Analysis of the effects of climate change on GDP by country.

By: Arnab Suklabaidya

NIT Agartala

Contents

Chapter 1

Abstract

For decades, there has been a growing body of research establishing links between climate change and GDP. In short, climate change impacts the health, agricultural, socioeconomic, and animal husbandry industries of an economy. The relationship between climate change and GDP has been long understood. It is crucial to further our understanding of climate change to anticipate its damages and is also central to policymaking that weighs both costs and benefits of climate change. The models applied in understanding climate change are not accurate, in other words, predictions are either shrouded in uncertainty or not reliable. There have been several predictions and conclusions exploring the links between climate change and GDP. Climate change impacts the poor countries more, as the increasing temperatures affect income growth, productivity, and standard of living. Further, unlike the rich countries, poor countries are ill-equipped to deal with the consequences of climate change.

Chapter 2

Introduction

Climate change is a global and long-term phenomenon, which requires global coordination and a forward looking policy approach. The project's objective was to find Impacts of Global Warming (3◦C) on the World GDP (% Change/Year) by 2027, 2037, 2047, 2067 and long run under the RCP 6.00 Scenarios for all the 25 examined countries. The data obtained from some external resources were explored and analyzed. Furthermore, I checked for stationarity in the data by visualizing the correlations. I performed tests on this data with some statistical tools like Principal component analysis (PCA) to summarize the information content in large data tables by means of a smaller set of "summary indices" that it can be more easily visualized and analyzed , K-Means Clustering to classify observations into **k** groups, based on their similarity , Hierarchical Clustering to draw inferences from unlabeled data . The overall objective was to examine how climate change affects the world GDP in near future.

Chapter 3

Methodology

3.1 Data Collection

The analysis was performed over the data collected form the various source (Ex: IBM) to find Impacts of Global Warming (3∘C) on the World GDP (% Change/Year) under the RCP 6.00 Scenarios. There are some future years as 2027, 2037, 2047, 2067 and years in long run on the data. I considered the data of 25 most growing countries for analysis. The data obtained was in good shape, and no cleaning was required.

3.1.1 Change in GDP data

Year	Australia	China	India	USA	Japan
2027	-0.051	-0.205	−1.023	−0.015	−0.042
2037	-0.107	-0.438	−2.099	−0.037	−0.100
2047	-0.172	−0.692	−3.222	−0.067	−0.173
2067	-0.326	−1.247	−5.532	−0.147	−0.356

3.2 Data Exploration

Basically here the impacts of Global Warming (3∘C) on the World GDP under the RCP 6.00 Scenarios have measured on the rate of change of GDP per year. To perform data exploration, I chose twenty-five countries out of which some are developed , some are developing and some under-developing countries. I take five future years (2027, 2037, 2047, 2067, 2087). Behalf of these years the future GDP data was calculated of those countries.

3.2.1 Data Summarization

First I loaded pacman and rio for data import and stored the loaded data onto **mydata** object . I used **str()** function for compactly displaying the internal structure of the dataset. Then I loaded the data of the respective future years onto respective objects for data summarization.

I calculated the mean , median , mode, standard deviation, varience and IQR by using the respective functions mean(), median(), Mode(), sd(), var(), quantile().

Year	Mean	Median	Mode	SD	Var
2027	-0.193	-0.033	-0.357	0.346	−0.042
2037	-0.419	-0.085	-0.829	0.730	−0.100
2047	-0.674	-0.154	-1.387	1.138	−0.173
2067	-1.246	-0.319	-2.674	1.991	−0.356
2087	-3.247	-1.209	-7.304	3.874	15.00

```r
#mode Function
Mode= function(x){
  ta = table(x)
  tam = max(ta)
  if(all(ta==tam))
    mod= NA
  else
    if(is.numeric(x))
      mod = as.numeric(names(ta)[ta==tam])
  else
    mod=names(ta)[ta==tam]
  return(mod)
  }

```

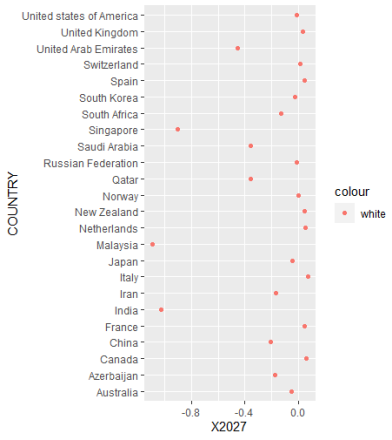
Figure 3.1: An algorithm for the determination of the Mode function

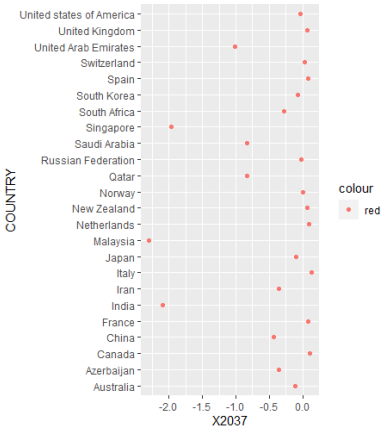
3.2.2 Data Visualization

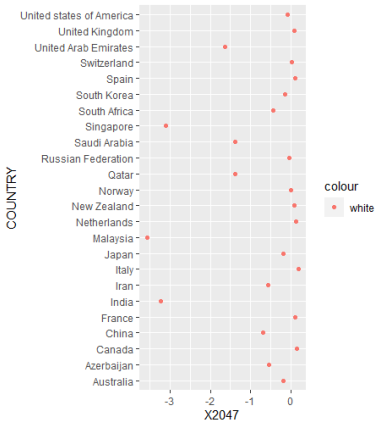
First I started by loading the required packages . ggplot2 package is required to plot the data obtained from the dataset. Installing the tidyverse package as its include ggplot2. **ggplot2** is a plotting package that makes it simple to create complex plots from data in a data frame. It provides a more programmatic interface for specifying what variables to plot, how they are displayed, and general visual properties.

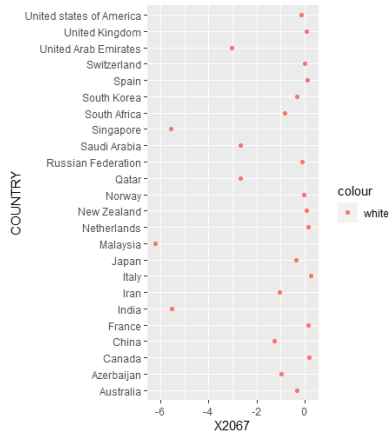
I plotted the data of the all respective years with respect to all the 25 countries given in the data set using this code:

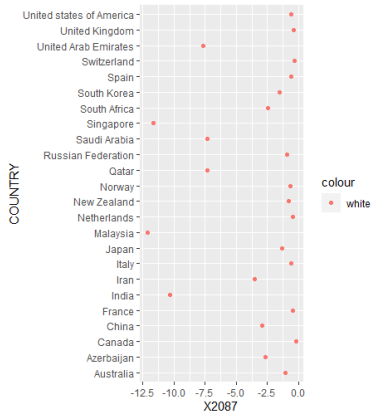
```r
ggplot(data = mydata) +
geom_point(mapping = aes(x= YEAR , y=COUNTRY,
                col = "white"))
```


United states of America
United Kingdom
United Arab Emirates
Switzerland
Spain
South Korea
South Africa
Singapore
Saudi Arabia
Russian Federation
Qatar
Norway
New Zealand
Netherlands
Malaysia
Japan
Italy
Iran
India
France
China
Canada
Azerbaijan
Australia
COUNTRY
-0.8
-0.4
0.0
X2027
colour
white

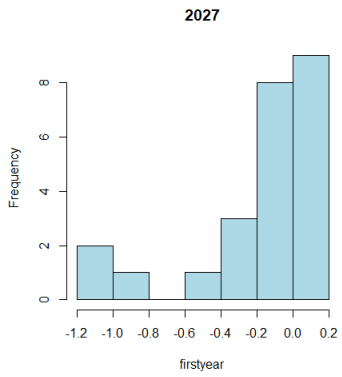

United states of America
United Kingdom
United Arab Emirates
Switzerland
Spain
South Korea
South Africa
Singapore
Saudi Arabia
Russian Federation
Qatar
Norway
New Zealand
Netherlands
Malaysia
Japan
Italy
Iran
India
France
China
Canada
Azerbaijan
Australia
COUNTRY
-2.0
-1.5
-1.0
-0.5
0.0
X2037
colour
red

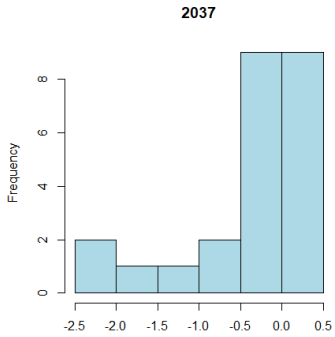

United states of America
United Kingdom
United Arab Emirates
Switzerland
Spain
South Korea
South Africa
Singapore
Saudi Arabia
Russian Federation
Qatar
Norway
New Zealand
Netherlands
Malaysia
Japan
Italy
Iran
India
France
China
Canada
Azerbaijan
Australia
COUNTRY
-3
-2
-1
0
X2047
colour
white

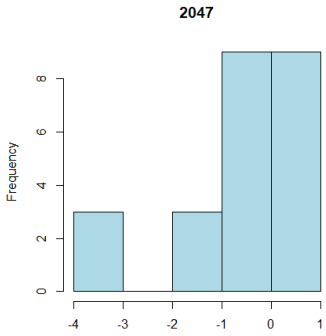

United states of America
United Kingdom
United Arab Emirates
Switzerland
Spain
South Korea
South Africa
Singapore
Saudi Arabia
Russian Federation
Qatar
Norway
New Zealand
Netherlands
Malaysia
Japan
Italy
Iran
India
France
China
Canada
Azerbaijan
Australia
COUNTRY
-6
-4
-2
0
X2067
colour
white

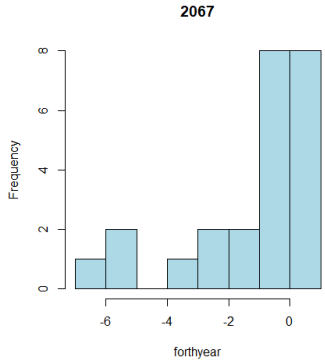

United states of America
United Kingdom
United Arab Emirates
Switzerland
Spain
South Korea
South Africa
Singapore
Saudi Arabia
Russian Federation
Qatar
Norway
New Zealand
Netherlands
Malaysia
Japan
Italy
Iran
India
France
China
Canada
Azerbaijan
Australia
COUNTRY
-12.5
-10.0
-7.5
-5.0
-2.5
0.0
X2087
colour
white

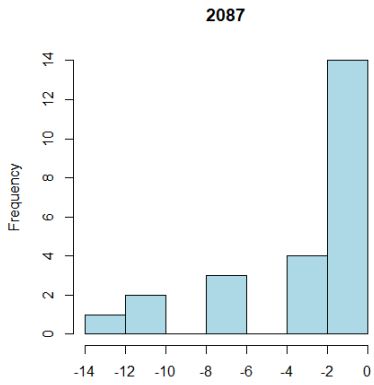
Secondly I plotted the numeric data of the dataset creating histogram using **hist()** function. This function takes a vector as an input and uses some more parameters to plot histograms.





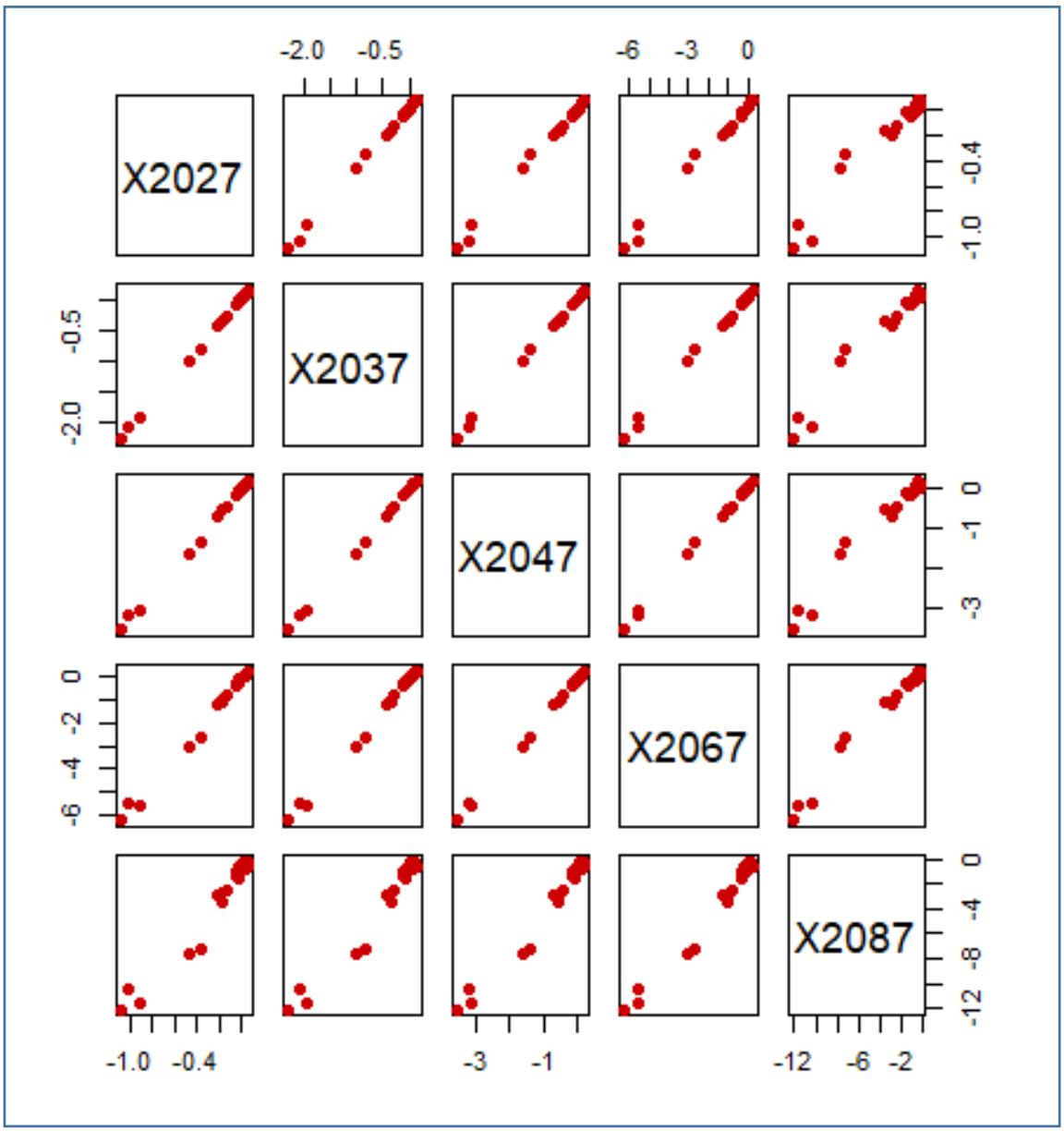






3.2.3 Data Normalization

Data Normalization is a **data** preprocessing step where we adjust the scales of the features to have a standard scale of measure. Here I took the numeric variables in the dataset and store it in **mydata_numeric** object and after that I plotted the data using **plot()** function.



3.3 Data Analysis

3.3.1 Principal component analysis (PCA)

Principal Component Analysis (PCA) is a useful technique for exploratory data analysis, allowing you to better visualize the variation present in a dataset with many variables. It is particularly helpful in the case of "wide" datasets, where you have many variables for each sample. In this tutorial, you'll discover PCA in R.

The first step in defining the principal components of p original variables is to find a linear function a'1y, where a1 is a vector of p constants, for the observation vectors that have maximum variance. This linear function is defined as:

$$a_1'y = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1p}x_p = \sum_{j=1}^{p} a_{1j}x_j$$

Principal component analysis continues to find a linear function a'2ya2'y that is uncorrelated with a'1y with maximized variance and so on up to k principal components.

In R the functions like **ggplot(), ggplot2(), prcomp()** are used from the packages stats, ggfortify, ggplot2 etc to do the principle component analysis of the given dataset. Here the given code I used for storing the numeric data into PCA object and plotting the ggfortify way and the ggplot2 way both in colored way.

```
1.   # creating the PCA object using  mydata set
2.
3.   mydata_numeric.pca <- mydata_numeric[c( 2, 3, 4)]
4.   pca.obj <- prcomp(mydata_numeric.pca)
5.
6.   # ggfortify way - w coloring
7.   library(ggfortify)
8.   autoplot(pca.obj) + theme_minimal()
9.
10.
11.  # ggplot2 way - w coloring
12.  library(ggplot2)
13.
14.  # the first two componets are selected
15.  dtp <- data.frame('GDP' = mydata$X2027, pca.obj$x[,1:2])
16.  ggplot(data = dtp) +
17.  geom_point(aes(x = PC1, y = PC2, col = GDP)) +
18.  theme_minimal()
19.
```

3.3.2 K-means Clustering

K-means clustering is one of the most commonly used unsupervised machine learning algorithm for partitioning a given data set into a set of k groups , where k represents the number of groups pre-specified by the analyst.

The basic idea behind k-means clustering consists of defining clusters so that the total intra-cluster variation is minimized.

There are several k-means algorithms available. The standard algorithm is the Hartigan-Wong algorithm , which defines the total within-cluster variation as the sum of squared distances Euclidean distances between items and the corresponding centroid:

$$\sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2$$

I used WSS plot function to plot the points of the numeric data and to choose number of clusters . Then I stored the kmeans() function to the KM object and used the autoplot() function for clustering the plot. At the last I used KM$centers for the determination of the cluster centers.

```
#....K means clustering.....#

#WSS Plot function
wssplot <- function(data, nc=15, seed=1234)
{
wss <- (nrow(data)-1)*sum(apply(data, 2, var))
for (i in 2:nc){
set.seed(seed)
wss[i] <- sum(kmeans(data, centers = i)$withinss)}
plot(1:nc, wss, type = "b", xlab = "Number of Clusters",
ylab = "Within groups sum of squares")
}

# WSS plot to choose maximum number of clusters
wssplot(mydata_numeric)


#kmeans clustering
KM = kmeans(mydata_numeric, 2)

#cluster Plot
autoplot(KM, mydata_numeric, frame= TRUE)

#cluster Centers
KM$centers

```

3.3.3 Hierarchical Clustering

Hierarchical clustering is an Unsupervised non-linear algorithm in which clusters are created such that they have a hierarchy(or a pre-determined ordering). For example, consider a family of up to three generations. A grandfather and mother have their children that become father and mother of their children.

First I used the Euclidean Method for storing the distance matrix in "d" object. The **distance()** function is implemented using the same *logic* as R's base functions **stats::dist()** and takes a matrix or **data.frame** as input. The corresponding matrix or data.frame should store probability density functions (as rows) for which distance computations should be performed.

Then I used "Ward" method the use the **hclust()** function and store it in fit object. Basically the **Ward's method** aims to minimize the total within-cluster variance. At each step the pair of clusters with minimum between-cluster distance are merged.

Then the plot the fit object to display the dendogram. After that used **cuttree()** function to cut the tree into 5 clusters and at the end used **rect.hclust()** function to draw the dendogram with red borders around the 5 clusters.

```
1.   # Hierarchical Clustering
2.
3.   # distance matrix
4.   d <- dist(mydata, method = "euclidean")
5.
6.   fit <- hclust(d, method="ward")
7.
8.   # display dendogram
9.   plot(fit)
10.
11. # cut tree into 5 clusters
12. groups <- cutree(fit, k=5)
13.
14. # draw dendogram with red borders around the 5 clusters
15. rect.hclust(fit, k=5, border="red")
16.
```

Chapter 4

Results

4.1 Principal component analysis (PCA)

Theoretically, PCA is a method of creating new variables (known as principal components, PCs), which are linear composites of the original variables. The values of PCs created by PCA are known as principal component scores (PCS). The maximum number of new variables is equivalent to the number of original variables.

PCA gives new indicators which are linear combinations of the original ones, thus the new indicators combines similar old indicators through their shared properties, you are going to redefine these new indicators according to your understanding of the potential shared properties. You are also going to choose a proper number of new indicators according to how much information is interpreted by these new indicators. Through the process, the number of indicators is reduced.

The VFs values which are greater than 0.75 (> 0.75) is considered as “strong”, the values range from 0.50-0.75 (0.50 ≥ factor loading ≥ 0.75) is considered as “moderate”, and the values range from 0.30-0.49 (0.30 ≥ factor loading ≥ 0.49) is considered as “weak” factor loadings.

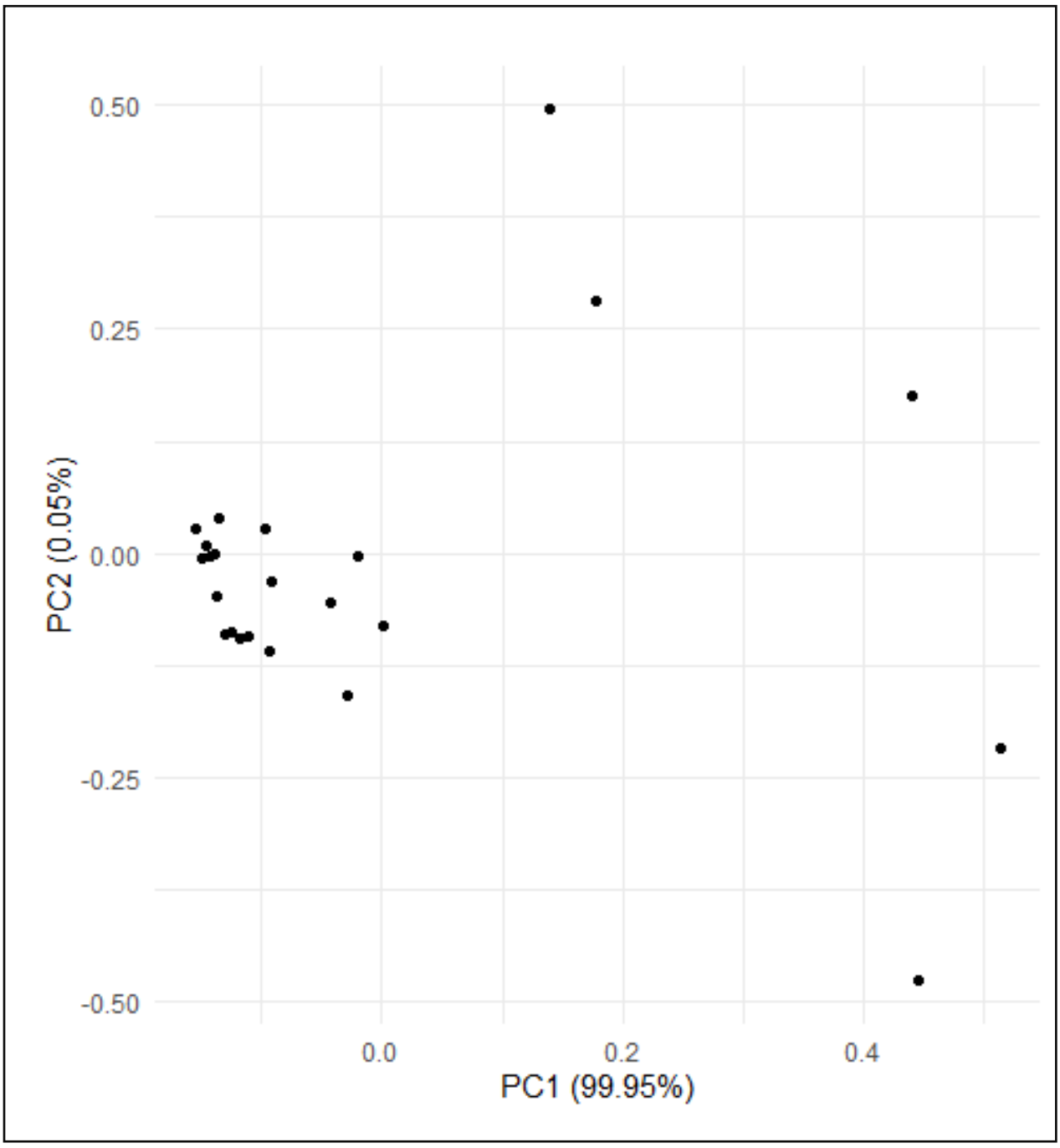


Fig: ggfortify way - w coloring

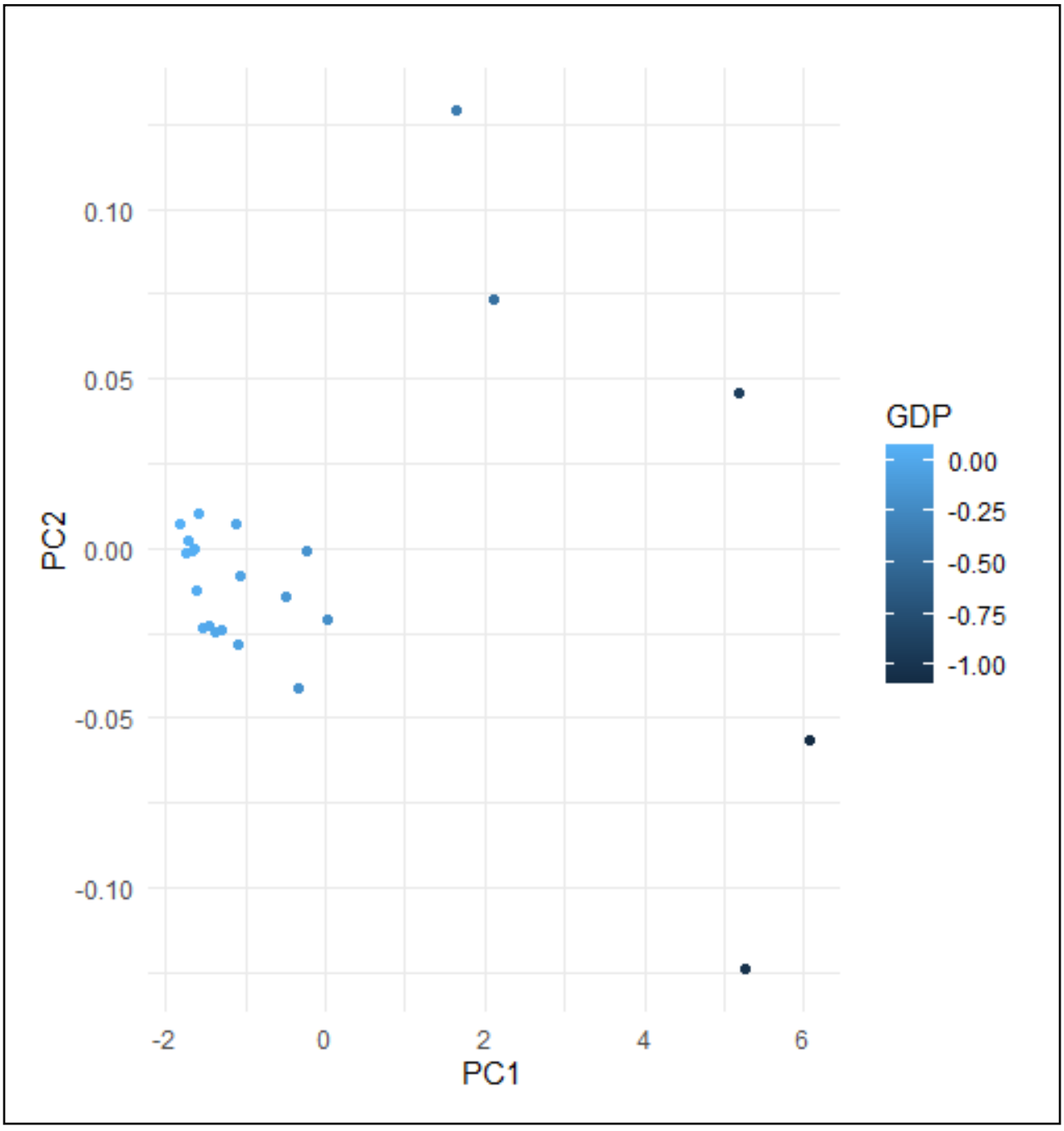


Fig: ggplot2 way - w coloring

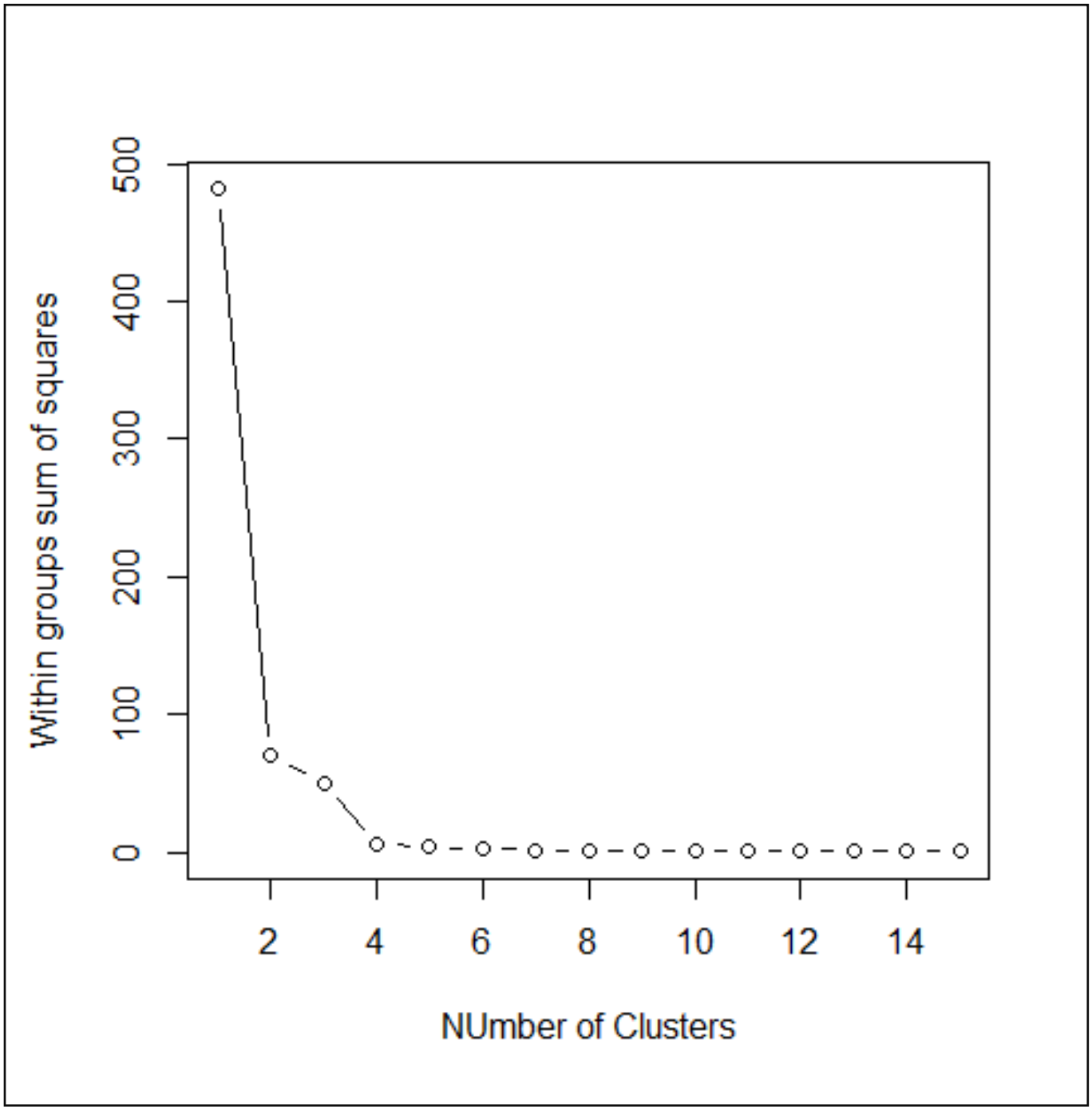
4.2 K-means Clustering

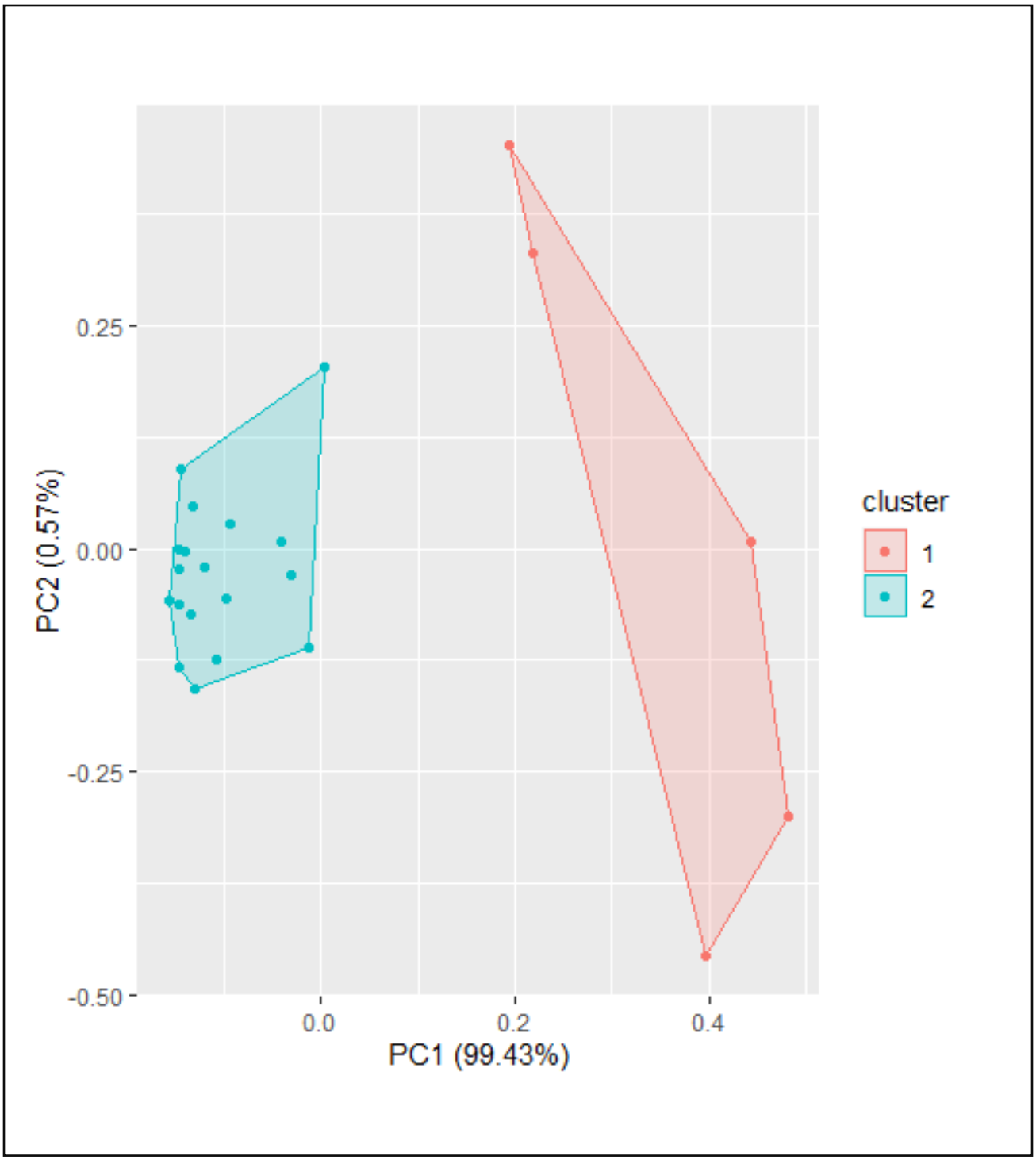
The **kmeans()** function outputs the results of the clustering. We can see the centroid vectors (cluster means), the group in which each observation was allocated (clustering vector) and a percentage (89.9%) that represents the **compactness** of the clustering, that is, how similar are the members within the same group. If all the observations within a group were in the same exact point in the n-dimensional space, then we would achieve 100% of compactness.

Since we know that, we will use that percentage to help us decide our **K** value, that is, a number of groups that will have satisfactory variance and compactness.

Each group is represented by the mean value of points in the group, known as the **cluster** centroid. K-means algorithm requires users to specify the number of **cluster** to generate. The **R** function kmeans() [stats package] can be used to compute k-means algorithm.

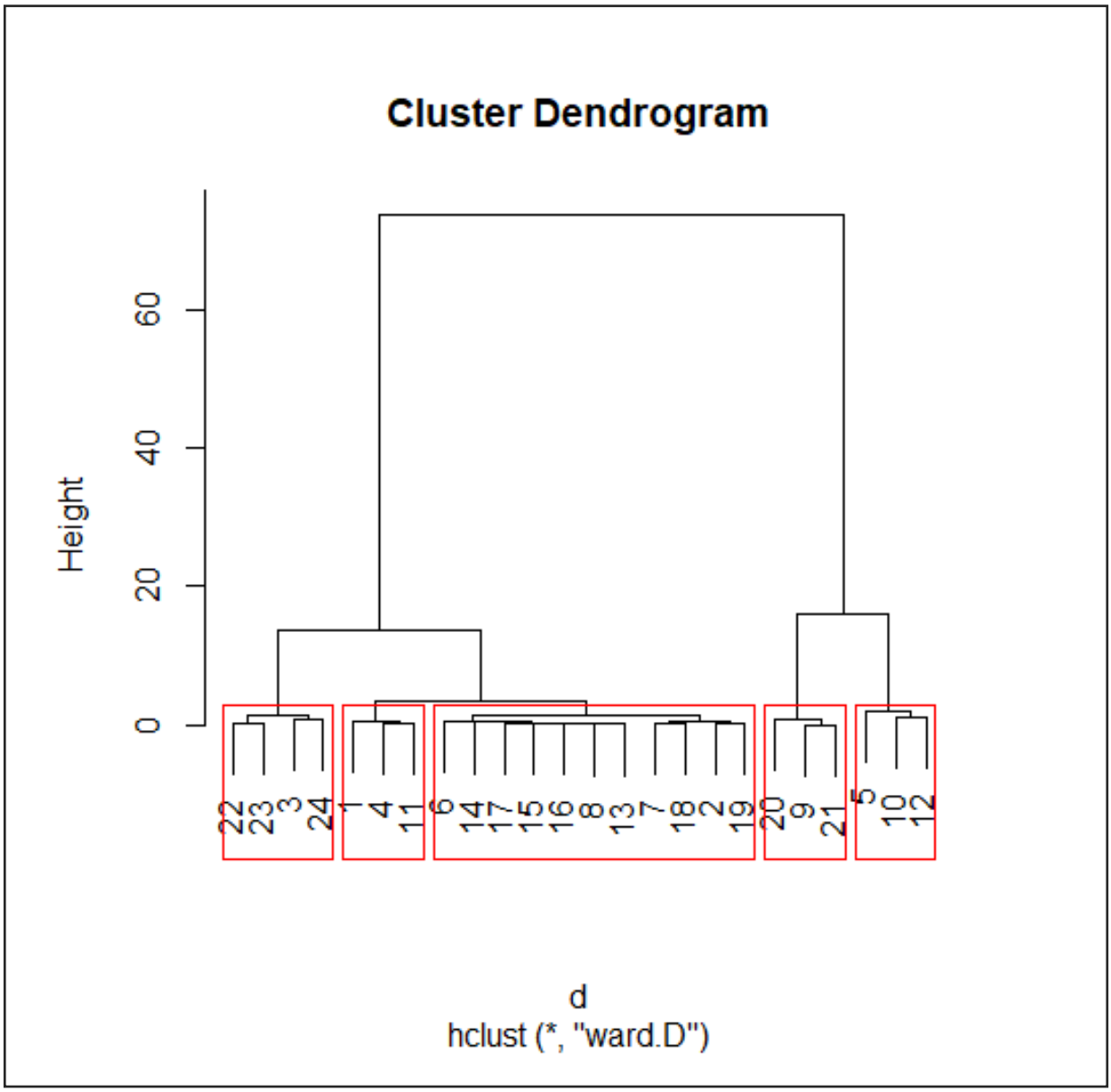
```
> KM$centers
       X2027       X2037      X2047      X2067     X2087
1 -0.69833333 -1.50250000 -2.3833333 -4.2825000 -9.402167
2 -0.02477778 -0.05894444 -0.1054444 -0.2350556 -1.196333
```


Within groups sum of squares
0
100
200
300
400
500
2
4
6
8
10
12
14
NUmber of Clusters


0.25
0.00
-0.25
-0.50
PC2 (0.57%)
0.0
0.2
0.4
PC1 (99.43%)
cluster
1
2

4.3 Hierarchical Clustering

In **hierarchical clustering**, I categorized the objects into a **hierarchy** similar to a tree-like diagram which is called a dendrogram. The distance of split or merge (called height) is shown on the y-axis of the dendrogram below. ... After that 5 was merged in the same **cluster** 1 followed by 3 resulting in two **clusters**.



Chapter 5

Conclusion

Global warming consequences are a significant threat to the Earth's future. Assessing climate change impacts to the global economy and national incomes, and the potential benefit of climate change agreements, however, is complex, requiring large-scale modeling to even approach a comprehensive answer. Since the residuals of the k-means clustering have fitted in this model very will so that we can visualize the difference between the change of growth of GDP in the future years. I conclude from the results that there are some countries of which GDP will going to change drastically with the change of climate in future years. The GDP analysis project can help various economists, statisticians and financial organizations as a reference when applying models for GDP forecasting. Scope of further research can be in refining the forecasting methods to calculate the change of GDP due to climate change with more aquaracy.

References

[1] IMF, "World economic outlook (april 2020)." https://www.imf.org/external/datamapper/datasets/WEO

[2] Earth's Future (July, 2018). https://www.researchgate.net/publication/326382587_The_Effects_of_Climate_Change_on_GDP_by_Country_and_the_Global_Economic_Gains_From_Complying_With_the_Paris_Climate_Accord

[3] IMF, "World economic outlook." https://www.imf.org/external/

[4] A. Kassambara, "ggpubr:"ggplot2" based publication ready plots," R package version 0.1, vol. 7, 2018.

[5] H. Wickham, R. Francois, L. Henry, K. Müller, et al., "dplyr: A grammar of data manipulation," R package version 0.4, vol. 3, 2015.

[6] Modern R with the tidyverse, Bruno Rodrigues ,11,2020. https://b-rodrigues.github.io/modern_R/