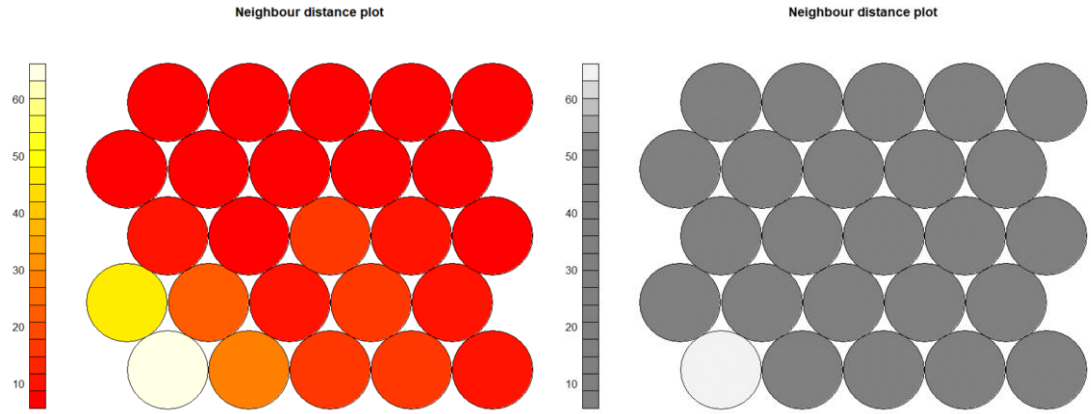
The node counts tell us how many samples are mapped to each node on the map.




Indeed, our node counts plot has a relatively uniform distribution which is what we are hoping to achieve. Except for tow nodes that are standing out with more than 10 observations in them which could indicate that our map is too big, but I tried a smaller map which resulted in several empty nodes.

U-Matrix

The Neighbour Distances as known as the U-Matrix which shows the distance between each node and its neighbours.



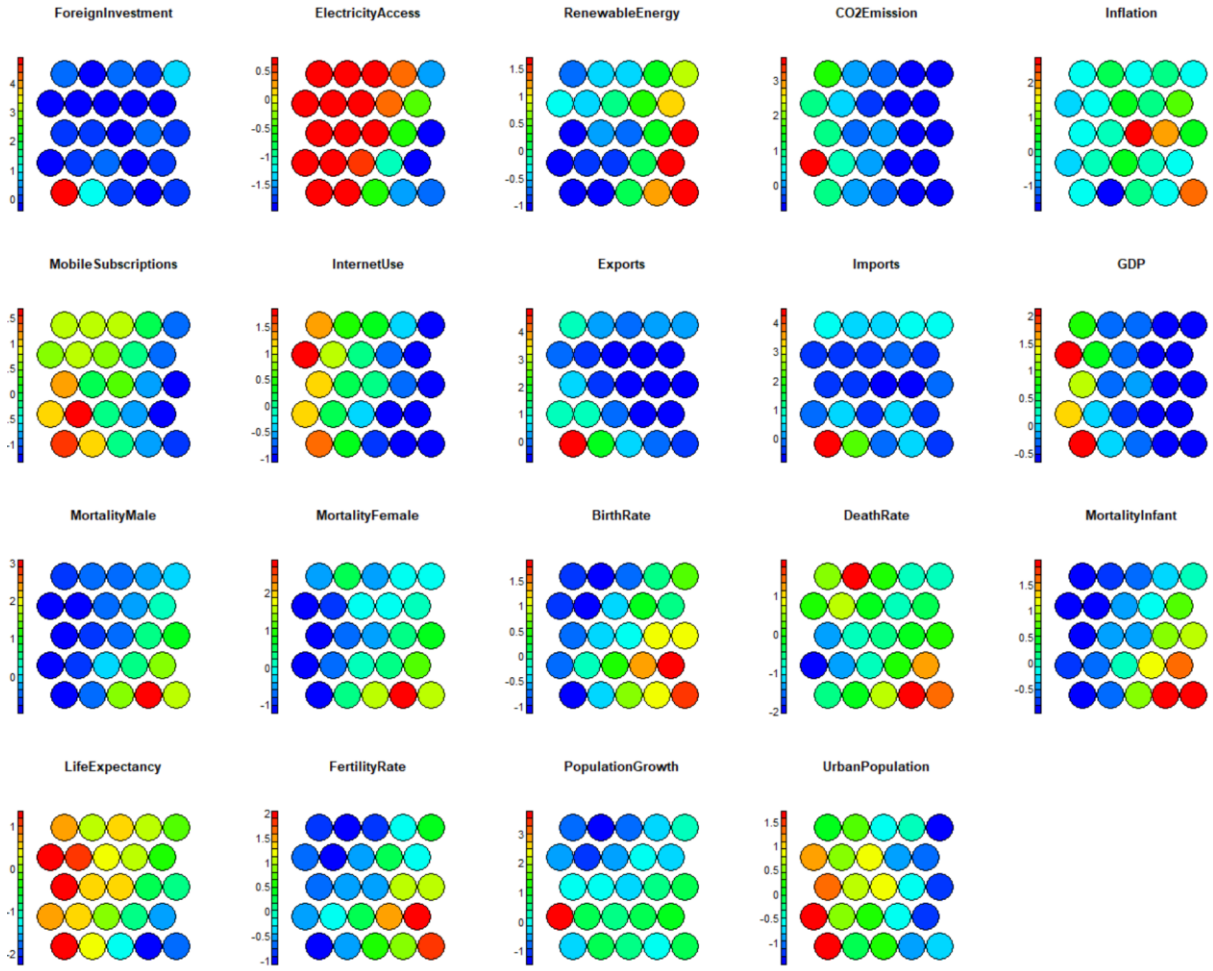

The graph is often viewed as a greyscale palette which helps us to identify clusters. In this case it is very straight forward to say that there would be two clusters.

Heatmap

Above we tried to interpret the codes plot which displayed the weights of the different variables. As we have 19 variables the codes plot is a bit difficult to interpret. The heatmap is supposed to give a better visualization of the distribution of a single variable across the map. I found a function[1] that helped me to quickly plot one heatmap for each variable. The following diagram shows the kohonen property plot for each variable. Note that the observations stay in the same node across all maps whilst the colouring of the heat map represents the dominance of one variable in each node of one map.

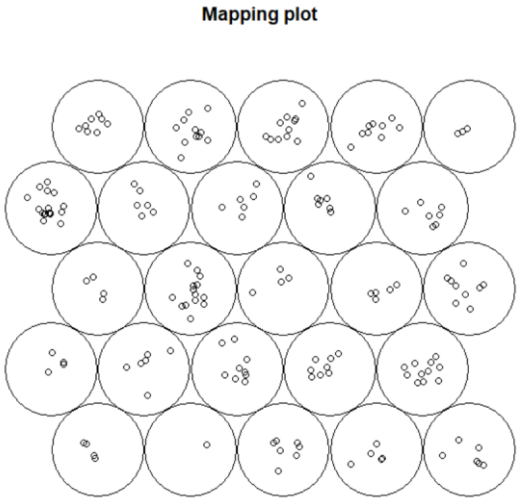
I briefly want to emphasize a few points deriving from the variables property plots. Foreign Investment is very much in line with the U-Matrix and seems to be one predominant factor when it comes to clustering the countries. It becomes obvious that economic values are strongly correlated. Import, Export, GDP and Foreign Investment have nearly the same structure. At the same time, they are correlating with the CO2 Emission variable which shows that economically strong countries have a worse impact on the environment. Another interesting correlation are the Mortality Rates (Female, Male, Infant) and the birth rates whereas Life Expectancy is the inverse of them. We can learn from this that in countries where mortality rates are higher people have more children and have shorter lives. Life Expectancy on the other hand correlates again with other features such as Mobile Subscriptions and Electricity Access.

[1] https://github.com/shanealynn/Kohonen-Self-organising-maps-in-R/blob/master/plotHeatMap.R



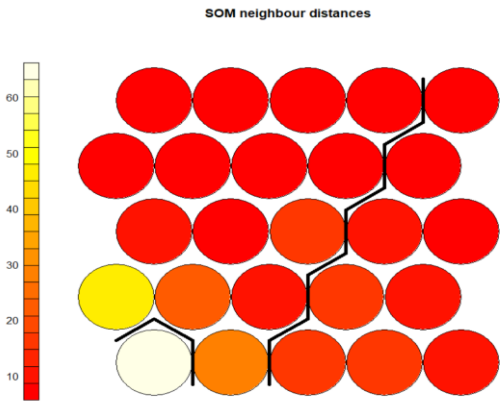

Mapping plot

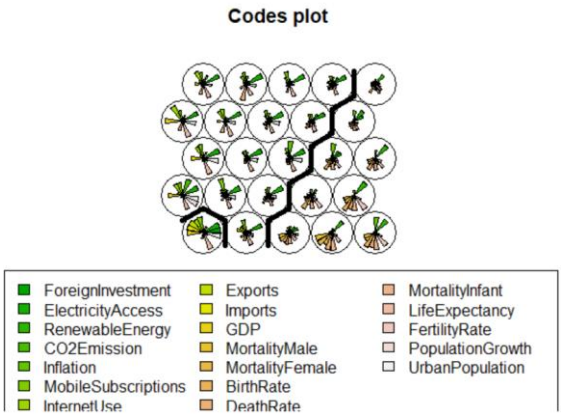
The following mapping plot shows us the actual distribution of country observations in the different nodes. We can see that there are a few quite dense nodes in the top left and a few nodes with only very little observations – one in the middle, one top right and two on the bottom left.




Finally, we want to finalize the clusters of the SOM using the neighbour distances. The following graphs show the three different clusters of the SOM. I must say that I personally prefer the codes plot which helps

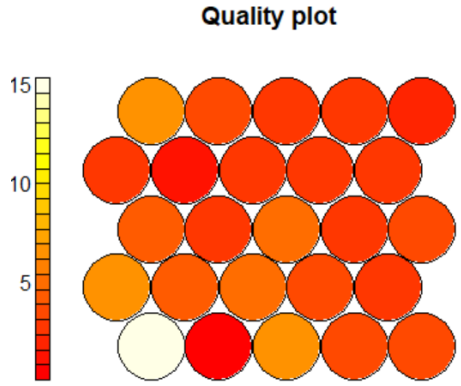
understand the different variables at one glance without looking at 19 different property plots, but certainly this is based on taste and the use case.




Evaluation

For evaluation of Self-organizing maps one can look at the following measures. The quantization error is calculated by taking the mean of the SOM model distances. In our model it is 3.68. The uniform classification measure basically only helps us to find empty cells which are to be avoided. Computing a Quality Plot helps us to understand the cluster sizes, ideally it would all be one colour. In our case we have some outliers like Hong Kong and Luxemburg which make it quite difficult.




## D) Conclusion

We already presented some interpretation work in Part 1 and the overall classification in developed, emerging and underdeveloped countries still apply after the new methods were performed. Nevertheless, the SOM add lots of value when it comes to visualization and interpretation of the variables that we use.

## References

*‘‘Introduction to Statistical Learning’’ by Gareth James, Daniela Witte, Trevor Hastie and Robert Tibshirani*

*‘‘The Elements of Statistical Learning’’ by Trevor Hastie, Robert Tibshirani and Jerome Friedman*

*‘‘Practical Guide To Cluster Analysis in R’’ – Alboukadel Kassambara*

*https://www.shanelynn.ie/self-organising-maps-for-customer-segmentation-using-r/*