# CLASSIFICATION BASED PREDICTION OF CARBON DIOXIDE EMISSION

*Project submitted to University of Madras*

*in partial fulfillment of the requirement*

*for the award of the degree on*

MASTER OF SCIENCE

IN

STATISTICS

By

MD JAFFAR MOLLAH (32818018)

MOHAMMED RAFFIQUE K (32818021)

SATHYA NARAYANAN R (32818023)

Under the guidance of

Dr.M.R.SRINIVASAN

Guest Faculty


DEPARTMENT OF STATISTICS

UNIVERSITY OF MADRAS

CHENNAI - 600 005

SEPTEMBER 2020


# CERTIFICATE

This is to certify that the project entitled **"CLASSIFICATION BASED PREDICTION OF CARBON DIOXIDE EMISSION"** submitted in partial fulfillment of the requirement for the Award of Degree in Master of Science in Statistics. This is a Bonafide record of work done under my guidance and supervision by **MD JAFFAR MOLLAH** (Reg. No. **32818018**), **K MOHAMMED RAFFIQUE** (Reg. No. **32818021**), **R SATHYA NARAYANAN** (Reg. No. **32818023**), in the Department of Statistics, University of Madras, Chennai-600 005 during the Academic year 2018-2020.

**Dr. M.R. SINDHUMOL**
Associate Professor and Head(i/c)
Department of Statistics
University of Madras
Chennai-600 005

**Dr.M.R.SRINIVASAN**
Guest Faculty
Former Head of the Department
Department of Statistics
University of Madras

**PLACE: Chennai**

**DATE: 04-09-2020**

## ACKNOWLEDGEMENT

We should thank our parents for guiding us towards goodness and supporting us to study till masters.

A ship without captain can hardly reach the destination. Thanks to our ship's captain, our guide and beloved professor **Dr.M.R.Srinivasan** for his interest, deep involvement, valuable guidance and encouragement at every stage of our project. We are grateful to him for expending his precious time in critical evaluation and completion of project in prescribed time.

We are grateful to our Head of the Department **Dr.M.R.Sindhumol** for liberally permitting us to choose our project topic. We thank our professors **Dr.M.Ramadurai** and **Dr.S.Suresh** for teaching us Statistics and for their support. We wish to thank retired professor **Dr.S.Sampath** for teaching us data mining techniques which we have made use in this project.

We express our sincere thanks to my fellow classmates and research scholars of the department for their support during our project. Last but not the least we thank our Administrative staffs for helping us throughout our course in various ways.

**MD JAFFAR MOLLAH**

**K MOHAMMED RAFFIQUE**

**R SATHYANARAYANAN**

# Contents

**Preface** **iv**

**1 Introduction** **1**

1.1 Motivation . . . . . 1

1.2 Nature of Data . . . . . 3

1.3 Longitudinal Data . . . . . 3

1.4 Advantages and Disadvantages of Longitudinal Data . . . . . 7

1.5 Outline of the working . . . . . 8

**2 Modeling** **9**

2.1 Introduction . . . . . 9

2.2 Summarizing and Visualizing Longitudinal Data . . . . . 9

2.3 Variable Selection and model building . . . . . 10

2.4 Linear Models for Longitudinal Continuous Data . . . . . 11

2.5 Generalized Linear Models for Longitudinal Data . . . . . 15

2.6 Robust Generalized Linear Models for Longitudinal Data . . . . . 22

**3 Cluster Analysis** **25**

3.1 Introduction . . . . . 25

3.2 Distance Metrics . . . . . 26

3.3 Partitioning Around Medoids (k-medoids) . . . . . 28

3.4 Average Silhouette Width and Elbow Method . . . . . 30

3.5 Clustering of Longitudinal Data . . . . . 31

**4 Classification** **33**

4.1 Introduction . . . . . 33

4.2 Support Vector Machines for Classification . . . . . 34
4.3 Decision Tree Induction for Classification . . . . . 36
4.4 Building Classifier . . . . . 40

**5 Validation and Model Adequacy Checking 41**
5.1 Introduction . . . . . 41
5.2 Model Adequacy Checking . . . . . 42
5.3 Validity Measures . . . . . 43
5.4 Validation for Regression Model . . . . . 44
5.5 Validation for Classification model . . . . . 47

**6 Real Life Dataset and Analysis 49**
6.1 Introduction . . . . . 49
6.2 Variable Description . . . . . 49
6.3 Methodology . . . . . 53
6.4 Visualization . . . . . 55
6.5 Analysis and Results . . . . . 62

**7 R Code 68**

**8 Conclusion 87**

**REFERENCE 89**

# Preface

*Taken as a whole, the range of published evidence indicates that the net damage costs of climate change are likely to be significant and to increase over time.*

-Intergovernmental Panel on Climate Change

Most projections of climate change presume that future changes-greenhouse gas emissions, temperature increases and effects such as sea level rise-will happen incrementally. A given amount of emission will lead to a given amount of temperature increase that will lead to given amount of smooth incremental sea level rise. However, the geological record for the climate reflects instances where a relatively small amount of change in one element of climate led to abrupt changes in the system as a whole. In other words, pushing global temperatures past certain thresholds could trigger abrupt, unpredictable and potentially irreversible changes that have massively disruptive and large-scale impacts. In this project, we deal with one of the aspects called carbon emissions. The study is based on finding groups of countries with same features and then forecasting the carbon emission using longitudinal data and then looking at the transition of status based on categories of the carbon emissions.

- **Chapter 1**: This chapter explains reasons for using longitudinal data in our study. Also the outline of the work has been presented using flowchart.
- **Chapter 2**: This chapter explains the various aspects of modeling like summarization, visualization, variable selection and the different models used in our study. The formulas and procedure used in modeling has also been clearly presented.
- **Chapter 3**: This chapter deals with defining the clustering technique used in our study. A new distance metric specially for longitudinal data has been defined. Also, the way to select optimal number of clusters has also been presented.
- **Chapter 4**: This chapter briefly describes the procedure and algorithm used for classification. However, the mathematical part of the techniques has been avoided purposely keeping in mind the objective of our study.

- **Chapter 5**: A model can only be accepted if it gives consistent results and this has been shown through the validation technique specially designed for longitudinal data. Even validity measures related to modeling and classification has been briefly presented which can be used in model comparisons.

- **Chapter 6**: A real life dataset was prepared from secondary sources to show the procedures and the results obtained as explained in the previous chapters.

- **Chapter 7**: The R code used in the study is given in the chapter

# Chapter 1

# Introduction

## 1.1 Motivation

Climate Change is the defining issue of our time and we are at a defining moment. From shifting weather patterns that threaten food production, to rising sea levels that increase the risk of catastrophic flooding, the impacts of climate change are global in scope and unprecedented in scale. Without drastic action today, adapting to these impacts in the future will be more difficult and costly.

Greenhouse gases occur naturally and are essential for the survival of humans and millions of other living things, by keeping some of the sun's warmth from reflecting back into space and making Earth livable. But after more than a century and a half of industrialization, deforestation, and large scale agriculture, quantities of greenhouse gases in the atmosphere have risen to record levels not seen in three million years. As populations, economies and standards of living grow, so does the cumulative level of greenhouse gas (GHGs) emissions. There are some basic well-established scientific links:

- The concentration of GHGs in the earth's atmosphere is directly linked to the average global temperature on Earth;
- The concentration has been rising steadily, and mean global temperatures along with it, since the time of the Industrial Revolution;
- The most abundant GHG, accounting for about two-thirds of GHGs, carbon dioxide ($CO_2$), is largely the product of burning fossil fuels.

Corresponding to the record-breaking global emissions of the last years, the carbon dioxide ($CO_2$) concentration in our atmosphere already exceeds the historical value of

400ppm. If this trend is not inverted, our chances to keep global warming well below $2^oC$ and to pursue efforts to limit the increase to $1.5^oC$ thus avoid climate change with all its expected impacts are virtually zero. The special report on **Global Warming of $1.5^oC$**, newly released by the International Panel on Climate Change (IPCC), sheds light on the substantial difference in impacts between warming of $1.5^oC$ and $2^oC$. With business as usual scenarios, we are at the moment even heading towards an average global warming of 4 to $6^oC$ and still towards an up to $3^oC$, if countries fulfill their publicly announced mitigation targets.

The report on **Global Warming of $1.5^oC$** finds that limiting global warming to $1.5^oC$ would require "rapid and far-reaching" transitions in land, energy, industry, buildings, transport, and cities. Global net human-caused emissions of carbon dioxide ($CO_2$) would need to fall by about 45 percent from 2010 levels by 2030, reaching 'net zero' around 2050. This means that any remaining emissions would need to be balanced by removing $CO_2$ from the air.

The UN family is at the forefront of the effort to save our planet. **At the 21st Conference of the Parties in Paris in 2015, also called Paris Climate Summit**, Parties to the UNFCCC reached a landmark agreement to combat climate change and to accelerate and intensify the actions and investments needed for a sustainable low carbon future. The Paris Agreement central aim is to strengthen the global response to the threat of climate change by keeping a global temperature rise this century well below 2 degrees Celsius above pre-industrial levels and to pursue efforts to limit the temperature increase even further to 1.5 degrees Celsius. The Paris Agreement requires all Parties to put forward their best efforts through nationally determined contributions (NDCs) and to strengthen these efforts in the years ahead. This includes requirements that all Parties report regularly on their emissions and on their implementation efforts.

The motivation behind the current study is very well explained and hence it is a small contribution in the effort to curb global warming and make the world sustainable for future. The current study deals with prediction of carbon dioxide emissions using longitudinal data based on certain factors that are responsible for growth of the countries and in turn leading to increase in carbon dioxide emissions. The study will help countries to predict their carbon dioxide emissions level in the coming future as well as helps in identifying countries which are under the same level. The effort to curb the climate change becomes more powerful if the countries can be grouped which are under similar conditions and policy formulation to reduce carbon emission can be done efficiently.

## 1.2 Nature of Data

The success of any regression analysis ultimately depends on the availability of the appropriate data. It is therefore essential that we spend some time discussing the nature, sources, and limitations of the data that one may encounter in the analysis. Three types of data may be available for regression analysis: time series, cross-section, and pooled (i.e., combination of time series and cross-section) data.

1. Time Series Data: A time series data is a set of observations on the values that a variable takes at different times. Such data may be collected at regular time intervals, such as daily (e.g., stock prices, weather reports), weekly (e.g., money supply figures), monthly [e.g., the unemployment rate, the Consumer Price Index (CPI)], quarterly (e.g., GDP), annually (e.g., government budgets), quinquennially, that is, every 5 years (e.g., the census of manufactures), or decennially (e.g., the census of population).

2. Cross-Section Data: Cross-section data are data on one or more variables collected at the same point in time, such as the census of population conducted by the Census Bureau every 10 years (the latest being in year 2000), the surveys of consumer expenditures conducted by the University of Michigan, and, of course, the opinion polls by Gallup and umpteen other organizations.

3. Pooled Data: In pooled, or combined, data are elements of both time series and cross-section data. One example is GNP per capita of all European countries over ten years.

## 1.3 Longitudinal Data

Panel, Longitudinal, or Micropanel Data is a special type of pooled data in which the same cross-sectional unit (say, a family or a firm) is surveyed over time. For example, the U.S. Department of Commerce carries out a census of housing at periodic intervals. At each periodic survey the same household (or the people living at the same address) is interviewed to find out if there has been any change in the housing and financial conditions of that household since the last survey. By interviewing the same household periodically, the panel data provides very useful information on the dynamics of household behavior.

### 1.3.1 Difference between pooled and longitudinal data: Explained

In **pooled data**, we will take random samples in different time periods, of different units, i.e. each sample we take, will be populated by different individuals. This is often used to see the impact of policy or programmes. For example we will take household income data on households X, Y and Z, in 1990. And then we will take the same income data on households G, F and A in 1995. Although we are interested in the same data, we are taking different samples (using different households) in different time periods.

In **longitudinal data**, we are following the same units i.e. the same households or individuals over time. For example we will follow the same set of households X, Y and Z, for each time period we collect data i.e. in 1990 and we will also interview the same households in 1995.

### 1.3.2 Longitudinal and Clustered Data

- The defining feature of longitudinal studies is that measurements of the **same individuals are taken repeatedly through time**, thereby allowing the direct study of change over time. With repeated measures on individuals, one can capture within-individual change.

- A distinctive feature of longitudinal data is that **they are clustered**. In longitudinal studies the clusters are composed of the repeated measurements obtained from a single individual at different occasions. Observations within a cluster will typically exhibit positive correlation, and this correlation must be accounted for in the analysis.

- Longitudinal data also have a **temporal order**; the first measurement within a cluster necessarily comes before the second measurement, and so on. The ordering of the repeated measures has important implications for analysis

- **Alternatively, clustered data can arise from random sampling of naturally occurring groups in the population**. Families, households, hospital wards, medical practices, neighborhoods, and schools are all instances of naturally occurring clusters in the population that might be the primary sampling units in a study.

### 1.3.3 Longitudinal Data: Basic Concepts

A longitudinal study design permits the discovery of individual characteristics that can explain these inter-individual differences in changes in outcomes over time. Thus the defining feature of a longitudinal study is that two or more observations of the response variable, taken at different times, are made on at least some of the study participants. Typically, although not always, longitudinal study designs call for a fixed number of repeated measurements to be made on all study participants at a set of common time points. The occasions of measurement are not necessarily distributed evenly throughout the duration of the study.

A longitudinal analysis of within-individual changes proceeds in two conceptually distinct stages. First, within-individual change in the response is characterized in terms of some appropriate summary of the changes in the repeated measurements on each individual during the period of observation (e.g., using "difference scores" or some form of "response trajectory"). Second, these estimates of within-individual changes are then related to inter-individual differences in selected covariates. Although these two stages of the analysis are conceptually distinct, they can be combined in a statistical model for longitudinal data. That is, a single statistical model for longitudinal data can be used to both capture how individuals change over time and to relate within-individual changes in the response to selected covariates.

In a longitudinal study the participants, or, more generally, the units being studied, are referred to as individuals or subjects. In many, but certainly not all, longitudinal studies, the individuals are human subjects. In other longitudinal studies, the individuals may be animals (e.g., laboratory mice or rats). Depending on the specific context, we use the terms individuals and subjects interchangeably to refer to the participants in a longitudinal study. Thus, adopting the terminology introduced so far, the defining feature of a longitudinal study design is that measurements of the response variable are taken on the same individuals at several occasions.

The number and the timing of the repeated measurements are the same for all individuals, regardless of whether the occasions of measurement are equally or unequally distributed throughout the duration of the study and it is called as being "balanced" or "unbalanced" over time respectively.

### 1.3.4 Notations

To set the stage for the statistical discussion to follow, it is helpful to present a unified notation for the various aspects of the longitudinal design. We index the N subjects in the longitudinal study as

$$i = 1, \cdots, N \text{ subjects}$$

For a balanced design in which all subjects have complete data, and are measured on the same occasions, we index the measurement occasions as

$$j = 1, \cdots, n \text{ observations}$$

or in the unbalanced case of unequal numbers of measurements or different time-points for different subjects

$$j = 1, \cdots, n_i \text{ observations for subject i}$$

The total number of observations is given by $\sum_{i=1}^{N} n_i$
The repeated responses, or outcomes, or dependent measures for subject $i$ are denoted as the vector

$$y_i = n_i \times 1$$

The values of the p predictors, or covariates, or independent variables for subject $i$ on occasion $j$ are denoted as (including an intercept term):

$$x_{ij} = p \times 1$$

For time-invariant predictors (between subject, e.g., sex), the values of $x_{ij}$ are constant for$j = 1, \cdots, n_i$. For time-varying predictors (within-subject,e.g., age), $thex_{ij}$ can take on subject-and timepoint-specific values. To describe the entire matrix of predictors for subject $i$, we use the notation

$$X_i = n_i \times p$$

### 1.3.5 Data Layout

In fixing ideas for the statistical development to follow, it is also useful to apply this previously described notation to describe a longitudinal dataset as follows.

| Subject | Observation | Response | Covariates |
|---|---|---|---|
| 1 | 1 | $y_{11}$ | $x_{111} \cdots x_{11p}$ |
| 1 | 2 | $y_{12}$ | $x_{121} \cdots x_{12p}$ |
| . | . | . | $\cdots$ |
| 1 | $n_i$ | $y_{1n_i}$ | $x_{1n_i1} \cdots x_{1n_ip}$ |
| . | . | . | $\cdots$ |
| . | . | . | $\cdots$ |
| N | 1 | $y_{N1}$ | $x_{N11} \cdots x_{N1p}$ |
| N | 2 | $y_{N2}$ | $x_{N21} \cdots x_{N2p}$ |
| . | . | . | $\cdots$ |
| N | $n_N$ | $y_{Nn_N}$ | $x_{Nn_N1} \cdots x_{Nn_Np}$ |

## 1.4 Advantages and Disadvantages of Longitudinal Data

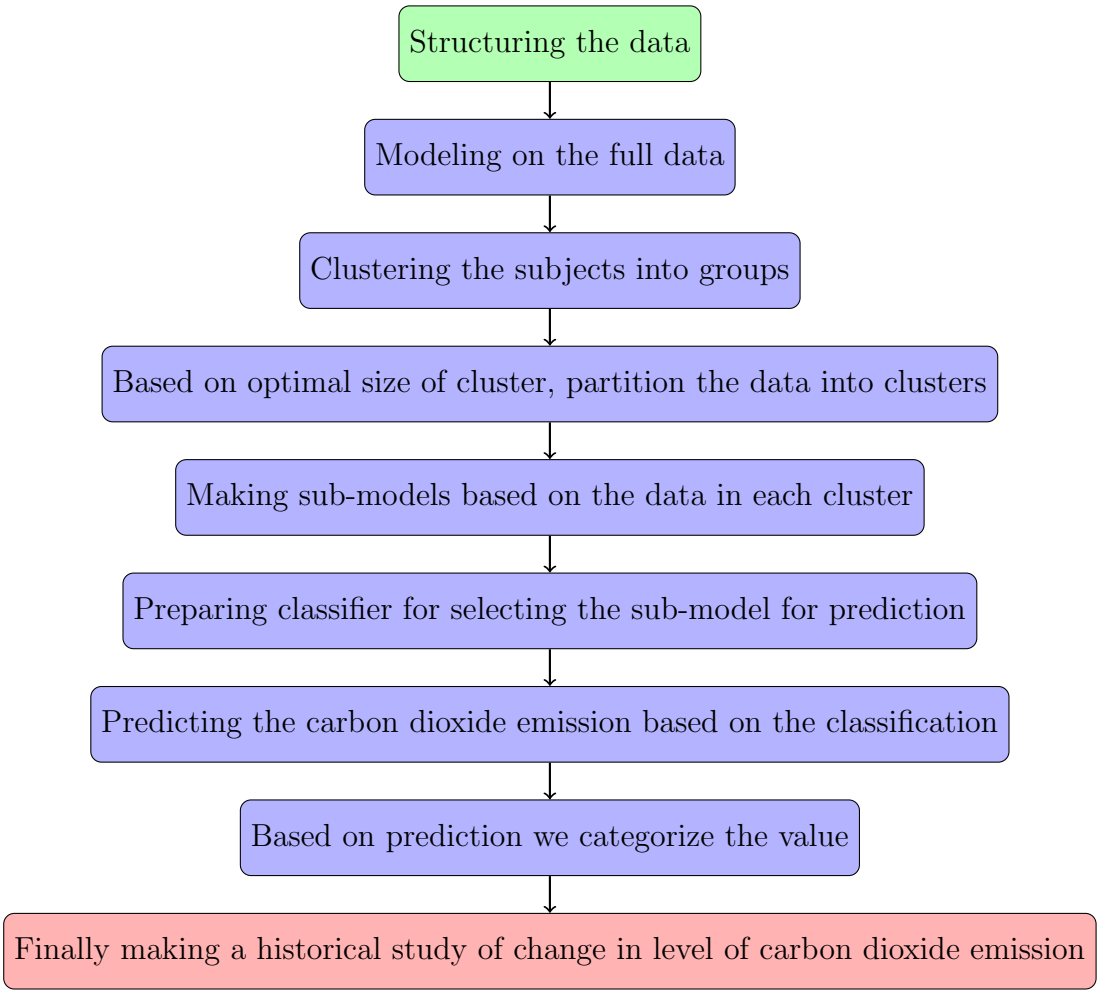
There are several advantages of the longitudinal data. First, in order to achieve similar level of statistical power as in cross-sectional data, fewer subjects are required in a longitudinal study. The reason for this is that repeated observations from the same subject, while correlated, are rarely, if ever, perfectly correlated. The net result is that the repeated measurements from a single subject provide more independent information than a single measurement obtained from a single subject. Second, in a longitudinal study, each subject can serve as his/her own control. Third, longitudinal studies allow an investigator to separate aging effects, (i.e. changes over time within individuals), from cohort effects (i.e. differences between subjects at baseline). Such cohort effects are often mistaken for changes occurring within individuals. Without longitudinal data, one cannot differentiate these two competing alternatives. Finally, longitudinal data can provide information about individual change, whereas cross-sectional data cannot. Statistical estimates of individual trends can be used to better understand heterogeneity in the population and the determinants of growth and change at the level of the individual.

Despite their advantages, longitudinal data are not without their challenges. Observations are not, by definition, independent and we must account for the dependency in data using more sophisticated statistical methods and these analytical methods are not well developed. An added complication that arises in the context of analysis of longitudinal data is the invariable presence of missing data. In some cases, a subject may be missing one of several measurement occasions; however, it is more likely that there are missing data due to attrition or "drop-out". Unfortunately, the available sample at the end of the study may have little resemblance to the sample initially randomized. Reasons for not completing the study may be confounded with the effects that the study was designed to investigate. Nevertheless, the presence of missing data, and its treatment in the statistical analysis, is a complicating

feature of longitudinal data, making analysis potentially far more complex than analysis of cross-sectional data. Yet another complicating feature of longitudinal data is that not only does the outcome measure changeover time, but the values of the predictors or independent variables can also change over time. The treatment of time-varying covariates in analysis of longitudinal data permits much stronger statistical inferences about dynamical relationships than can be obtained using cross-sectional data but makes model complex. Finally, in some cases, the repeated measurements involve different conditions that the same subjects are exposed to.

## 1.5 Outline of the working

Based on the motivation, the steps followed in the dissertation are as given in the flowchart:



# Chapter 2

# Modeling

## 2.1 Introduction

Regression analysis is a statistical technique for investigating and modeling the relationship between variables. Applications of regression are numerous and occur in almost every field. Regression models are widely used and provide a very general approach for analyzing data. We refer regression model to any model that describes the dependence of the response variable on a set of covariates in terms of some form of regression equation. In particular, the regression parameters express how the mean of the response variable depends on the covariates.

## 2.2 Summarizing and Visualizing Longitudinal Data

Simple statistical methods are not the best method to analyze longitudinal data. But keeping in mind the objective of our study, a common way to simplify longitudinal data is to reduce each multivariate response $Y_i$ to a univariate summary and then apply a familiar analysis or we can do a year wise analysis since the sample drawn comes from a single population. In order to understand the dynamics of the data, we can study the summary statistics of each country or we can study groups or we can take the complete picture as a whole.

The best way to graphically display longitudinal data are time plots and spaghetti plots (another name for profile plots). In order to understand the trend and outliers, scatter plots and boxplots prove to be good visualization tools. Even yearwise scatterplot matrices prove to be a great tool for analysis. But too many subjects tend to destroy the visualization

so we should be careful in selecting the kind of plot we choose for our analysis.

## 2.3 Variable Selection and model building

***The best strategy to build model is:***

- Fit the full model (the model with all of the regressors under consideration).
- Perform a thorough analysis of this model, including a full residual analysis. Often, we should perform a thorough analysis to investigate possible collinearity. But since we are working with longitudinal data, collinearity is not a problem.
- Determine if transformations of the response or of some of the regressors are necessary.
- Use the t tests on the individual regressors to edit the model or choose significant variables.
- Perform a thorough analysis of the edited model, especially a residual analysis, to determine the model's adequacy.

***Strategy for variable selection (Stepwise regression):***

- Fit the largest model possible to the data.
- Determine if all possible regressions are feasible.
- If all possible regressions are feasible, perform all possible regressions using such criteria as $R^2$, Adjusted $R^2$, AIC or BIC.
- If all possible regression is not feasible then we go for stepwise selection which is of three types: forward selection, backward elimination and stepwise selection.
- Forward Selection is done on the basis of addition of variables based on correlation with highest correlation variable entering first followed by variables based on the increasing order of partial correlation. Backward Elimination is a powerful method where variables get eliminated based on ordering due to partial F statistics and finally stepwise selection is a combination of both forward and backward in which backward elimination is applied on variables selected through forward selection.
- Compare and contrast the best models recommended by each method.

- Finally, based on subject knowledge and the above process, we select the variables and fit the model.

## 2.4 Linear Models for Longitudinal Continuous Data

A feature of the regression modeling approach is that it can incorporate mixture of both discrete (qualitative) and continuous (quantitative) covariates. It is not necessary to distinguish whether the covariates are continuous or discrete (or a mixture of two) within a regression paradigm. However, from a purely historical perspective, linear models for continuous response with only discrete covariates have often been referred to as *analysis of variance* (ANOVA) models while linear models for a continuous response with only continuous covariates (or mixture of both) have often been referred as *linear regression* models. Later, it was recognized that linear regression is a very general model that incorporates analysis of variance as a special case.

We view this regression paradigm as a very flexible and versatile approach for analyzing longitudinal and correlated data as well. Regression models can provide a parsimonious description or explanation of how the mean response in a longitudinal study changes with time and how these changes are related to covariates of interest. ***Acknowledging and representing correlation among responses on the same individual over time is central to modeling and analysis of longitudinal data***. The conceptual framework clarifies that correlation comes about because of phenomena acting both within and among individuals, which are represented in different ways within different modeling strategies. Our primary goal is to provide a simple description of the discernible patterns of change in the response over time, and their relation to covariates, via regression coefficients that bear directly on the scientific questions of main interest.

### 2.4.1 Notation

We assume that a sample of N subjects is measured repeatedly over time. Let $Y_{ij}$ denote the response variable for the $i^{th}$ subject on the $j^{th}$ measurement occasion. Subjects may not have same number of repeated measures and may not be measured at the same set of occasions. To accommodate both the features, we assume that there are $n_i$ repeated measurements of the response on the $i^{th}$ subject and that each $Y_{ij}$ is observed at time $t_{ij}$. When the number of repeated measures is the same for all subjects in the study (and there

are no missing data) it is not necessary to include the index i in $n_i$(since $n_i = n$, for i=1,...,N). Similarly if the repeated measures are observed at the same set of occasions it is not necessary to include the index i in $t_{ij}$(since $t_{ij} = t_j$, for i=1,...,N. Since our study is based on balanced data so it is convenient to group the response variables for the $i^{th}$ subject into an $n \times 1$

$$Y_i = \begin{pmatrix} Y_{i1} \\ Y_{i2} \\ \vdots \\ Y_{in} \end{pmatrix}, i = 1, ..., N$$

Associated with each response $Y_{ij}$, there is a $p \times 1$ vector of covariates

$$X_{ij} = \begin{pmatrix} X_{ij1} \\ X_{ij2} \\ \vdots \\ X_{ijp} \end{pmatrix}, i = 1, ..., N; j = 1, ..., n$$

We can group the vectors of covariates into a $n \times p$ matrix of covariates

$$X_i = \begin{pmatrix} X_{i1}^{'} \\ X_{i2}^{'} \\ \vdots \\ X_{in}^{'} \end{pmatrix} = \begin{pmatrix} X_{i11} & X_{i12} & \cdots & X_{i1p} \\ X_{i21} & X_{i22} & \cdots & X_{i2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{in1} & X_{in2} & \cdots & X_{inp} \end{pmatrix}, i = 1, 2, ..., N$$

Next, we consider a linear regression model for changes in the mean response over time and for relating the changes to the covariates,

$$Y_{ij} = \beta_0 + \beta_1 X_{ij1} + \beta_2 X_{ij2} + ... + \beta_p X_{ijp} + e_{ij}, j = 1, 2, ..., n$$

where the unknown regression parameters are grouped together into a $p \times 1$ vector, $\beta = (\beta_1, \beta_2, ..., \beta_p)^{'}$ Here the random errors, with mean zero, representing the deviations of the responses from their corresponding predicted means and hence

$$E(Y_{ij}|X_{ij}) = \beta_0 + \beta_1 X_{ij1} + \beta_2 X_{ij2} + ... + \beta_p X_{ijp}, j = 1, 2, ..., n$$

Finally, using vector and matrix notation, the regression model can be expressed in an even more compact form as

$$Y_i = X_i \beta + e_i, i = 1, ..., N$$

where $e_i = (e_{i1}, e_{i2}, ..., e_{in})^{'}$ is an $n \times 1$ vector of random errors associated with the corresponding elements of the vector of responses on the $i^{th}$ subject. The regression model is

given by

$$\begin{pmatrix} Y_{i1} \\ Y_{i2} \\ \vdots \\ Y_{in} \end{pmatrix} = \begin{pmatrix} 1 & X_{i11} & X_{i12} & \cdots & X_{i1p} \\ 1 & X_{i21} & X_{i22} & \cdots & X_{i2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{in1} & X_{in2} & \cdots & X_{inp} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} e_{i1} \\ e_{i2} \\ \vdots \\ e_{in} \end{pmatrix}.$$

### 2.4.2 Assumptions

We introduced the basic regression model for longitudinal data. However, some changes need to be made to the model based on the assumptions on source of variation and the form of mean vectors and also in estimating the parameters for our objective. However, we restrict our study to the case of balanced data i.e. where all units have repeated measurements at the same n time points.

- All units or subjects are from the single population.
- $Y_i$ has a multivariate probability distribution such that $E(Y_i) = \mu$ and $Var(Y_i) = \sum$
- It is natural to be concerned that components $Y_{ij}, j = 1, 2, ..., n$ are correlated.
- It is unrealistic to expect that $cov(Y_{ij}, Y_{ik}) \neq 0$ for any $j \neq k = 1, 2, ..., n$ such that $\sum$ is unlikely to be a diagonal matrix but if observations are taken far enough apart in time; they might be viewed as independent.
- On the other hand, if each $Y_i$ corresponds to different units and the units are not related in any way, then it seems reasonable to suppose that $Y_1, Y_2, ..., Y_N$ are mutually independent.
- Based on previous assumption and response variable being continuous in nature, it is often assumed that $e_i$ are independent $N(0, \sigma^2 I_n)$ such that $Y_i \sim N(X_i'\beta, \sigma^2 I_n)$. The reason for covariance to be a scalar is that $e_i$ are assumed to be independent within the units.
- Even if we consider within units observation to be correlated or the off-diagonal elements to be unequal we can use Generalized Least Square or Weighted Least Square approach respectively to get constant scalar variance.

### 2.4.3 Estimation: Maximum Likelihood

Based on the full assumptions that have been made about the response, a very general approach to the estimation is the method of maximum likelihood. The values of the parameters that maximize the likelihood function are called maximum likelihood estimates of the parameters.

Suppose that the data arise from a series of cross-sectional studies that are repeated at n different occasions. At each occasion, the data is obtained on a sample of N units. Hence it is reasonable to assume that the observations are independent of one another, since each unit is measured at only one occasion. Also for ease of calculation, we assume that the variance is constant, say $\sigma^2$. The mean response is related to the covariates via the following linear regression model:

$$E(Y_{ij}) = X_{ij}^{'}\beta \tag{2.1}$$

To obtain maximum likelihood estimates, we must find the values of the regression parameters that maximize the joint normal probability density function of all the observations, evaluated at the observed values of the response, and regarded as a function of $\beta$ and $\sigma^2$. The univariate normal density function for $Y_{ij}$ can be expressed as

$$f(y_{ij}) = (2\pi\sigma^2)^{-1/2}\exp\{-\frac{1}{2}(y_{ij} - x_{ij}^{'}\beta)^2/\sigma^2\} \tag{2.2}$$

When all observations are independent of one another, the likelihood function is simply the product of individual univariate normal pdf for $Y_{ij}$

$$\prod_{i=1}^{N}\prod_{j=1}^{n} f(y_{ij}) \tag{2.3}$$

The log-likelihood function will be

$$l = log\Big\{\prod_{i=1}^{N}\prod_{j=1}^{n} f(y_{ij})\Big\} = -\frac{nN}{2}log(2\pi\sigma^2) - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{n}(y_{ij} - x_{ij}^{'}\beta)^2/\sigma^2 \tag{2.4}$$

We maximize the likelihood by differentiating the above equation with respect to $\beta$ and $\sigma^2$ and equating it to zero. That is,the least squares (ML) estimator solves the set of p equations.

$$\sum_{i=1}^{N}\sum_{j=1}^{n}(Y_{ij} - x_{ij}^{'}\beta)x_{ij} = 0 \tag{2.5}$$

The following results are obtained:

$$\hat{\beta} = \left\{ \sum_{i=1}^{N} \sum_{j=1}^{n} (x_{ij} x_{ij}^{'}) \right\}^{-1} \sum_{i=1}^{N} \sum_{j=1}^{n} (x_{ij} y_{ij}) \tag{2.6}$$

$$\hat{\sigma^2} = \sum_{i=1}^{N} \sum_{j=1}^{n} \frac{(y_{ij} - x_{ij}^{'} \hat{\beta})^2}{nN} \tag{2.7}$$

## 2.5 Generalized Linear Models for Longitudinal Data

In previous section, we saw methods for analyzing longitudinal data when response variable is continuous but this approach has several deficiencies as a model for count, binary, or some positive continuous data:

- The normal distribution may not be a good probability model.
- Variance may not be constant across the range of the response

Transformation of the response variable is often a very effective way to deal with both response non-normality and inequality of variance. Weighted least squares is also a potentially useful way to handle the non - constant variance problem. In this chapter, we present an alternative approach to data transformation when the "usual" assumptions of normality and constant variance are not satisfied. This approach is based on the generalized linear model (GLM).

The GLM is a unification of both linear and nonlinear regression models that also allows the incorporation of non-normal response distributions. In a GLM, the response variable distribution must only be a member of the exponential family, which includes the normal, Poisson, binomial, exponential, and gamma distributions as members. Furthermore, the normal-error linear model is just a special case of the GLM, so in many ways, the GLM can be thought of as a unifying approach to many aspects of empirical modeling and data analysis.

### 2.5.1 Salient Features of Generalized Linear Models

Generalized linear models extend the standard linear regression model in a number of important ways, while also retaining some of its distinctive features. In particular, a generalized linear model for $Y_i$ has the following three-part specification:

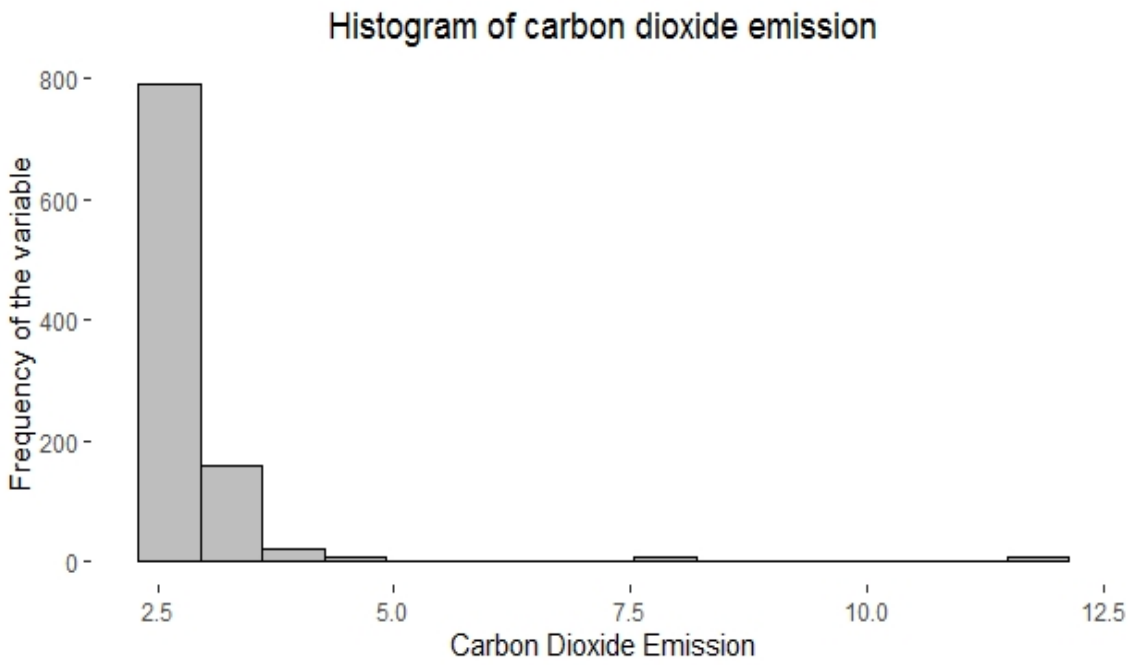
1. a distributional assumption,

2. a systematic component and

3. a link function

### *Distributional Assumption*

Generalized Linear models assume that the response variable has a probability distribution belonging to the so-called exponential family of distributions. The distributional assumption specifies the random component of the model which in turn specifies a probabilistic mechanism by which the responses are assumed to be generated. In order to know about the distribution, the basic step that can be taken is to get a histogram plot of the response variable and also know the type of the data. In our study the response variable is the carbon dioxide emission and the histogram appears as:




From the above plot and continuous nature of response variable, we can say that distributional assumption for this data is Gamma distribution

### *Systematic component*

Generalized linear model not only share a common family of distributions but also share a common regression formulation. An important aspect of the standard linear regression model that is retained in all generalized linear models is the linear regression component.

This is the systematic component of a generalized linear model and it specifies that the effects of the covariates, $X_i$ on the mean of $Y_i$ can be expressed in terms of the following linear predictor

$$\eta_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} +, ..., +\beta_p X_{ip} \tag{2.8}$$

### *Link Function*

The final way in which generalized linear models extend the standard linear regression model is by taking a suitable transformation of the mean response and relating the transformed mean response to the covariates. This is achieved through link function. The link function applies a transformation to the mean and then links the covariates, via the linear predictor, to the transformed mean of the distribution of the responses,

$$g(\mu_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} +, ..., +\beta_p X_{ip} \tag{2.9}$$

where the link function g(.) is some well-known function like log, inverse or identity.

## 2.5.2 Continuous Data with Constant Coefficient of Variation – The Gamma Distribution

The normal distribution says that values to the left and right of its mean are equally likely to be seen, by virtue of the symmetry inherent in the form of the probability density. This may not be realistic for biological and other kinds of data. A common phenomenon is to see "unusually large" values of the response with more frequency than "unusually small" values. Other probability models are available for continuous response that better represent these features. Several such models are possible; we consider one of these.

The gamma probability distribution describes the probabilities with which a random variable Y takes on values, where Y can only be positive. More precisely, the probability density function for value y is given by

$$f(y_{ij}) = \frac{1}{y_{ij}\Gamma(\frac{1}{\sigma^2})}\left(\frac{y_{ij}}{\sigma^2\mu}\right)^{1/\sigma^2} \exp\left(-\frac{y_{ij}}{\sigma^2\mu}\right), \mu, \sigma^2 > 0, y > 0 \tag{2.10}$$

- It is clearly visible that $E(Y_i) = \mu$
- Also the variance $Var(Y_i) = \sigma^2\mu^2$. The variance here is non-constant; it depends on the value of $\mu$.Thus, if $Y_1$ and $Y_2$ are both gamma random variables, then the only way

that they can have the same variance is if they have the same mean $\mu$ and the same value of the parameter $\sigma^2$

- Thus, for regression, if $Y_1$ and $Y_2$ correspond to responses taken at different covariate settings, it is inappropriate to take them to have the same variance. Thus, as above, the assumption of constant variance is not appropriate for a response that is well-represented by the gamma probability model.

- We note here that the symbol $\sigma^2$ is being used in a different way how we showed previously, to represent a variance. Here, it turns out that $\sigma$ has the interpretation as the coefficient of variation, defined for any random variable $Y_i$ as

$$CV = \frac{\{Var(Y_i)\}^{\frac{1}{2}}}{E(Y_i)} \tag{2.11}$$

  that is, CV is the ratio of standard deviation of the response to mean, or "noise to signal". This ratio may be expressed as a proportion or a percentage; in either case, CV characterizes the "quality" of the data by quantifying how large the "noise" is relative to the size of the thing being measured.

- "Small" CV ("high quality") is usually considered to be $CV \leq 0.30$. "Large" CV ("low quality") is larger.

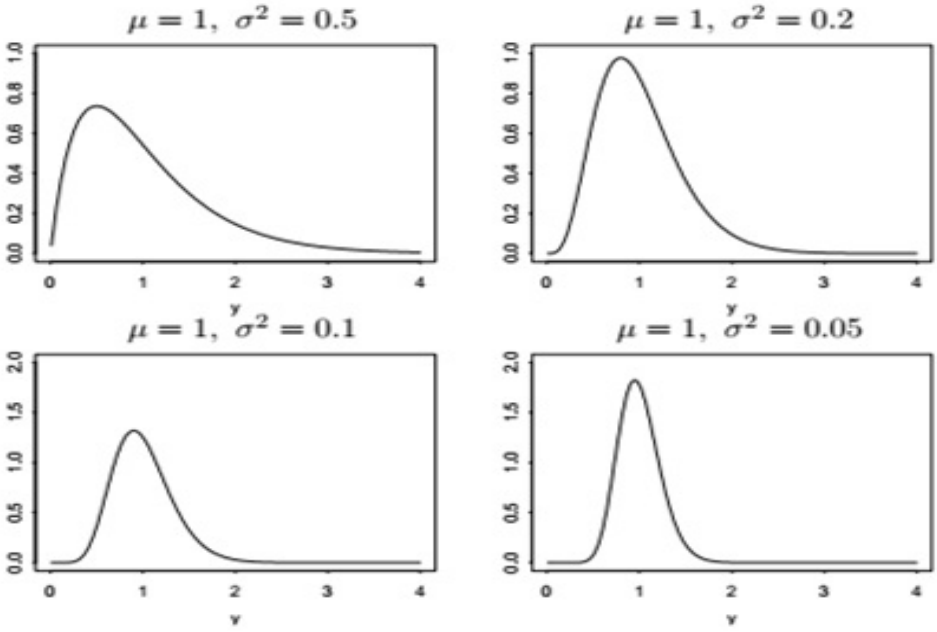
- Note that for gamma distribution,

$$CV = \frac{\{\sigma^2\mu^2\}^{\frac{1}{2}}}{\mu} = \sigma \tag{2.12}$$

  so that, regardless of the value of $\mu$, the ratio of "noise" to "signal" is the same. Thus, rather than having constant variance, the gamma distribution imposes constant coefficient of variation.

- As $\sigma^2$ becomes smaller, the shape of the curve begins to look more symmetric. Thus, if CV is "small" ("high quality" data), gamma probability distribution looks very much like a normal distribution.

- On the other hand, when $\sigma^2$ is relatively large, so that CV is "large" ("low quality" data), the shape is skewed. The figure below shows variation in the shape of the values with respect to change in the value of $\sigma^2$




Based on the above knowledge, we present our model for this study as follows:

(i) Mean: For regression modeling, we wish to represent the mean for $Y_i$ as a function of the covariates $x_i$ . If the size of the responses is not too large,then using a linear model $E(Y_i) = x_i^{'}\beta$ could be dangerous; thus, it is preferred to use a model that enforces positivity. One common model is the log linear model, which is also commonly used for count data. Both types of data share the requirement of positivity, so this is not surprising. When the size of the response is larger, it is often the case that the positivity requirement is not a big concern—even if a linear model is used to represent the data, because the responses are all so big, estimated means will still all be positive for covariate settings like those of the original data. This opens up the possibility for other models for the mean. **More generally, for small number of covariates the log scale proves to be much reasonable and for large number of covariates inverse or reciprocal proves to be appropriate**. Hence the model will be like

$$\log\{E(Y_i)\} = x_i^{'}\beta$$

$$and\ \frac{1}{E(Y_i)} = x_i^{'}\beta\ or\ E(Y_i) = \frac{1}{x_i^{'}\beta}$$

(ii) Probability distribution: The $Y_i$ are assumed to arise at each setting $x_i$ from a gamma distribution with mean as given above, or some other model deemed appropriate. The $Y_i$ are also assumed to be independent.

(iii) Variance: Under the gamma assumption, the variance of $Y_i$ is proportional to the square of the mean response; i.e. constant coefficient of variation. Thus, if the mean is represented as above, then we must have that the variance of $Y_i$ is given by

$$Var(Y_i) = \sigma^2 E(Y_i)^2 = \sigma^2 \{exp(x_i^{'}\beta)\}^2$$

$$and\ Var(Y_i) = \sigma^2 E(Y_i)^2 = \sigma^2 \left(\frac{1}{x_i^{'}\beta}\right)^2$$

### 2.5.3 General Notation

We are now in a position to state all of this more formally. A generalized linear model is a regression model for response $Y_i$ with the following features:

- The mean of $Y_i$ is assumed to be of the form

$$E(Y_i) = f(x_i^{'}\beta) \tag{2.13}$$

It is customary to express this a bit differently, however. The function f is almost always chosen to be monotone; that is, it is a strictly increasing or decreasing function of $x_i^{'}\beta$. This means that there is a unique function g, say, called the inverse function of f, such that we may re-express model in the form

$$g\{E(Y_i)\} = x_i^{'}\beta \tag{2.14}$$

The function g is called the link function because it links the mean and the covariates. The linear combination of covariates and regression parameters $x_i^{'}\beta$ is called the linear predictor

- The probability distribution governing $Y_i$ is assumed to be one of those from the scaled exponential family class.

- The variance $Y_i$ is assumed to be of the form

$$Var(Y_i) = \phi V\{E(Y_i)\} \tag{2.15}$$

where the function V depends on the distribution and $\phi$ might be equal to a known constant. The function V is referred to as the variance function. The parameter $\phi$ is often called the dispersion parameter because it has to do with variance.

### 2.5.4 Estimation: Maximum Likelihood and Iteratively Reweighted Least Squares

***MAXIMUM LIKELIHOOD:*** The class of generalized linear models may be thought of as extending the usual classical linear model to handle special features of different kinds of data. The extension introduces some complications, however. In particular:

- The model for mean response need no longer be a linear model.
- The variance is allowed to depend on the mean; thus, the variance depends on the regression parameter $\beta$.

A natural approach to estimating $\beta$ in all generalized linear models is thus to appeal to the principle of maximum likelihood. It turns out that, fortuitously, the form of the joint density of random variables $Y_1, Y_2, ..., Y_N$ that arise from any of the distributions in the scaled exponential family class has the same general form. Thus, it turns out that the ML estimator for $\beta$ in any generalized linear model solves a set of p equations of the same general form:

$$\sum_{i=1}^{N}\sum_{j=1}^{n}\frac{1}{V\{f(x_{ij}^{'}\beta)\}}\{Y_{ij}-f(x_{ij}^{'}\beta)\}f^{'}(x_{ij}^{'}\beta)x_{ij}=0 \tag{2.16}$$

where $f^{'}(u)=\frac{d}{du}f(u)$, the derivative of f with respect to its argument.
The equation (2.16) and the equation for the linear, normal, constant variance model (2.5) share the feature that they are both linear functions of the data $Y_{ij}$ and are equations we would like to solve in order to obtain the maximum likelihood estimator for fl. Thus, they are very similar in spirit. However, they differ in several ways:

- Each deviation $Y_{ij}-f(x_{ij}^{'}\beta)$ in (2.16) is weighted in accordance with its variance (the scale parameter is a constant). Of course, so is each deviation in for normal but, in that case, the variance is constant for all j. Recall that weighting in accordance with variance is a sensible principle, so it is satisfying to see that, despite the difference in probability distributions, this principle is still followed. Here, the variance function depends on $\beta$ , so now the weighting depends on $\beta$. Thus, $\beta$ appears in this equation in a very complicated way.
- Moreover, $\beta$ also appears in the function f, which can be quite complicated–the function f is certainly not a linear function of $\beta$.

The result of these differences is that, while it is possible to solve (2.5) explicitly, it is not possible to do the same for (2.16). Rather, the solution to (2.16) must be found using a numerical algorithm. The numerical algorithm is straightforward and works well in practice, so this is not an enormous drawback.

***ITERATIVELY REWEIGHTED LEAST SQUARES:*** It turns out that there is a standard algorithm that is applicable for solving equations of the form (2.16). The basic idea is (operating on the observed data)

- Given a starting value, or guess for $\beta$, $\beta^0$, say, evaluate the weights at $\beta^0 : 1/V\{f(x_{ij}, \beta^0)\}$
- Pretending the weights are fixed constants not depending on $\beta$ , solve equation (2.16). This still requires a numerical technique, but may be accomplished by something that is approximately like solving (2.5). This gives a new guess for $\beta, \beta^1$ , say.
- Evaluate the weights at $\beta^1$ and repeat. Continue updating till the two successive $\beta$ values are the same.

The repeatedly updating of the weights along with the approximation to solve an equation like (2.16) gives this procedure its name: iteratively reweighted least squares, often abbreviated as IRWLS or IWLS. Luckily, there are standard ways to find the starting value based on the data and knowledge of the assumed probability distribution. Thus, the user need not be concerned with this (usually); software typically generates this value automatically.

## 2.6 Robust Generalized Linear Models for Longitudinal Data

Our data presents a high level of skewness, with a relatively small number of subjects accounting for a large portion of carbon dioxide emission. For this reason, the data can be modeled as a positive random variable with asymmetric distribution. However, due to the presence of extreme observations and the skewness of the resulting distribution, the mean is a difficult parameter to be estimated well, even in the univariate case: the sample mean, which is the natural estimate, is absolutely non-robust. Moreover, when comparing emission means among different countries over different periods of time, a few a typical observations can drastically change the mean estimate, with the consequence that the common tests based on means (e.g. t-test and its variants) become highly unreliable. The process of

removing outliers from the data set to reduce the impact of such extreme observations can give misleading results.

In the multivariate framework, one of the main approaches to regression with asymmetric errors is Generalized Linear Models (GLM), which allows modeling of the mean with the help of covariates using maximum likelihood estimation. The GLM Gamma model, according to the current study, can be seen as a target model in terms of fitting performances. Such solution has been widely discussed and compared to several other approaches resulting perhaps to be the best compromise between fitting capability and handiness of implementation. Nevertheless, since the GLM technique is based on maximum likelihood or quasi-likelihood, it is very sensitive to spurious observations, producing very imprecise estimates if the log-scale error is heavy tailed. Hence, we propose use of robust estimating equations via M-estimation for the Gamma model and a class of high efficiency and high breakdown point estimators.

### 2.6.1 Method

In modeling framework of GLM Gamma with inverse link the random response variable $Y_i$, for i=1,2,. . ., N such that:
$E(Y_i|x_i) = f(x_i^{'}\beta) = \mu_i, Var(Y_i|x_i) = \phi v\{\mu_i\}$ *and* $g(\mu_i) = x_i^{'}\beta$ *for* $i = 1, 2, ..., n$ where $\beta \epsilon R^P$ is the vector parameters, $x_i \epsilon R^{n\times p}$ is a set of explanatory variables, g(.) is the inverse function and the variance $V\{\mu_i\}$ is proportional to $\mu_i^2$. This model will fit the positively skewed data by means of the model $1/y_i = x_i^{'}\beta + e_i$, where error term has constant variance.

The Gamma Robust estimator of Cantoni and Ronchetti is based on a set of M-estimating equations for the $\beta$ regression parameters of the form $\sum_{i=1}^{n} \Psi(y_i, \beta, v) = 0$, where $\Psi$ is the so called Huber's function applied on the Pearson residuals, that down weights observations far from the majority of the data:

$$\sum_{i=1}^{n} \left[ \Psi(r_i; c) w(x_i) \frac{1}{(\phi v(\mu_i))^{\frac{1}{2}}} \mu_i^{'} - \alpha(\beta) \right] = 0$$

where $r_i = (y_i - \mu_i)/v^{\frac{1}{2}}(\mu_i)$are the Pearson residuals and $\mu_i^{'} = \partial\mu_i/\partial\beta$. The correction term $\alpha(\beta) = \frac{1}{n}\sum_{i=1}^{n} E[\Psi(r_i; c) w(x_i) \frac{1}{(\phi v(\mu_i))^{\frac{1}{2}}} \mu_i^{'}]$ ensures Fisher consistency with respect to the mean parameter $\mu$ at the model. To estimate the Huber's proposal 2 is applied.

Huber function $\Psi(r_i; c)$ :

$$\Psi(r_i; c) = \begin{cases} r_i, & |r_i| < c \\ c.sign(r_i), & otherwise \end{cases}$$

Weights on the design: either based on the hat matrix (e.g. $w(x_i) = \sqrt{\{1 - h_{ii}\}}$ or on the Mahalanobis distance, e.g. $w(x_i) = 1/\sqrt{1 + 8max\big(0, d_i^2 - \frac{q}{\sqrt{2q}}\big)}$, with robust estimates of the center and the scatter in $d_i$

Estimation of $\phi$ needs to be done robustly as well

Given $Var(\frac{Y_i - \mu_i}{\mu_i}) = \phi$, use a robust estimator of scale, for example Huber's proposal 2:

$$\sum_{i=1}^{n} \chi\left(y_i - \frac{\mu_i}{\sqrt{\phi v(\mu_i)}}; c\right) = 0$$

where $\chi(u; c) = \Psi^2(u; c) - \delta \; and \; \delta = E(\Psi^2(u; c))$ is a constant that ensures Fisher consistency for the estimation of $\phi$.

# Chapter 3

# Cluster Analysis

## 3.1 Introduction

Clustering is a process of identifying natural subgroups within a dataset. Clustering groups the data into clusters, such that objects within a cluster have high similarity in comparison to one another and objects between clusters have high dissimilarity. It has its roots in many areas like data mining, statistics, biology and machine learning. One of the uses of clustering is to deal with large data to group them into a set of clusters. The similarity between the objects is calculated by the use of a similarity function. The similarity measures used most commonly is based on distance functions such as Euclidean distance, Manhattan distance, Minskowski distance, Cosine similarity, etc. for interval valued data, simple matching, Jaccard coefficient, Pearson's phi-like coefficients, etc. are employed for binary data and Simple matching coefficient, Eskin measure, etc. for nominal data to group objects in clusters. Clustering using distance functions, called distance based clustering.

Clustering algorithms can be categorized broadly into the following categories:

1. Partitioning Algorithm
2. Density based clustering
3. Hierarchical Clustering

### 3.1.1 Partitioning Algorithm

Partition clustering algorithm divides the data points into 'k' partitions, where each partition represents a cluster. The partition is done based on a certain objective function.

The clusters are formed such that the data objects within a cluster are 'similar' and the data objects in different clusters are 'dissimilar'.

Partitioning algorithm methods are useful in applications where the number of clusters required is static. K-means, PAM (Partitioning around medoids) and CLARA are a few of the partitioning clustering algorithms.

### 3.1.2 Density-based clustering

Density-based clustering algorithms create arbitrary-shaped clusters. In this kind of clustering approach, a cluster is considered as a region in which the density of data objects exceeds a particular threshold value. DBSCAN algorithm is a famous example of density-based clustering.

### 3.1.3 Hierarchical Clustering

Hierarchical clustering algorithms work to divide or merge a particular dataset into a sequence of nested partitions. The hierarchy of these nested partitions can be of two types, viz. agglomerative, i.e., bottom-up or divisive, i.e., top-down

## 3.2 Distance Metrics

Distance metrics play an important role in order to measure the similarity between the data objects. The main requirement of metric calculation in a specific problem is to obtain an appropriate distance/similarity function. The distance metric for each type of data is explained as below:

### 3.2.1 Interval-valued data

Data can be measured on a continuum or scale is called an interval-valued data. It can use any of the following distance measures to calculate dissimilarity among them.

#### *Euclidean distance*

Euclidean distance is considered as standard metric for geometrical problems. It is simply the ordinary distance between two points. The Euclidean distance determines the root of

square differences between the coordinates of a pair of objects.

$$d(O_i, O_k) = \sqrt{|x_{i1} - x_{k1}|^2 + |x_{i2} - x_{k2}|^2 + \cdots + |x_{ip} - x_{kp}|^2}, i \neq k$$

### *Manhattan distance*

Manhattan distance is a distance metric that calculates the absolute differences between coordinates of pair of data objects

$$d(O_i, O_k) = |x_{i1} - x_{k1}| + |x_{i2} - x_{k2}| + \cdots + |x_{ip} - x_{kp}|, i \neq k$$

### *Minskowski distance*

Minskowski distance is the generalization of the Euclidean distance and the Manhattan distance. When q=2, Minskowski becomes Euclidean distance and when q=1, it becomes Manhattan distance.

$$d(O_i, O_k) = \sqrt[q]{(|x_{i1} - x_{k1}|^q + |x_{i2} - x_{k2}|^q + \cdots + |x_{ip} - x_{kp}|^q)}, i \neq k$$

### 3.2.2 Ordinal data

Ordinal data is a categorical, statistical data type where the variables have natural, ordered categories and the distances between the categories is not known. The ordinal scale is distinguished from the nominal scale by having a ranking. It also differs from interval and ratio scales by not having category widths that represent equal increments of the underlying attribute. Sometimes data on an interval scale or ratio scale are grouped onto an ordinal scale: for example, individuals whose income is known might be grouped into the income categories 0−19,999, 20,000−39,999, 40,000−59,999, ..., which then might be coded as 1, 2, 3, 4, ....

There are a couple of approaches to dealing with ordinal data. One approach is to use normalized ranks, Spearman distance or Footrule distance – which amount to treating ordinal data as interval data. Then there are Kendall, Cayley, and Ulam distances which are all variations on counting the number of steps needed to edit one set of rankings into the other. We will deal with ordinal data as interval data. For this we follow the rule:

- Convert the ordinal value into Ranks ranging from 1 to R

- Normalize the rank into standardized value of zero to 1 by using the formula

$$x = \frac{r-1}{R-1}$$

- Compute the dissimilarity using methods for interval-scaled variables

## 3.3 Partitioning Around Medoids (k-medoids)

PAM(Partitioning Around Medoids) was developed by Kaufman and Housseeuw. To find k clusters, PAM's approach is to determine a representative object for each cluster. This representative object, called a medoid, is meant to be the most centrally located object within the cluster. Once the medoids have been selected, each non-medoid is grouped with the medoid to which it is the most similar.

More precisely, if $O_j$ is a non-medoid, and $O_i$ is a selected medoid then $O_j$ belongs to the cluster represented by $O_i$. The notation $d(O_a, O_b)$ denotes the dissimilarity distance between objects $O_a$ and $O_b$ . Finally, the quality of a clustering (the combined quality of the chosen medoids) is measured by the average dissimilarity between an object and the medoid of its cluster.

To find the k medoids, PAM begins with an arbitrary selection of objects. Then in each step, a swap between the medoid $O_i$ and a non-medoid object $O_h$ is made, as long as such a swap would result in an improvement of the quality of the clustering. In particular, to calculate effect of such a swap between $O_i$ and $O_h$, PAM computes cost $C_{ih}^j$ for all non-medoid objects $O_j$.Depending on which of the following cases $O_j$ is in, $C_{ih}^j$ is defined by one of the equations below.

***First Case:*** Suppose $O_j$ currently belongs to the cluster represented by $O_i$. Furthermore, let $O_j$ be more similar to $O_{j,2}$ than $O_h$ i.e.$d(O_j, O_h) \geq d(O_j, O_{j,2})$, where $O_{j,2}$ is the second most similar medoid to $O_j$. Thus, if $O_i$ is replaced by $O_h$ as a medoid $O_j$, would belong to the cluster represented by $O_{j,2}$ . Hence, the cost of the swap as far as $O_j$ is concerned is:

$$C_{ih}^j = d(O_j, O_{j,2}) - d(O_j, O_i) \tag{3.1}$$

This equation always gives a non-negative $C_{ih}^j$, indicating that there is a non-negative cost obtained in replacing $O_i$ with $O_h$.

***Second Case:***$O_j$ currently belongs to the cluster represented by $O_i$. But this time, $O_j$ is

less similar to $O_{j,2}$ than $O_h$, i.e.$d(O_j, O_h) < d(O_j, O_{j,2})$. Then, if $O_i$ is replaced by $O_h, O_j$, would belong to the cluster represented by $O_h$. Thus the cost for $O_j$ is given by,

$$C_{ih}^j = d(O_j, O_h) - d(O_j, O_i) \tag{3.2}$$

***Third Case:***Suppose that $O_j$ currently belongs to a cluster other than the one represented by $O_i$. Let $O_{j,2}$ represent that cluster. Furthermore, let $O_j$ be more similar to $O_{j,2}$ than $O_h$. Then even if $O_i$ is replaced by $O_h, O_j$ would stay in the cluster represented by $O_{j,2}$. Thus, the cost is:

$$C_{ih}^j = 0 \tag{3.3}$$

***Fourth Case:***$O_j$ currently belongs to the cluster represented by $O_{j,2}$. But $O_j$ is less similar to $O_{j,2}$ than $O_h$. Thus replacing $O_i$ with $O_h$ would cause $O_j$ to jump to the cluster of $O_h$ from that of $O_{j,2}$ Thus, the cost is:

$$C_{ih}^j = d(O_j, O_h) - d(O_j, O_{j,2}) \tag{3.4}$$

is always negative. Combining the four cases above, the total cost of replacing $O_i$ with $O_h$ is given by:

$$TC_{ih} = C_{ih}^j \tag{3.5}$$

***Algorithm of PAM***

1. Select k representative objects arbitrarily
2. Compute $TC_{ih}$ for all pairs of objects $O_i, O_h$ where $O_i$ is currently selected, and $O_h$ is not
3. Select the pair $O_i, O_h$ which corresponds to $\min_{O_i,O_h} TC_{ih}$. If the minimum $TC_{ih}$ is negative, replace $O_i$ with $O_h$, and go back to Step(2).
4. Otherwise, for each non-selected object, find the most similar representative object. Halt

## 3.4 Average Silhouette Width and Elbow Method

### 3.4.1 Silhouette Coefficient

When there is no information on the class of the objects in a data, the silhouette coefficient can be used to know the quality of the clusters obtained. The coefficient is based on the cohesion and separation of individual points and clusters. In order to construct silhouettes, two things are needed: the partition obtained (by application of some clustering technique) and the collection of all proximities between objects. For each object I, a silhouette value S(i) is introduced.

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{3.6}$$

a(i) is average distance between the object i and its co-members in its cluster

b(i) is the minimum of average distances computed over all others clusters excluding its parent cluster

When cluster with object i contains only a single object it is unclear how a(i) should be defined and then s(i)=0. This choice is of course arbitrary, but a value of zero appears to be most neutral. Indeed, from the above definition it can be seen that

$$-1 \leq s(i) \leq 1, \text{for each object i}$$

Here we need the following summary measures for elbow method

The average of the S(i) for all objects i in a cluster, which is called the average silhouette width of that cluster.

The average of the S(i) for i = 1,2,...,N which is called the average silhouette width for the entire data set, denoted by $\check{S}$(k)

### 3.4.2 Finding Optimal Number of clusters

In order to find optimal number of clusters, we look at the average silhouette width $\check{S}$(k) as a function of number of clusters. One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. More precisely, if one plots the average silhouette width $\check{S}$(k) against the number of clusters, the first clusters will add much information but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion". This "elbow" cannot always be unambiguously identified.

We select the k for which average silhouette width $\check{S}(k)$ is as high as possible.

## 3.5 Clustering of Longitudinal Data

Panel data is a complex data construction format, so before in-depth analysis, it is necessary to do a pretreatment to panel data, which can help us to understand the data format and descriptive statistical character and to obtain some useful information. The structure of multivariable panel data is more complex and getting distance between objects is a tricky part. Here, in our study the clustering is done based on the subjects and hence the time points needs to handled.

The cluster analysis of multivariable panel data is comparatively complex. Presently, there is no relevant software to use directly, which is an important reason why panel data seldom be used in the multivariable statistics research. If not much strictness is there on working, we can adopt an idea of retrogression. Every index will be averaged in the time dimensionality, and abstracted as a case in some special time. So the time dimensionality is obliterated, and the panel data will become section data as retrogression. But this process has problem of losing lot of information and in recessive hypothesis every same index of all objects changes along same direction. Hence, inaccurate or wrong conclusion will appear. But if strictness is there, we need to reconstructs the distance function and the clustering algorithm. Now we will presents a panel data clustering analysis method.

Supposing that the number of the objects in the collectivity is N, the character of every object is denoted by p indexes$(X_1, X_2, ..., X_p)$, the length is T. So $X_{ij}(t)$is the value of index j of object i at the time t. To bring a simple structure of kinds from a group of complex data, it is necessary to measure the proximity. When these objects are aggregated, 'approach'can be depicted by some distance. The distance between object r and object k in the collectivity can be marked as $d_{rk}$, and $d_{rk}$ should satisfy some conditions as follows.

1. $d_{rk} \leq 0, if f X_r = X_k, then d_{rk} = 0$
2. $d_{rk} = d_{kr}, X_r, X_k$
3. $d_{rk} \geq d_{rj} + d_{kj}, X_r, X_k, X_j$

Familiar distance function includes Block distance, Euclidean distance, Minkowski distance, Chebychev distance, Mahalanobis distance, and so on. In our study, we choose Euclidean distance to describe the proximity degree between objects. Certainly, the distance

function of multivariable panel data added time dimensionality is different from the distance function of section data. Here, it is new called 'Euclidean distance timed and spaced', the 'Euclidean distance timed and spaced' between object r and object k is marked as $d_{rk}$, which is defined as follows:

$$d_{rk} = \left\{ \sum_{i=1}^{T} \sum_{j=1}^{P} [X_{rj}(t) - X_{kj}(t)]^2 \right\}^{\frac{1}{2}}$$

Then, the distance of all objects between two will form into a distance matrix, apparently, this matrix is symmetrical, and its diagonal elements are all zero. Here, it is marked as a lower triangular matrix shown as follows.

$$\begin{bmatrix} 0 & & & \\ d_{21} & 0 & & \\ d_{31} & d_{32} & \ddots & \\ \cdots & \cdots & \cdots & \ddots \\ d_{N1} & d_{N2} & \cdots & 0 \end{bmatrix}$$

The process that we follow in our study is:

1. We calculate distance between objects based on Euclidean distance as explained above

   **Note:** Compared to the k-means approach, the function pam has the following features: (a) it also accepts a dissimilarity matrix; (b) it is more robust because it minimizes a sum of dissimilarities instead of a sum of squared euclidean distances; (c) it provides a novel graphical display, the silhouette plot (d) it allows to select the number of clusters.

2. Looking at the above advantages, PAM is employed as the clustering technique as we can employ it for data matrix.

3. We look at silhouette plot to find the optimal number of clusters.

4. For a fixed k, we find the clusters and merge it with our dataset

# Chapter 4

# Classification

## 4.1 Introduction

Classification is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels. Classification techniques are most suited for predicting or describing data sets with binary or nominal categories. Such analysis can help provide us with a better understanding of the data at large. Many classification methods have been proposed by researchers in machine learning, pattern recognition, and statistics. Classification has numerous applications, including fraud detection, target marketing, performance prediction, manufacturing, and medical diagnosis.

Data classification is a two-step process, consisting of a learning step (where a classification model is constructed) and a classification step (where the model is used to predict class labels for given data). In the first step, a classifier is built describing a predetermined set of data classes or concepts. This is the learning step (or training phase), where a classification algorithm builds the classifier by analyzing or "learning from" a training set made up of data set tuples and their associated class labels. Each tuple, X, is assumed to belong to a predefined class as determined by another database attribute called the class label attribute. The class label attribute is discrete-valued and unordered. It is categorical (or nominal) in that each value serves as a category or class. In the context of classification, data tuples can be referred to as samples, examples, instances, data points, or objects.

## 4.2 Support Vector Machines for Classification

Support Vector Machines (SVMs), a method for the classification of both linear and nonlinear data. In a nutshell, an SVM is an algorithm that works as follows. It uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane (i.e., a "decision boundary" separating the tuples of one class from another). With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane. The SVM finds this hyperplane using support vectors ("essential" training tuples) and margins (defined by the support vectors).

They are much less prone to overfitting than other methods. The support vectors found also provide a compact description of the learned model. SVMs can be used for numeric prediction as well as classification. They have been applied to a number of areas, including handwritten digit recognition, object recognition, and speaker identification, as well as benchmark time-series prediction tests.

### 4.2.1 Linearly Separable Case

We consider a simple case of two class problem where classes are linearly separable. Let the dataset D be given as $(X_1, y_1), (X_2, y_2), \ldots, (X_N, y_N)$,where $X_i$is the set of training tuples with associated labels $y_i$.Each $y_i$ can take one of two values, either +1 or -1, corresponding to classes respectively. We use the word linearly separable because a straight line can be drawn to separate all the tuples of class +1 from all the tuples of class -1.

There are an infinite number of separating lines that could be drawn. We want to find the "best" one that will have the minimum classification error. Generalizing to n dimensions, we want to find the best hyperplane. We will use "hyperplane" to refer to the decision boundary that we are seeking, regardless of the number of input attributes. An SVM approaches this problem by searching for the maximum marginal hyperplane. Intuitively, however, we expect the hyperplane with the larger margin to be more accurate at classifying future data tuples than the hyperplane with the smaller margin. This is why (during the learning or training phase) the SVM searches for the hyperplane with the largest margin, that is, the maximum marginal hyperplane (MMH). The associated margin gives the largest separation between classes.

Getting to an informal definition of margin, we can say that the shortest distance from a hyperplane to one side of its margin is equal to the shortest distance from the hyperplane

to the other side of its margin, where the "sides" of the margin are parallel to the hyperplane. Any training tuples that fall on hyperplanes H1 or H2 (i.e., the "sides" defining the margin) and are called support vectors. The complexity of the learned classifier is characterized by the number of support vectors rather than the dimensionality of the data. Hence, SVMs tend to be less prone to overfitting than some other methods. The support vectors are the essential or critical training tuples they lie closest to the decision boundary (MMH). If all other training tuples were removed and training were repeated, the same separating hyperplane would be found. Furthermore, the number of support vectors found can be used to compute an (upper) bound on the expected error rate of the SVM classifier, which is independent of the data dimensionality.An SVM with a small number of support vectors can have good generalization, even when the dimensionality of the data is high.

### 4.2.2 Linearly Inseparable Case

In cases where no straight line can be found that would separate the classes, linear SVMs can be extended to create nonlinear SVMs for the classification of linearly inseparable data (also called nonlinearly separable data). Such SVMs are capable of finding nonlinear decision boundaries (i.e., nonlinear hypersurfaces). We obtain a nonlinear SVM by extending the approach for linear SVMs as follows. There are two main steps:

- In the first step, we transform the original input data into a higher dimensional space using a nonlinear mapping. Several common nonlinear mappings can be used in this step, as we will further describe next.
- Once the data have been transformed into the new higher space, the second step searches for a linear separating hyperplane in the new space.

We transform each observation,$X_1 \in R^p$ using some non-linear mapping

$$\phi : R^p \rightarrow H$$

where $H$ is an $N_h$ dimensional space. The space $H$ may be high dimensional.
Let $\phi(X_i) = (\phi(X_i), \ldots, \phi_{N_h}(X_i))^T \in H, i = 1, 2, \ldots, N$
The transformed sample is then$\{\phi(X_i), y_i\}$,where $y_i \in \{-1, 1\}$

It so happens that in solving the quadratic optimization problem of the linear SVM (i.e., when searching for a linear SVM in the new higher dimensional space), the training tuples appear only in the form of dot products $\phi(X_i).\phi(X_j)$.Instead of computing the dot

product on the transformed data tuples, it turns out that it is mathematically equivalent to instead apply a kernel function,$K(X_i, X_j) = \phi(X_i).\phi(X_j)$to the original input data. In other words, everywhere that$\phi(X_i).\phi(X_j)$ appears in the training algorithm, we can replace it with $K(X_i, X_j)$.In this way, all calculations are made in the original input space, which is of potentially much lower dimensionality. After applying this trick, we can then proceed to find a maximal separating hyperplane. Three admissible kernel functions are:

**Polynomial kernel of degree h:**$K(X_i, X_j) = (X_i.X_j + 1)^h$

**Gaussian Radial Basis function kernel:**$K(X_i, X_j) = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}}$

**Sigmoid kernal:**$K(X_i, X_j) = \tanh(kX_i.X_j - \delta)$

Each of these results in a different nonlinear classifier in(the original) input space. There are no golden rules for determining which admissible kernel will result in the most accurate SVM. We explained both linear and non-linear SVMs for binary class. This can be extended to multiclass case also.

**Multiclass SVM**

- **One versus rest:** Divide the k class problem into binary classification subproblems of the type '$k^{th}$ class' versus 'not $k^{th}$ class',k=1,2,...,K. Corresponding to the $k^{th}$ sub problem,a classifier is constructed in which the$k^{th}$ class is coded as positive and the union of other classes is coded as negative. A new object is assigned to the class with largest value of

  $$\hat{f}_k(x_k), k = 1, 2, \ldots, K$$

  where $f_k(x_k)$ is the optimal SVM solution for the binary problem of the $k^{th}$ class.

- **One-versus-one:**Divide the k-class problem into $\binom{k}{2}$comparisons of all pairs of classes. A classifier$\hat{f_{ij}}(x)$ is constructed on coding the $j^{th}$ class as positive and $k^{th}$ class as negative,$j, k = 1, 2, \ldots, K; j \neq$ k.Then for a new x, aggregate the votes for each class and assign x to the class having the most votes.

## 4.3 Decision Tree Induction for Classification

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flowchart-like tree structure, where each internal node (non-leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and

each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node. Some decision tree algorithms produce only binary trees (where each internal node branches to exactly two other nodes), whereas others can produce non-binary trees.

Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision trees can easily be converted to classification rules. The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle multidimensional data. Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans. The learning and classification steps of decision tree induction are simple and fast. In general, decision tree classifiers have good accuracy. However, successful use may depend on the data at hand. Decision tree induction algorithms have been used for classification in many application areas such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology.

***Algorithm to generate decision tree***
***Input:***

- Data partition, D, which is a set of training tuples and their associated class labels
- Attribute list, the set of candidate attributes or variables or features
- Attribute selection method, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a splitting attribute and, possibly, either a split-point or splitting subset.

***Method:***

1. Create a node N
2. If tuples in D are all of the same class, C, then return N as a leaf node labeled with the class C
3. If attribute list is empty then return N as a leaf node labeled with the majority class in D
4. Apply Attribute selection method(D, attribute list) to find the "best" splitting criterion

5. Label node N with splitting criterion
6. If splitting attribute is discrete-valued and multiway splits allowed. It is not restricted to binary trees
7. Attribute list=Attribute list –splitting attribute
8. For each outcome j of splitting criterion, partition the tuples and grow subtrees for each partition
9. Let $D_j$ be the set of data tuples in D satisfying outcome j
10. If $D_j$ is empty then attach a leaf labeled with the majority class in D to node N
11. Else attach the node returned by Generate decision tree(Dj, attribute list) to node N
12. Return N

### 4.3.1 Attribute Selection Measures

An attribute selection measure is a heuristic for selecting the splitting criterion that "best" separates a given data partition, D, of class-labeled training tuples into individual classes. If we were to split D into smaller partitions according to the outcomes of the splitting criterion, ideally each partition would be pure (i.e., all the tuples that fall into a given partition would belong to the same class). Conceptually, the "best" splitting criterion is the one that most closely results in such a scenario. Attribute selection measures are also known as splitting rules because they determine how the tuples at a given node are to be split. The attribute selection measure provides a ranking for each attribute describing the given training tuples.

The attribute having the best score for the measure is chosen as the splitting attribute for the given tuples. If the splitting attribute is continuous-valued or if we are restricted to binary trees, then, respectively, either a split point or a splitting subset must also be determined as part of the splitting criterion. The tree node created for partition D is labeled with the splitting criterion, branches are grown for each outcome of the criterion, and the tuples are partitioned accordingly. Out of three popular attribute selection measures information gain, gain ratio, and Gini index. Here we are explaining the information gain.

**Information Gain**

This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of messages. Let node N represents or hold the tuples of partition D. The attribute with the highest information gain is chosen as the splitting attribute for node N. This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or "impurity" in these partitions. Such an approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple (but not necessarily the simplest) tree is found. The expected information needed to classify a tuple in D is given by

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(pi)$$

where $p_i$ is the nonzero probability that an arbitrary tuple in D belongs to class $C_i$. Info(D) is also called the entropy of D. Now, suppose we were to partition the tuples in D on some attribute A having $v$ distinct values,$\{a_1, a_2, \ldots, a_v\}$ as observed from the training data.

If A is discrete-valued, these values correspond directly to the v outcomes of a test on A. Attribute A can be used to split D into v partitions or subsets,$\{D_1, D_2, \ldots, D_v\}$ where $D_j$ contains those tuples in D that have outcome $a_j$ of A.These partitions would correspond to the branches grown from node N. Ideally, we would like this partitioning to produce an exact classification of the tuples. That is, we would like for each partition to be pure. However, it is quite likely that the partitions will be impure. Therefore, we need more information and this is measured by

$$Indo_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

Information gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on A). That is,

$$Gain(A) = Info(D) - Info_A(D)$$

It is the expected reduction in the information requirement caused by knowing the value of A. The attribute A with the highest information gain, Gain(A), is chosen as the splitting attribute at node N. This is equivalent to saying that we want to partition on the attribute A that would do the "best classification", so that the amount of information still

required to finish classifying the tuples is minimal (i.e., minimum $Info_A(D)$)

For a continuous valued data, we must determine the "best" split-point for A, where the split-point is a threshold on A. We first sort the values of A in increasing order. Typically, the midpoint between each pair of adjacent values is considered as a possible split-point. Therefore, given $v$ values of $A$, $v-1$ possible splits are evaluated. For example, the midpoint between the values $a_i$ is $(a_i + a_{i+1})/2$. For each possible split-point for A, we evaluate $Info_A(D)$, where the number of partitions is two, that is, v=2 (or j=1,2). The point with the minimum expected information requirement for A is selected as the split point for A. $D_1$ is the set of tuples in D satisfying $A \leq$ split point, and $D_2$ is the set of tuples in D satisfying $A >$ split point.

## 4.4 Building Classifier

In our study, we use longitudinal data but since we did clustering on the basis of subjects or units and labels are assigned to subjects irrespective of the year to which the observation belongs. Hence we build classifier on the basis of features irrespective of the year in which observations were obtained. The procedure to be followed in our study is explained below:

1. Based on the clusters obtained in the previous chapter, we build classifier on the basis of these labels

2. We measure the performance of the classifiers based on the measure given in the next chapter

3. Finally for the test set we get the classification number to select the sub-model

4. Hence, we reach the end of our study based on the prediction obtained from the sub-model selected

# Chapter 5

# Validation and Model Adequacy Checking

## 5.1 Introduction

We distinguish between model adequacy checking and model validation. Model adequacy checking includes residual analysis, testing for lack of fit, searching for high - leverage or overly influential observations, and other internal analyses that investigate the fit of the regression model to the available data. Model validation is the task of confirming that the outputs of a statistical model are acceptable with respect to the real data-generating process. In other words, model validation is the task of confirming that the outputs of a statistical model have enough fidelity to the outputs of the data-generating process that the objectives of the investigation can be achieved. Model validation can be based on two types of data: data that was used in the construction of the model and data that was not used in the construction.

Validation based on the first type usually involves analyzing the validation measure on the training set. Validation based on the second type usually involves analyzing whether the model's predictive performance deteriorates non-negligibly when applied to new data. Ideally, if we had enough data, we would set aside a validation set and use it to assess the performance of our prediction model. Since data are often scarce, this is usually not possible. Here, we intend to do model validation for both regression and classification.

## 5.2 Model Adequacy Checking

In regression analysis, we make assumptions to estimate our parameters and build a statistical significant model. We should always consider the validity of these assumptions to be doubtful and conduct analyses to examine the adequacy of the model we have tentatively entertained. Gross violations of the assumptions may yield an unstable model in the sense that a different sample could lead to a totally different model with opposite conclusions. We usually cannot detect departures from the underlying assumptions by examination of the standard summary statistics, such as the t or F statistics, or $R^2$.These are "global" model properties, and as such they do not ensure model adequacy.

### 5.2.1 Residual Plots

Graphical analysis of residuals is a very effective way to investigate the adequacy of the fit of a regression model and to check the underlying assumptions. They should be examined routinely in all regression modeling problems. We often plot externally studentized residuals because they have constant variance.

**Normal Probability Plot:** Small departures from the normality assumption do not affect the model greatly, but gross non-normality is potentially more serious as the t or F statistics and confidence and prediction intervals depend on the normality assumption. Furthermore, if the errors come from a distribution with thicker or heavier tails than the normal, the least - squares fit may be sensitive to a small subset of the data. Heavy - tailed error distributions often generate outlier that "pull" the least - squares fit too much in their direction. A very simple method of checking the normality assumption is to construct a normal probability plot of the residuals. This is a graph designed so that the cumulative normal distribution will plot as a straight line.Let $t_{[1]} < t_{[2]} < \cdots < t_{[n]}$ be the externally studentized residuals ranked in increasing order. If we plot $t_{[i]}$ against the cumulative probability $P_i = (i - 0.5)/n, i = 1, 2, \ldots, n$,on the normal probability plot, the resulting points should lie approximately on a straight line.

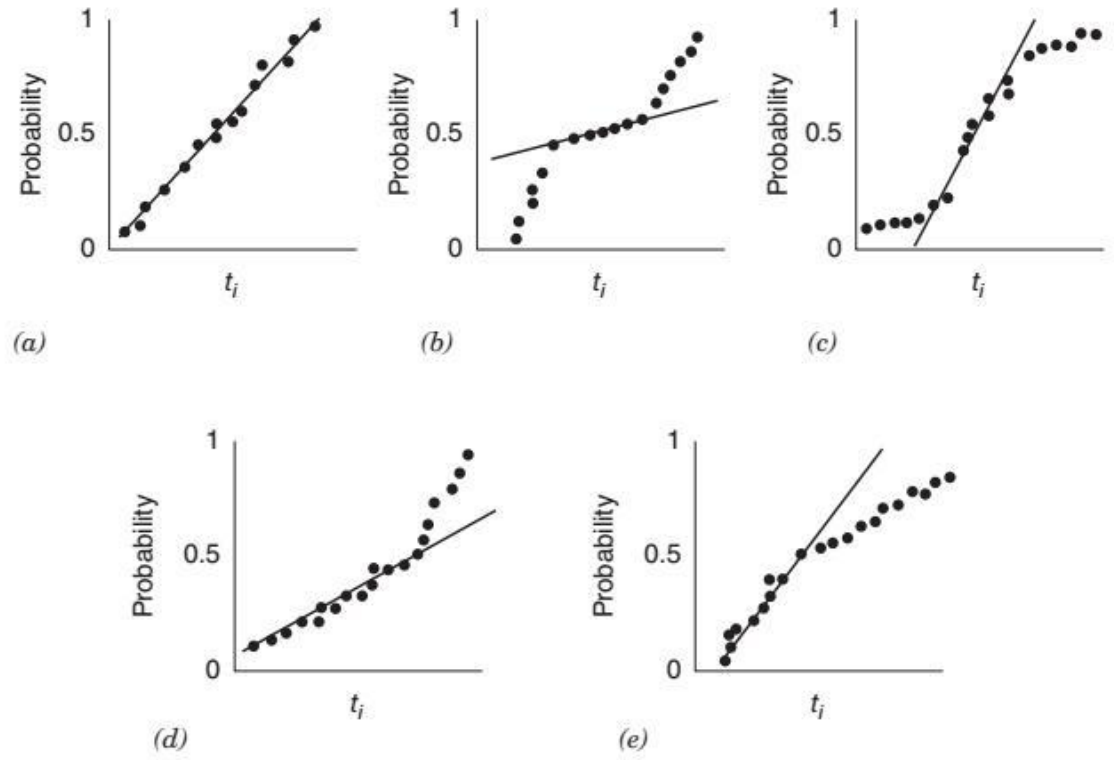



Figure 5.1: Normal probability plots: (a) ideal (b) light-tailed distribution (c) heavy-tailed distribution (d) positive skew (e) negative skew

**Another alternative to this is we plot histogram for raw residuals to look at the shape of the distribution of the residuals.**

## 5.3 Validity Measures

Once we build regression model or classifier, we would like to estimate of how accurately the regression model or classifier can predict the future values on which our model has not been trained. In order to estimate the performance of the model in question, we need some validity measures to evaluate our models or compare our models. Here we list two validity measures-one for regression and another for classifier.

### 5.3.1 Mean Squared Error

Mean Squared Error (MSE) or Mean Squared Deviation (MSD) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value.

If a vector of n predictions generated from a sample of n data points on all variables, and $Y_i$ is the vector of observed values of the variable being predicted, with $Y_i$ being the predicted values, then the within-sample MSE of the predictor is computed as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

When used for cross-validation, it is also called mean squared prediction error. The MSE is a measure of the quality of an estimator, it is always non-negative, and values closer to zero are better.

### 5.3.2 Confusion Matrix and Accuracy Score

A confusion matrix of size n x n associated with a classifier shows the predicted and actual classification, where n is the number of different classes.

| | Predicted Negativity | Predicted Positivity |
|---|---|---|
| Actual Negative | a | b |
| Actual Positive | c | d |

Here,

- a represents the number of correct negative predictions
- b is the number of incorrect positive predictions
- c is the number of incorrect negative predictions
- d is the number of correct positive predictions

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

## 5.4 Validation for Regression Model

Ideally, if we had enough data, we would set aside a validation set and use it to assess the performance of our prediction model. Since data are often scarce, this is usually not possible. To finesse the problem, K-fold cross-validation uses part of the available data to

fit the model, and a different part to test it. We split the data into K roughly equal-sized parts. For the $k^{th}$ part, we fit the model to the other K-1 parts of the data, and calculate the prediction error of the fitted model when predicting the $k^{th}$ part of the data. We do this for $k = 1, 2, \ldots, K$ and combine the K estimates of prediction error.Two of the most common types of cross-validation are k-fold cross-validation and hold-out cross-validation.

However, when it comes to time series prediction, practitioners are often unsure of the best way to evaluate their models. There is often a feeling that we should not be using future data to predict the past. In addition, the serial correlation in the data, along with possible non-stationarities, makes the use of CV appear problematic as it does not account for these issues. Usually, practitioners resort to usual out-of-sample (OOS) evaluation instead, where a section from the end of the series is withheld for evaluation. However, in this way, the benefits of CV, especially for small datasets, cannot be exploited.

***Reasons for cross-validation being different with time series:***

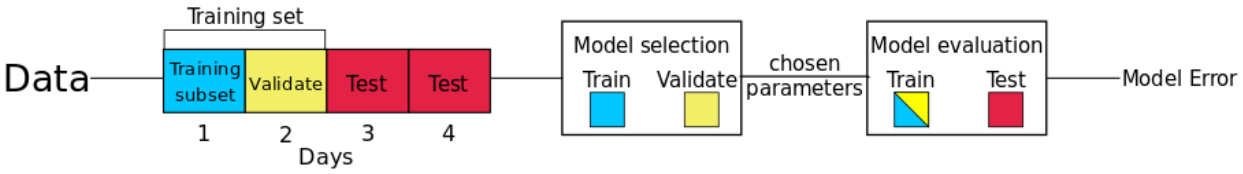
1. ***Temporal Dependencies:*** With time series data, particular care must be taken in splitting the data in order to prevent data leakage. So, rather than use k-fold cross-validation, for time series data we utilize hold-out cross-validation where a subset of the data (split temporally) is reserved for validating the model performance. We know that where the test set data comes chronologically after the training set. Similarly, the validation set comes chronologically after the training subset.

2. ***Arbitrary Choice of test set:*** Usually the choice of our test is arbitrary but in our study the test set cannot be arbitrarily chosen. Since we cannot use the present to predict the part. So arbitrary choice of test set or even validation set in training set is not possible. To address this, we use a method called Nested Cross-Validation

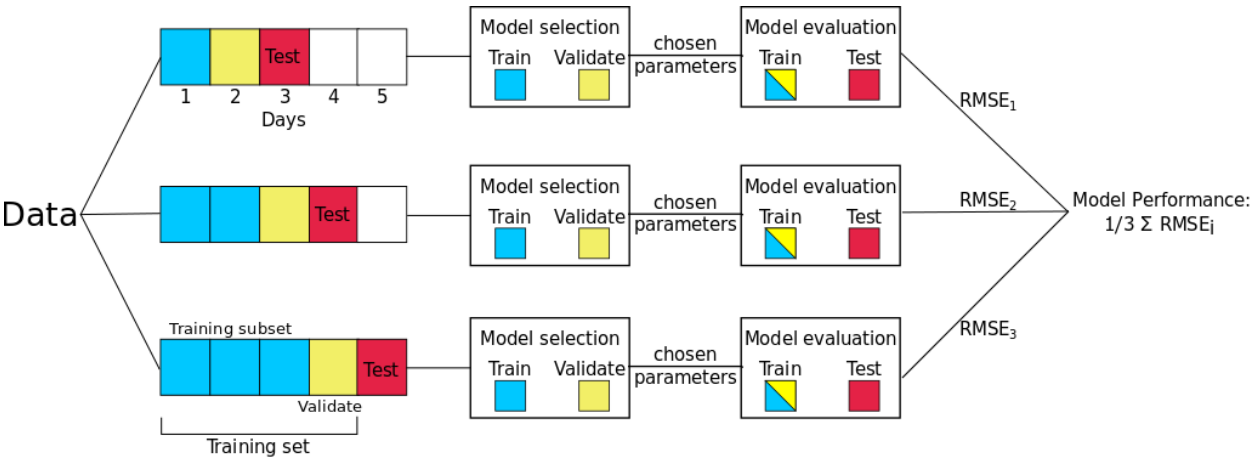
### 5.4.1 Nested Cross-Validation for a single Time Series

We suggest two methods for nested CV with data from a single time series:

1. **Predict Second Half:** It is the "base case" of nested CV with only 1 *train/test* split. The advantage to this is that this method is easy to implement; however, it still suffers from the limitation of an arbitrarily-chosen test set. The first half of data (split temporally) is assigned to the training set and the latter half becomes the test set. The validation set size can vary based on the given problem, but it is important to ensure that the validation set is chronologically subsequent to the training subset.




2. **Day Forward-Chaining:** One shortcoming of the Predict Second Half nested cross-validation method is that the arbitrary choice of the hold-out test set can produce biased estimates of prediction error on an independent test set. In order to produce a better estimate of model prediction error, a common approach is to create many *train/test* splits and average the errors over all the splits. The technique we use, called Day Forward-Chaining is based on a method called forward-chaining (also referred to in the literature as rolling-origin evaluation and rolling-origin-recalibration evaluation). Using this method, we successively consider each day as the test set and assign all previous data into the training set. This method produces many different *train/test* splits and the error on each split is averaged in order to compute a robust estimate of the model error.




### 5.4.2 Nested Cross-Validation with Multiple Time Series

Now that we have two methods for splitting a single time series, we discuss how to handle a dataset with multiple different time series. Again, we use two types:

1. **Regular:** For "regular" nested cross-validation, the basic idea of how the *train/validation/test* splits are made is the same as before. The only change is that the splits now contain data from each subject in our dataset. For instance, if there are two participants, Participant A and B, the training set would contain the data from the first half of days from Participant A and the data from the first half of days from Participant B. Likewise, the testing set would contain the second half of days for each participant.

2. **Population-Informed:** For "population-informed nested cross-validation" we take advantage of the independence between different participants' data. This allows us to break the strict temporal ordering, at least between individuals' data (it is still necessary within an individual's data). Because of this independence, we can slightly modify the regular nested cross-validation algorithm. Now the test and validation sets only contain data from one participant, say Participant A, and all data from all other participants in the dataset are allowed in the training set. The Participant A's 18th day is the testing set (colored red), the previous three days are the validation set (colored yellow), and the training set (colored green) contains all the previous data from Participant A, as well as all the data from the rest of the participants. The important thing to emphasize is that there is no data leakage from using "future" observations from the other participants precisely due to the independence of these participants' time series.

Here in our study we are going to use Regular Predict Second Half and Regular Day Forward Chaining Validation technique to validate our regression model.

## 5.5 Validation for Classification model

In our classification technique, the clusters are allocated irrespective of years so we can take the liberty to treat the classification on the data as if from a cross sectional data pattern. For achieving a particular accuracy level we will validate our model using the k-fold cross-validation technique. K-fold cross-validation uses part of the available data to fit the model, and a different part to test it. We split the data into K roughly equal-sized parts. For the $k^{th}$ part, we fit the model to the other K-1 parts of the data, and calculate the accuracy score of the fitted model when predicting the $k^{th}$ part of the data. We do this for$k = 1, 2, \ldots, K$ and combine the K estimates of accuracy score. Typical choices of K are 5 or 10 (see below). The case K = N is known as leave-one-out cross-validation. In this case $k(i)$, and for the $i^{th}$ observation the fit is computed using all the data except the $i^{th}$, With $K = N$, the cross-validation estimator is approximately unbiased for the true (expected) prediction error, but can have high variance because the N "training sets" are so similar to one another. The computational burden is also considerable, requiring N applications of the learning method. Here is the correct way to carry out cross-validation in this example:

1. Divide the samples into K cross-validation folds (groups) at random.

2. For each fold $k = 1, 2, \ldots, K$

   (a) Find a subset of "good" predictors that show fairly strong (univariate) correlation with the class labels, using all of the samples except those in fold k.

   (b) Using just this subset of predictors, build a multivariate classifier, using all of the samples except those in fold k.

   (c) Use the classifier to predict the class labels for the samples in fold k.

The error estimates from step 2(c) are then accumulated over all K folds, to produce the cross-validation estimate of prediction error.

# Chapter 6

# Real Life Dataset and Analysis

## 6.1 Introduction

In earlier chapters, we have described the procedures and the kind of calculations done in our study. In this chapter, we describe about the data used and analysis done as described in the outline of the first chapter. Our dataset consists of 100 countries as subjects, with initially 14 variables across 10 years' time period (2007-2016). We did not account for missing observations as this would make our calculations more complex. So countries with complete information were taken in our study. The target in our study is to predict $CO_2$ Emissions from the countries in the future years. The models have been so prepared that even predictions can be made for other countries based on some information available initially. Hence, subject based modeling has not been done. In order to test our model for future predictions, the latest two years have been kept as test set and is not involved in training the model.

## 6.2 Variable Description

The following table presents the description about the variables and source from which secondary data was collected

| **Variable Name** | **Units** | **Description** | **File Source** |
|---|---|---|---|
| Population Total (pop_total) | Integer value | Total population is based on the de facto definition of population,which counts all residents regardless of legal status or citizenship.The values shown are midyear estimates. | World Bank Data Repository |

| | | | |
|---|---|---|---|
| Carbon Dioxide Emission (response variable) (co2emission) | Megatonne (Mt) | Territorial Emissions: Carbon dioxide emissions attributed to the country in which they physically occur. The values include:<br>• Coal: Carbon dioxide emissions from the oxidation of coal.<br>• Oil: Carbon dioxide emissions from the oxidation of oil.<br>• Gas: Carbon dioxide emissions from the oxidation of gas.<br>• Gas flaring: Carbon dioxide emissions from the combustion of vented natural gas and the venting of CO2 in the oil and gas industry converting methane into carbon dioxide.<br>• Cement: Carbon dioxide emissions from chemical reactions in the manufacture of cement. | Global Carbon Atlas Cite as: Gilfillan et al. (2019), UNFCCC (2019), BP (2019) |
| Gross National Income per capita (gnipercapita) | 2011 PPP $ | • GNI per capita is gross national income divided by mid-year population.<br>• GNI per capita based on purchasing power parity (PPP).<br>• PPP GNI is gross national income (GNI) converted to international dollars using purchasing power parity rates.<br>• Data are in constant 2011 international dollars. | United Nations Development Programme (UNDP) |
| Access to Electricity (acctoelectricity) | % of population | • Access to electricity is the percentage of population with access to electricity.<br>• Electrification data are collected from industry, national surveys and international sources. | World Bank Data Repository |

| | | | |
|---|---|---|---|
| Inflation, GDP deflator (annual %) (inflationpercent) | Annual % | • Inflation as measured by the annual growth rate of the GDP implicit deflator shows the rate of price change in the economy as a whole.<br>• The GDP implicit deflator is the ratio of GDP in current local currency to GDP in constant local currency. | World Bank Data Repository |
| Fuel Export (fuel_export) | Current US $ | Fuels comprise the commodities in SITC section 3 (mineral fuels, lubricants and related materials). It consists of materials:<br>• Coal, coke and briquettes<br>• Petroleum, petroleum products and related materials<br>• Gas, natural and manufactured<br>**Note:** Here values are obtained by multiplying percentage of total export merchandise with absolute value of total export merchandise | World Bank Data Repository |
| Fuel Import (fuel_import) | Current US $ | Fuels comprise the commodities in SITC section 3 (mineral fuels, lubricants and related materials). It consists of materials:<br>• Coal, coke and briquettes<br>• Petroleum, petroleum products and related materials<br>• Gas, natural and manufactured<br>**Note:** Here values are obtained by multiplying percentage of total import merchandise with absolute value of total import merchandise | World Bank Data Repository |
| Gross Domestic Product (gdp) | Current US$ | GDP at purchaser's prices is the sum of gross value added by all resident producers in the economy plus any product taxes and minus any subsidies not included in the value of the products. | World Bank Data Repository |

| | | | |
|---|---|---|---|
| Forest Area (forest_area) | sq. km. | Forest area is land under natural or planted stands of trees of at least 5 meters in situ, whether productive or not, and excludes tree stands in agricultural production systems (for example, in fruit plantations and agroforestry systems) and trees in urban parks and gardens. | World Bank Data Repository |
| Agricultural land area (agri_area) | sq. km. | Agricultural land refers to the share of land area that is arable, under permanent crops, and under permanent pastures (arable-temporary crops, permanent pastures-used for 5 or more years). | World Bank Data Repository |
| Land Area (land_area) | sq. km. | Land area is a country's total area, excluding area under inland water bodies, national claims to continental shelf, and exclusive economic zones. In most cases the definition of inland water bodies includes major rivers and lakes. | World Bank Data Repository |
| Human Development Index (hdi) | Real number (Value lies between 0-1) | • The Human Development Index (HDI) is a summary measure of average achievement in key dimension of human development: a long and healthy life, being knowledgeable and have a decent standard of living.<br>• The HDI is the geometric mean of normalized indices for each of the three dimensions.<br>• Health dimension includes life expectancy at birth, Education dimension includes mean of years of schooling for adults aged 25 years and more and expected years of schooling for children of school entering age and standard of living dimension is measured by gross national income per capita. | United Nations Development Programme (UNDP) |

| | | | |
|---|---|---|---|
| Temperature Change (temp_change) | °Centigrade | • Observed mean surface temperature changes by country, over the period 1961-2017 with annual updates.<br>• The data provide information on monthly, seasonal and annual mean temperature anomalies, i.e., temperature changes with respect to a baseline period, 1951–1980.<br>• Data are based on GISTEMP, the Global Surface Temperature Change data of the National Aeronautics and Space Administration Goddard Institute for Space Studies (NASA-GISS). | Food and Agriculture Organisation of the United Nations |

## 6.3 Methodology

The methodology followed in our study is explained briefly in the following steps:

***Regression***

- **Step 1:** We partition the data as training set and test set as explained above
- **Step 2:** We build GLM model on the training set
- **Step 3:** Model Adequacy Checking and Cross-Validation is done to check the validity of our model
- **Step 4:** Prediction is done on the test set to get the mean squared prediction error
- **Step 5:** We perform Robust GLM on the same data and follow Step 3 and Step 4
- **Step 6:** Comparison is done between the models based on the mean squared prediction error

***Clustering***

- **Step 1:** Some pre-processing is done on ordinal data and log transformation on the continuous variables. Finally we normalize all the variables.
- **Step 2:** Distance matrix is calculated as explained in Section 3.5

- **Step 3:** PAM is performed on the distance matrix and elbow method conducted as explained in Section 3.4
- **Step 4:** Based on optimal number of clusters, clustering is done on the data
- **Step 5:** Visualization is done after obtaining the clusters

***Classification***

- **Step 1:** Data pre-processing is done to build classifier
- **Step 2:** Classifier is built as explained in Section 4.5
- **Step 3:** k-fold Cross-Validation is done on the various classifiers
- **Step 4:** Prediction is done using the test set
- **Step 5:** Comparison is done between the classifiers to select the best classifier

***Steps performed in our study***

- **Step 1:** A study was made to know about the data using summary statistics
- **Step 2:** Visualization is done on the response variable as explained in Section 2.2
- **Step 3:** Variables are analyzed to drop the insignificant variables which is explained in Section 2.3
- **Step 4:** Regression is done on complete data as mentioned above
- **Step 5:** After clustering is performed, we make sub-models based on the number of clusters. In our case, 3 sub-models are made using the same procedure as explained in Regression section above.
- **Step 6:** Some of the levels are merged in the categorical variables to make predictions using sub-models.

***Prediction***

- **Step 1:** Based on the classifier selected we get cluster number for test set observations
- **Step 2:** Based on the cluster number we predict $CO_2$Emission using that sub-model as well as using the complete model (care is taken in sub-models as categorical variables are imputed to cover missing levels)

- **Step 3:** Based on the prediction we classify the countries on the scale of $CO_2$ Emission and check the historical data of actual values to see the jump in level of $CO_2$ Emissions and in future by the countries.

## 6.4 Visualization

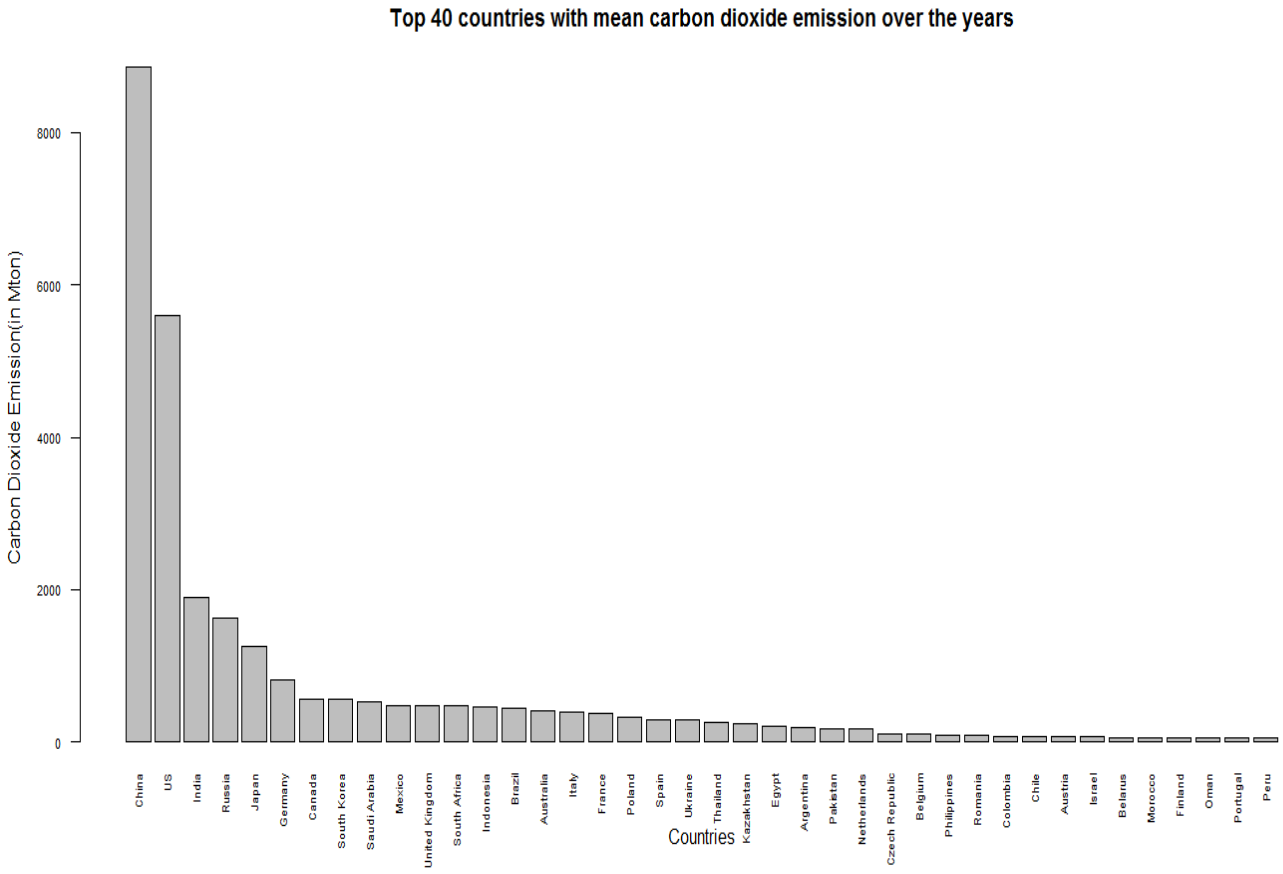



Figure 6.1: Top 40 countries with carbon emission over the years

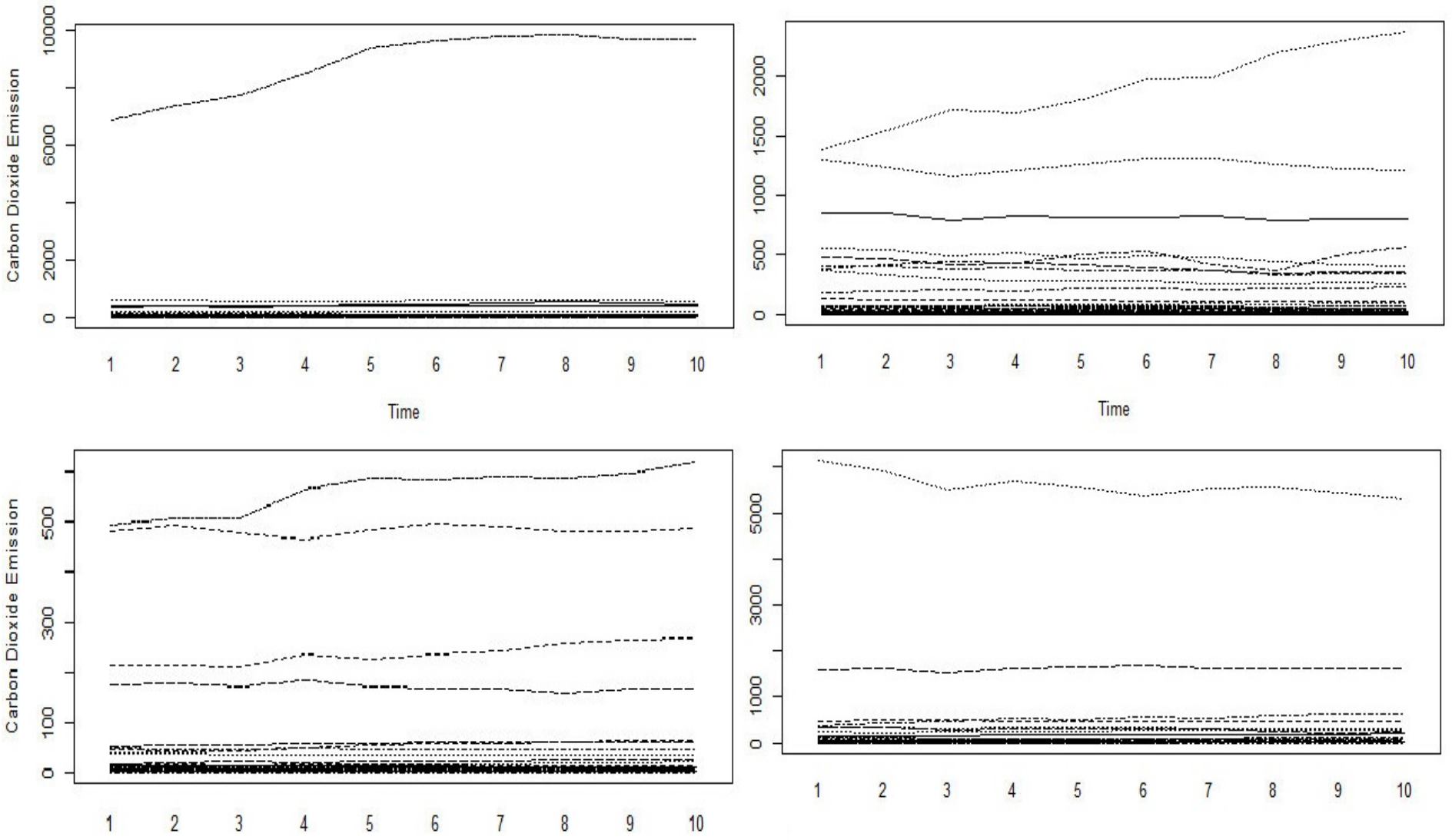



Figure 6.2: Spahgetti Plot for Complete Data (Carbon Emission vs Time each representing 25 countries)

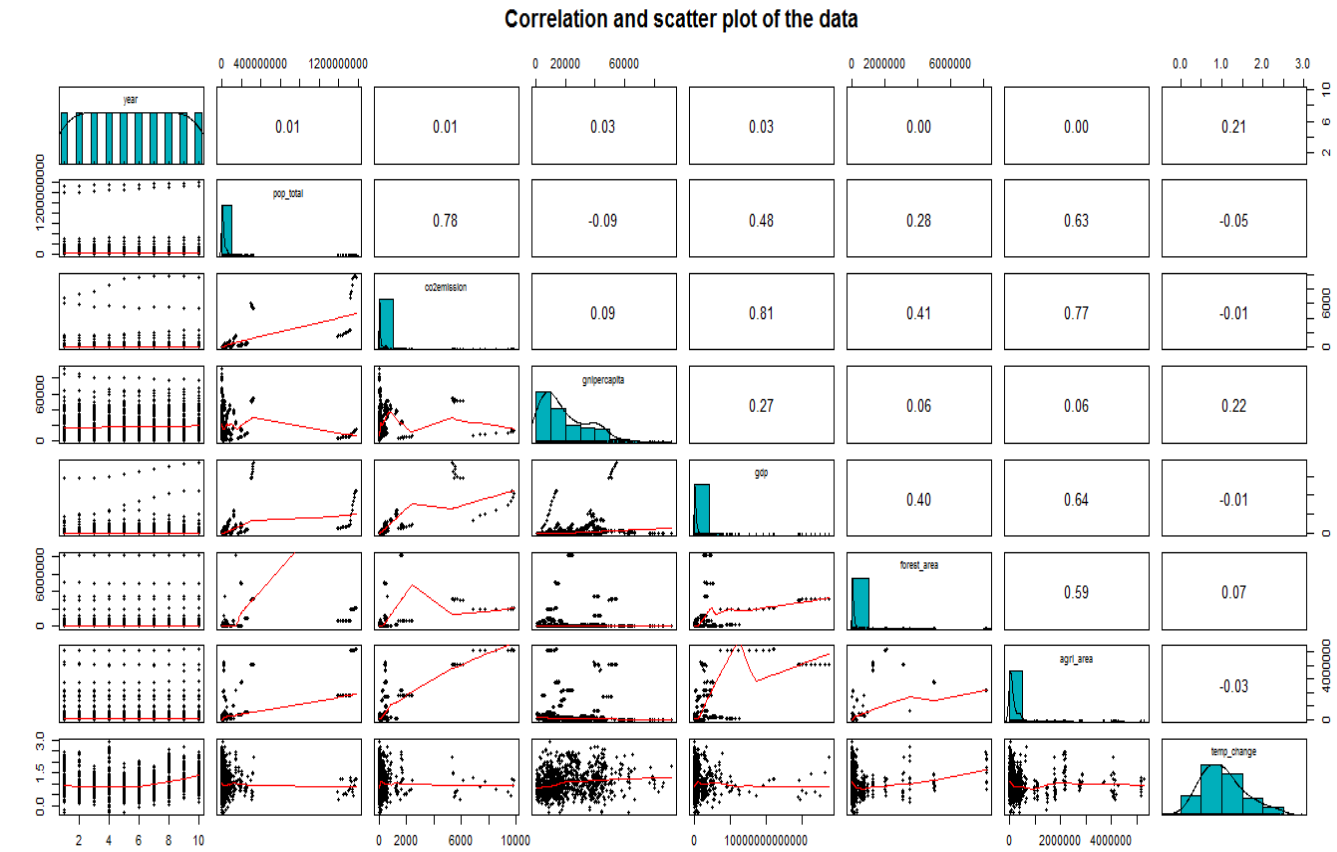



Figure 6.3: Correlation Matrix for variables

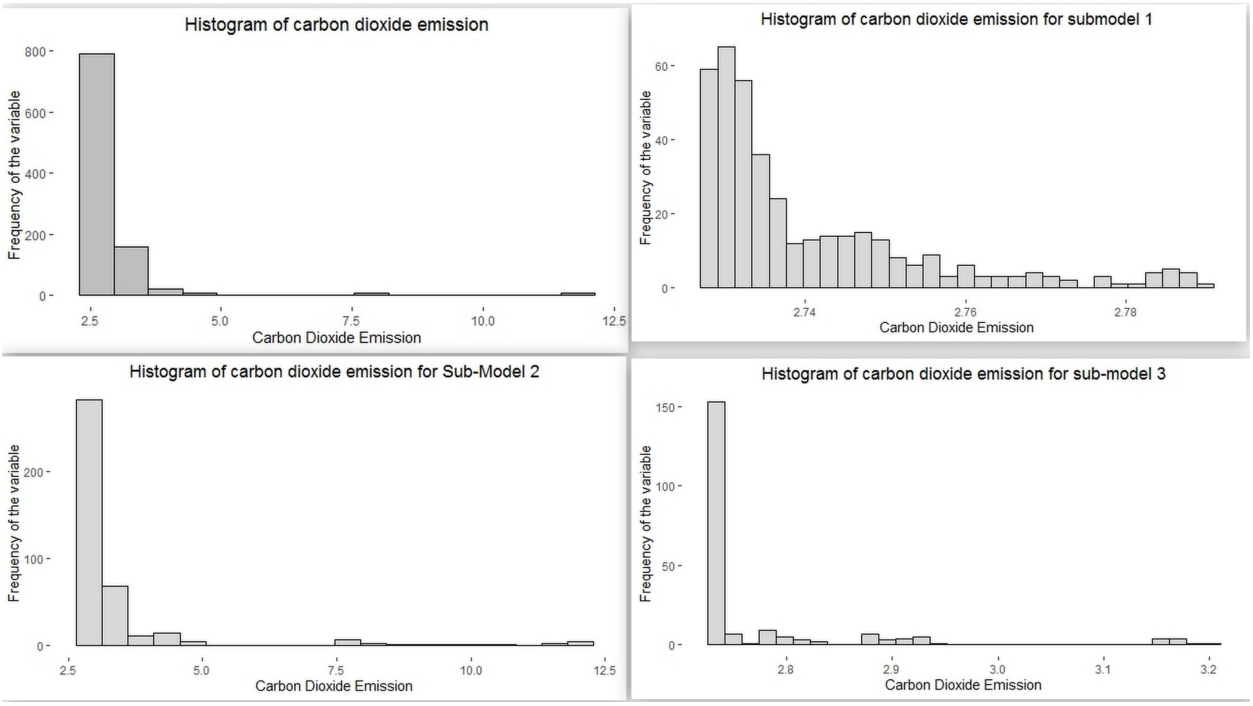



Figure 6.4: Histogram of carbon dioxide emission for Complete data, Cluster 1, Cluster 2, Cluster 3 (clockwise from topleft)

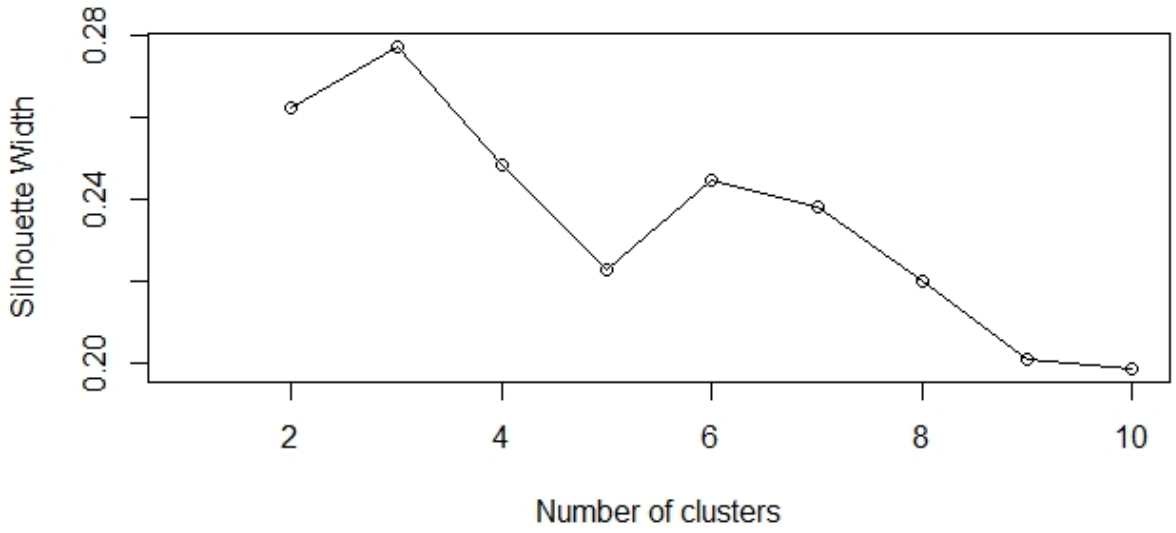



Figure 6.5: Silhouette Plot

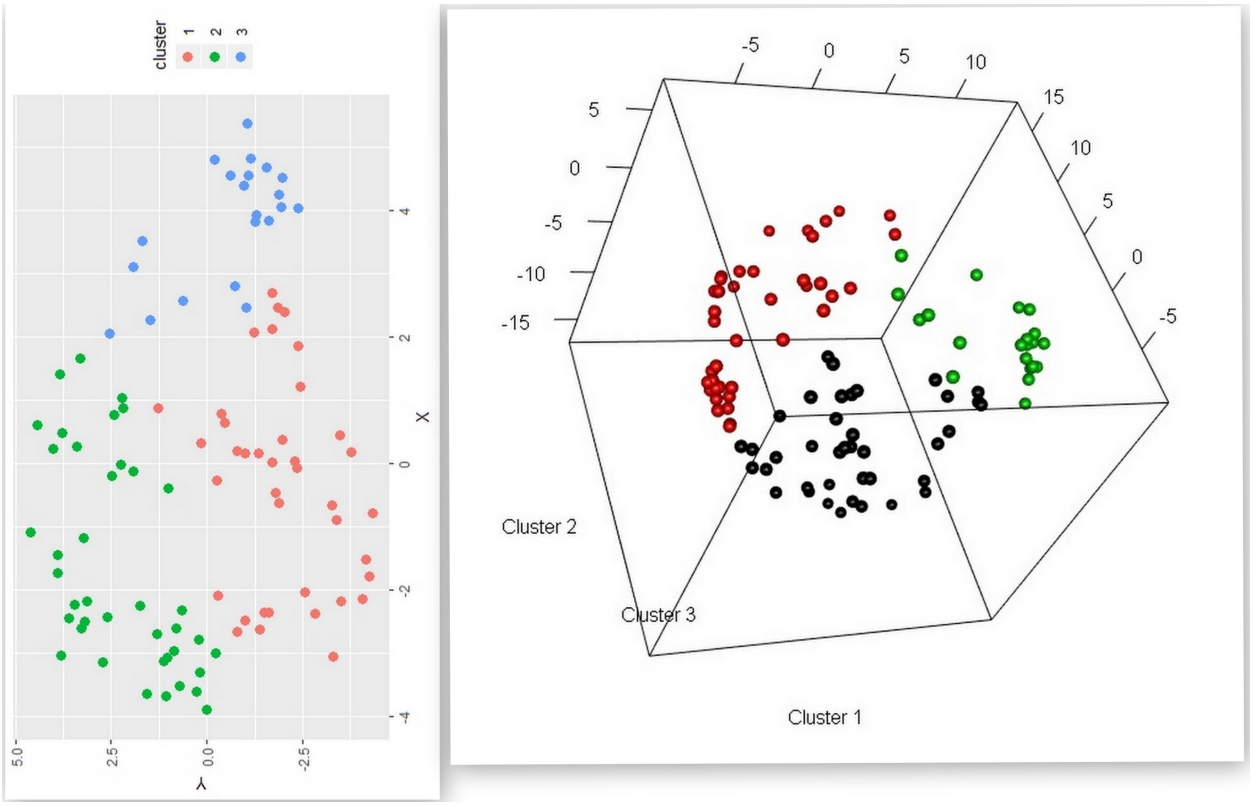



Figure 6.6: Plotting of clusters-2d plot,3d plot

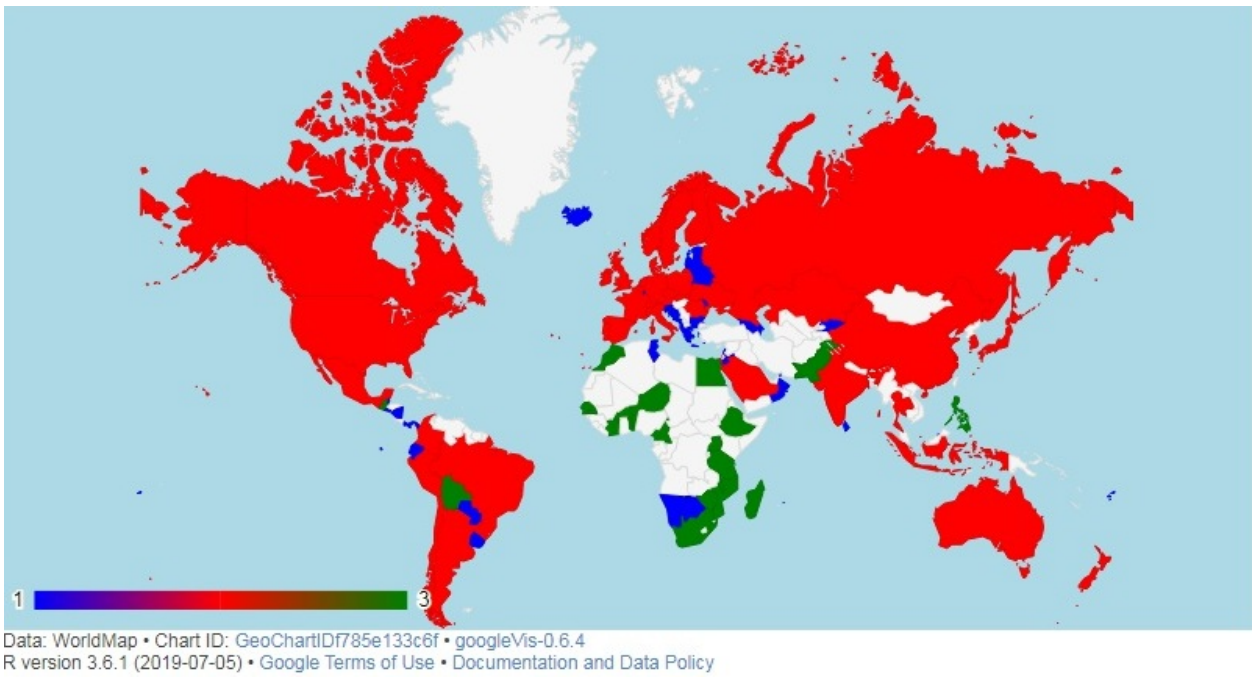



Figure 6.7: Clusters on the World Map. Note-Malta,Bahrain,Barbados,St.Lucia not plotted due to small area

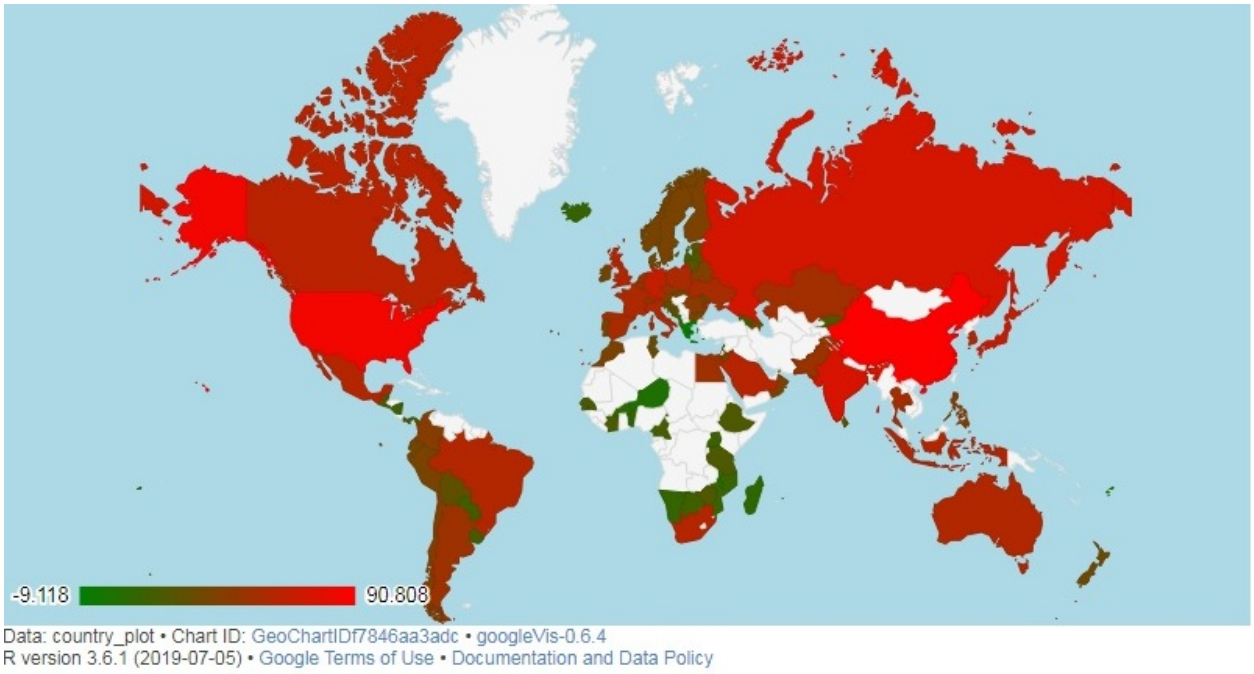



Figure 6.8: Visualization of sum of log of co2emission over the years

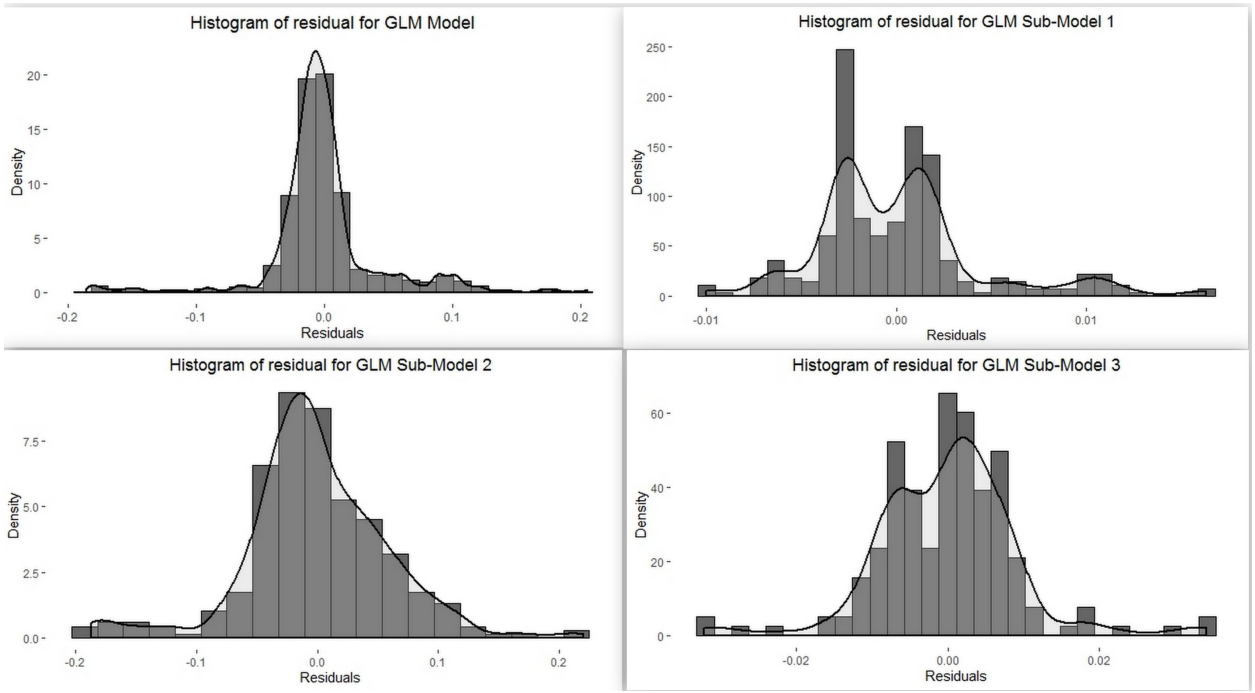



Figure 6.9: Histogram of GLM Residuals for Complete data, Sub-Model1, Sub-Model2, Sub-Model3 (clockwise from topleft)

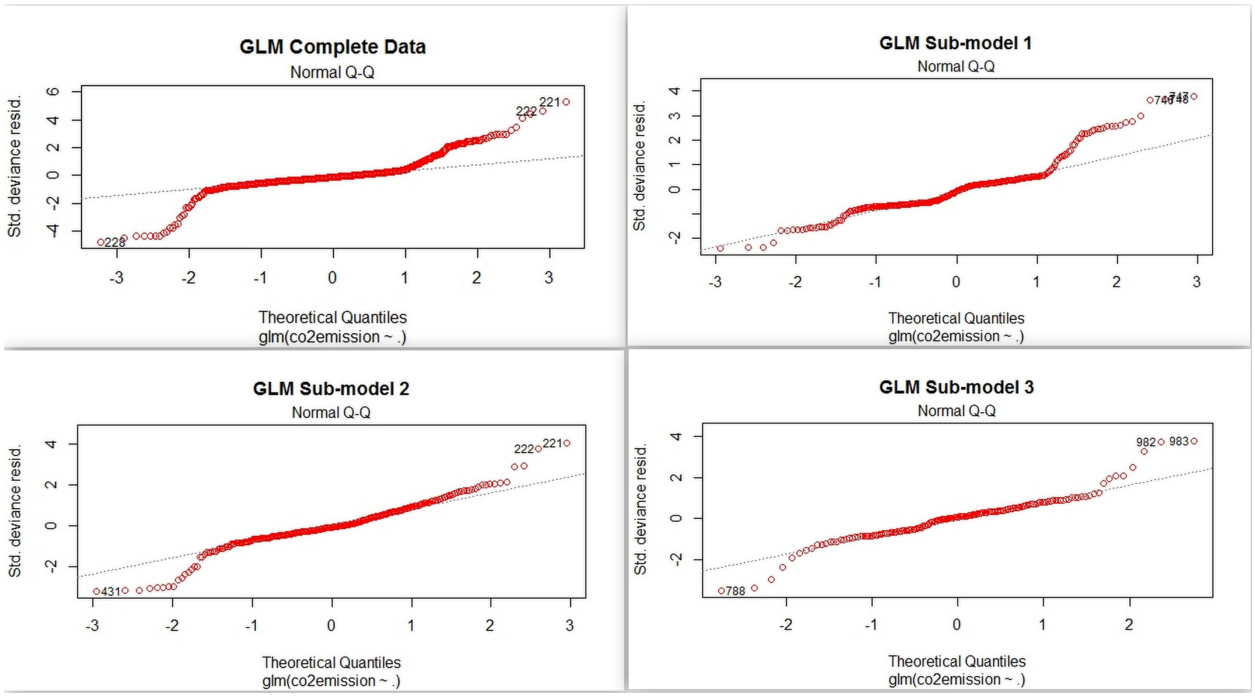



Figure 6.10: Q-Q plot of GLM Residuals for Complete data, Sub-Model1, Sub-Model2, Sub-Model3 (clockwise from topleft)

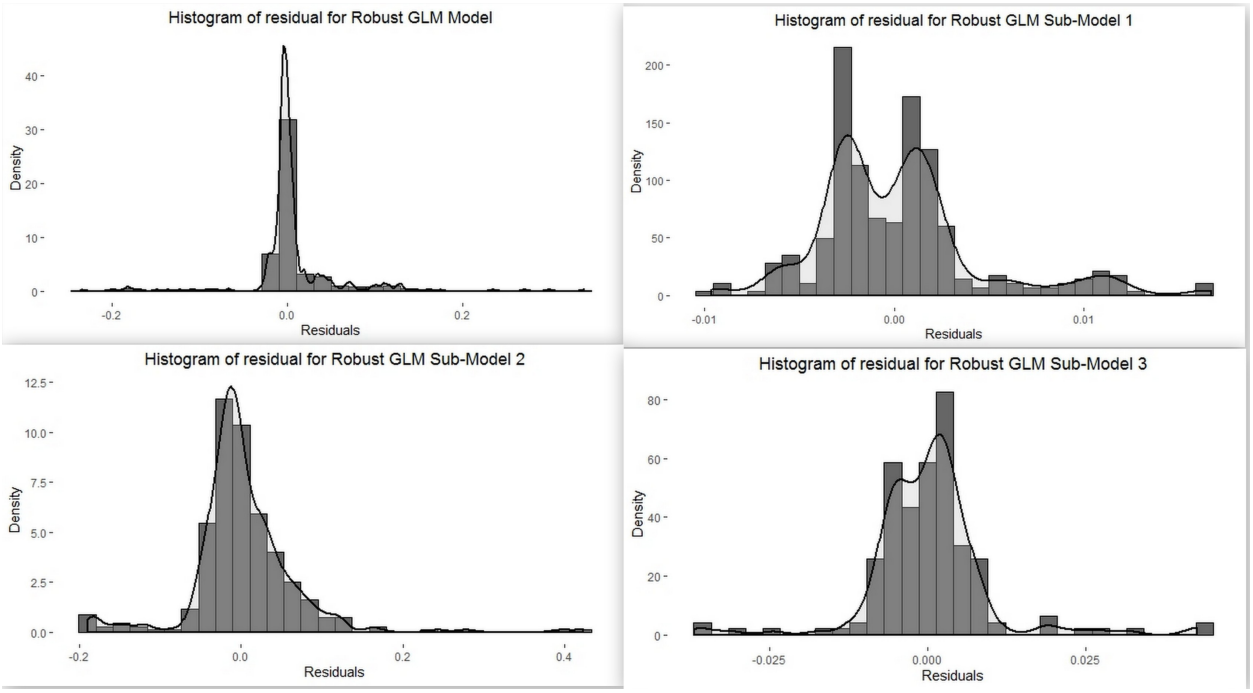



Figure 6.11: Histogram of Robust GLM Residuals for Complete data, Sub-Model1, Sub-Model2, Sub-Model3 (clockwise from topleft)

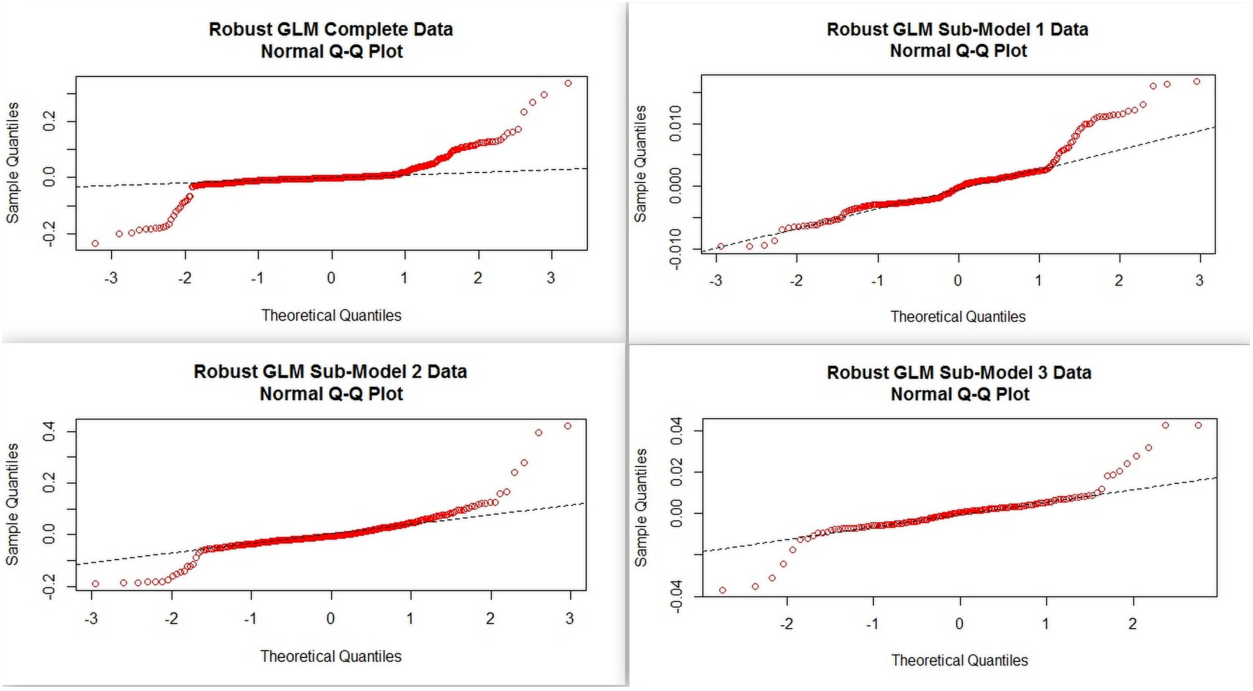



Figure 6.12: Q-Q plot of Robust GLM Residuals for Complete data, Sub-Model1, Sub-Model2, Sub-Model3 (clockwise from topleft)

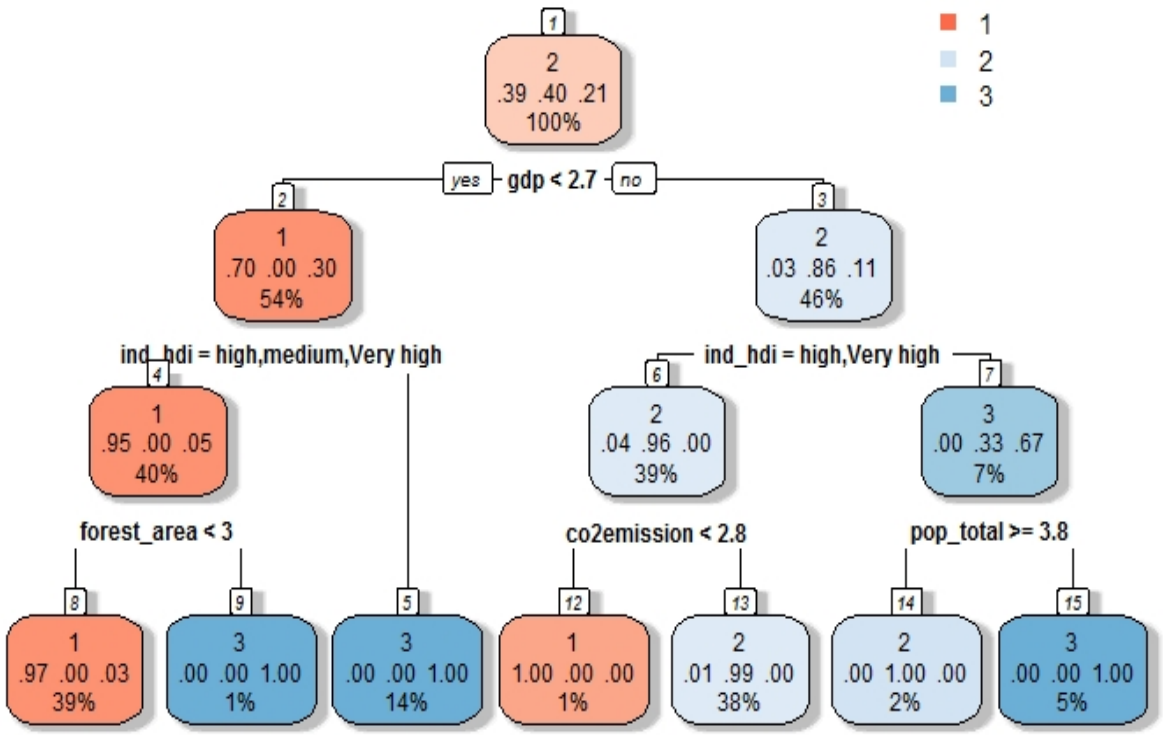



Figure 6.13: Decision Tree for classification

## 6.5 Analysis and Results

Initially the variables with percentage or index were categorized into categorical values which are ordinal in nature. They are:

| Variable | Category |
|---|---|
| ind_hdi | [0.8,1]: Very High<br>[0.7,0.8): High<br>[0.55,0.7): Medium<br>[0,0.55): Low |
| ind_inflation | [ 20,):Galloping<br>[10,20):Running<br>[4,10):Walking/Moderate<br>[0,4):Creeping/Low<br>( ,0):Deflation |
| ind_elec | [80,100]:Very High<br>[60,80):High<br>[40,60):Moderate<br>[20,40):Low<br>[0,20):Very Low |

***Summary Statistics of the variables***

| pop_total | | | co2emission | | | gnipercapita | | |
|---|---|---|---|---|---|---|---|---|
| Min. | : | 167639 | Min. | : | 0.385 | Min. | : | 772 |
| 1st Qu. | : | 4218751 | 1st Qu. | : | 6.125 | 1st Qu. | : | 7206 |
| Median | : | 10630472 | Median | : | 31.323 | Median | : | 15028 |
| Mean | : | 56411715 | Mean | : | 291.945 | Mean | : | 20320 |
| 3rd Qu. | : | 37992743 | 3rd Qu. | : | 166.831 | 3rd Qu. | : | 31750 |
| Max. | : | 1378665000 | Max. | : | 9820.361 | Max. | : | 91519 |

| fuel_export | | | fuel_import | | | gdp | | |
|---|---|---|---|---|---|---|---|---|
| Min. | : | 1360 | Min. | : | 294481365 | Min. | : | 1278745519 |
| 1st Qu. | : | 10027258116 | 1st Qu. | : | 97099216447 | 1st Qu. | : | 17113784287 |
| Median | : | 210000000000 | Median | : | 376500000000 | Median | : | 60442694915 |
| Mean | : | 1699617118420 | Mean | : | 2284152539170 | Mean | : | 642442089374 |
| 3rd Qu. | : | 1240000000000 | 3rd Qu. | : | 1480000000000 | 3rd Qu. | : | 387500000000 |
| Max. | : | 37500000000000 | Max. | : | 50300000000000 | Max. | : | 18700000000000 |

| forest_area | | | agri_area | | | land_area | | | temp_change | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Min. | : | 3 | Min. | : | 82 | Min. | : | 320 | Min. | : | -0.3200 |
| 1st Qu. | : | 9770 | 1st Qu. | : | 18461 | 1st Qu. | : | 55568 | 1st Qu. | : | 0.6472 |
| Median | : | 37295 | Median | : | 57900 | Median | : | 201670 | Median | : | 0.9775 |
| Mean | : | 329230 | Mean | : | 380194 | Mean | : | 993216 | Mean | : | 1.0417 |
| 3rd Qu. | : | 133207 | 3rd Qu. | : | 282860 | 3rd Qu. | : | 579882 | 3rd Qu. | : | 1.3727 |
| Max. | : | 8151356 | Max. | : | 5278330 | Max. | : | 16377740 | Max. | : | 2.9440 |

| ind_hdi | | | ind_inflation | | | ind_elec | | |
|---|---|---|---|---|---|---|---|---|
| high | : | 284 | creeping/low | : | 497 | high | : | 32 |
| low | : | 147 | deflation | : | 103 | low | : | 44 |
| medium | : | 151 | galloping | : | 40 | moderate | : | 51 |
| very high | : | 418 | running | : | 87 | very high | : | 818 |
| | | | walking/moderate | : | 273 | very low | : | 55 |

***Analyzing categorical variables***

ind_hdi

| high | low | medium | Very high |
|---|---|---|---|
| 284 | 147 | 151 | 418 |

ind_inflation

| creeping/low | deflation | galloping | running | walking/moderate |
|---|---|---|---|---|
| 497 | 103 | 40 | 87 | 273 |

ind_elec

| high | low | moderate | very high | very low |
|---|---|---|---|---|
| 32 | 44 | 51 | 818 | 55 |

***Variable Selection***

- The categorical variable ind_elec is dropped since one level is dominating over other levels
- The variables fuel_import and fuel_export is dropped since it is already a part of gdp
- The variable land_area is removed since it contains already forest_area and agri_area
- The variable temp_change is dropped since it has no correlation with response variable as in figure 6.3.
- After dropping these variables as mentioned above we go for stepwise regression to look at statistically insignificant variables and hence we get gnipercapita as insignificant which is dropped.
- ***Finally the variables that are selected are:*** year, pop_total, gdp, forest_area, agri_area, ind_hdi, ind_inflation

***Clustering***

By looking at the Silhouette plot in Figure 6.5, we see that the optimal number of clusters is 3. Hence we make 3 clusters from the complete data.

```
pam_cluster

  1  11  21  31  41  51  61  71  81  91 101 111 121 131 141 151 161 171 181 191 201
  1   2   1   2   2   1   2   3   3   1   1   1   1   1   3   2   1   1   1   2   2
211 221 231 241 251 261 271 281 291 301 311 321 331 341 351 361 371 381 391 401 411
  2   2   3   3   2   2   2   2   1   3   2   1   3   2   1   2   2   1   1   3   1
421 431 441 451 461 471 481 491 501 511 521 531 541 551 561 571 581 591 601 611 621
  2   2   2   1   2   2   1   2   2   1   2   1   1   1   1   1   1   3   1   3   2
631 641 651 661 671 681 691 701 711 721 731 741 751 761 771 781 791 801 811 821 831
  1   1   3   1   3   1   3   1   2   2   2   1   3   1   2   3   2   2   1   2   2
841 851 861 871 881 891 901 911 921 931 941 951 961 971 981 991
  2   3   1   2   1   2   3   2   1   3   3   2   1   2   3   3
```

***Descriptive Statistics of the clusters formed***

| | cluster1 | cluster2 | cluster3 |
|---|---|---|---|
| No of countries | 39 | 40 | 21 |
| year | (1.10) | (1.10) | (1.10) |
| pop_total | (2.6969,2.8103) | (2.7188,10.1254) | (2.7279,3.7933) |
| co2emission | (2.7273,2.7891) | (2.7584,11.9119) | (2.7276,3.1967) |
| gdp | (2.6683,2.8508) | (2.7123,12.3421) | (2.669,2.8828) |
| forest_area | (2.6884,2.8575) | (2.6899,10.4027) | (2.6891,3.2347) |
| agri_area | (2.5635,3.0091) | (2.5692,8.6246) | (2.6018,3.6785) |
| ind_hdi1 | high 190 | high 91 | high 3 |

| | | | |
|---|---|---|---|
| ind_hdi2 | low 0 | low 0 | low 147 |
| ind_hdi3 | medium 69 | medium 22 | medium 60 |
| ind_hdi4 | Very high 131 | Very high 287 | Very high 0 |
| ind_inflation1 | creeping/low 171 | creeping/low 251 | creeping/low 75 |
| ind_inflation2 | deflation 45 | deflation 40 | deflation 18 |
| ind_inflation3 | galloping 15 | galloping 14 | galloping 11 |
| ind_inflation4 | running 37 | running 23 | running 27 |
| ind_inflation5 | walking/moderate 122 | walking/moderate 72 | walking/moderate 79 |
| ind_hdi change | ind_hdi(low+medium) | ind_hdi(low+medium) | ind_hdi(Very high +high+medium) |
| ind_inflation change | ind_inflation(galloping +running) | no change | ind_inflation(galloping +running) |

***Regression results***

Nested Cross Validation Results (Result represents the mean squared error)

| Data | Model | Technique | year | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Complete data | GLM | Regular Predict Second Half | 0.0296 | | 0.0498 | | 0.0497 | | 0.2936 |
| | | Regular Day Forward Chaining | 0.0296 | 0.0713 | 0.0308 | 0.0315 | 0.0924 | 0.1235 | 0.1602 |
| | Robust GLM | Regular Predict Second Half | | | 0.0437 | | 0.1683 | | 0.6065 |
| | | Regular Day Forward Chaining | | 0.0342 | 0.0408 | 0.0629 | 0.1278 | 0.2194 | 0.2411 |
| Cluster 1 data | GLM | Regular Predict Second Half | 0.00013 | | 0.00012 | | 0.00014 | | 0.00017 |
| | | Regular Day Forward Chaining | 0.00013 | 0.00012 | 0.00012 | 0.00012 | 0.00017 | 0.00017 | 0.00018 |
| | Robust GLM | Regular Predict Second Half | | | 0.00012 | | 0.00014 | | 0.00016 |
| | | Regular Day Forward Chaining | | 0.00012 | 0.00012 | 0.00012 | 0.00017 | 0.00017 | 0.00018 |
| Cluster 2 data | GLM | Regular Predict Second Half | 0.0606 | | 0.3671 | | 0.1690 | | 1.0066 |
| | | Regular Day Forward Chaining | 0.0606 | 0.5175 | 0.0658 | 0.0558 | 0.4432 | 0.2840 | 0.3105 |
| | Robust GLM | Regular Predict Second Half | | | 0.3812 | | 0.2870 | | 1.3731 |
| | | Regular Day Forward Chaining | | 0.3742 | 0.0629 | 0.0873 | 0.3950 | 0.4163 | 0.4147 |
| Cluster 3 data | GLM | Regular Predict Second Half | 0.00141 | | 0.00156 | | 0.00274 | | 0.00156 |
| | | Regular Day Forward Chaining | 0.00141 | 0.00091 | 0.00169 | 0.0014 | 0.00098 | 0.00081 | 0.00093 |
| | Robust GLM | Regular Predict Second Half | | | 0.00135 | | 0.0031 | | 0.00185 |
| | | Regular Day Forward Chaining | | 0.00073 | 0.00186 | 0.00166 | 0.00088 | 0.00080 | 0.00106 |

***Model Comparison***

| Data | Model | MSE for test set | R squared |
|---|---|---|---|
| Complete data | GLM | 0.2215 | 0.9236 |
| | Robust GLM | 0.4386 | 0.9273 |
| Cluster 1 data | GLM | 0.000165 | 0.3112 |
| | Robust GLM | 0.000165 | 0.3099 |
| Cluster 2 data | GLM | 0.3728 | 0.9346 |
| | Robust GLM | 0.5643 | 0.9326 |
| Cluster 3 data | GLM | 0.001246 | 0.9330 |
| | Robust GLM | 0.001403 | 0.9257 |

***Classification table***

| Algorithm | | Confusion Matrix | | | Accuracy on test set |
|---|---|---|---|---|---|
| | | Actual | | | |
| SVM | Predicted | 1 | 2 | 3 | 97.5% |
| | | 76 | 0 | 0 | |
| | | 2 | 80 | 3 | |
| | | 0 | 0 | 39 | |
| Decision Tree | Predicted | 76 | 0 | 3 | 96% |
| | | 2 | 80 | 3 | |
| | | 0 | 0 | 36 | |

Row vs Column: Year 1 vs Year 8

| | 1.Low (>100Mt) | 2.Moderate (100-500Mt) | 3.High (500-1000Mt) | 4.Very High (1000-1500Mt) | 5.Extremely High (>1500Mt) |
|---|---|---|---|---|---|
| 1.Low(<100Mt) | 70 | 1 | 0 | 0 | 0 |
| 2.Moderate(100-500Mt) | 2 | 16 | 3 | 0 | 0 |
| 3.High(500-1000 Mt) | 0 | 1 | 2 | 0 | 0 |
| 4.Very High(1000-1500 Mt) | 0 | 0 | 0 | 1 | 1 |
| 5.Extremely High(>1500 Mt) | 0 | 0 | 0 | 0 | 3 |

Row vs Column: Year 1 vs Year 9

| | 1.Low (>100Mt) | 2.Moderate (100-500Mt) | 3.High (500-1000Mt) | 4.Very High (1000-1500Mt) | 5.Extremely High (>1500Mt) |
|---|---|---|---|---|---|
| 1.Low(<100Mt) | 68 | 3 | 0 | 0 | 0 |
| 2.Moderate(100-500Mt) | 7 | 12 | 2 | 0 | 0 |
| 3.High(500-1000 Mt) | 0 | 2 | 1 | 0 | 0 |
| 4.Very High(1000-1500 Mt) | 0 | 0 | 1 | 0 | 1 |
| 5.Extremely High(>1500 Mt) | 0 | 0 | 0 | 1 | 2 |

Year 1 vs Year 8: Transition of change(Count) 8

Year 1 vs Year 9: Transition of change(Count) 17

# Chapter 7

# R Code

```
#importing file
carbon<-read.csv('D:/Project work/final_data_climate_change.csv',header=T)
#Creating a copy of the file
carbon.copy<-carbon

#Creating some functions
#Performance Evaluation of training set using day forward technique
#GLM Model
day_forward.glm<-function(x)
{
  avgmse=vector()
  c=0
  for(i in 2:max(x$year))
  {
    train=x[x$year>=1 & x$year<i,]
    test=x[x$year==i,]
    regressor=glm(formula=co2emission~.,data=train[,3:ncol(train)],
                  family = Gamma(link=inverse))
    yhat=predict(regressor,test[,3:ncol(test)],type='response')
    yobs=test[,5]
    avgmse=c(avgmse,mean((yhat-yobs)^2))
    c=c+1
  }
  print(avgmse)
  return(sum(avgmse)/c)
}
# Robust Regression
day_forward.glmrob<-function(x)
```

```
{
  library(robustbase)
  avgmse=vector()
  c=0
  for(i in 2:max(x$year))
  {
    train=x[x$year>=1 & x$year<i,]
    test=x[x$year==i,]
    regressor=glmrob(formula=co2emission~.,data=train[,3:ncol(train)],model=T,
                     family = Gamma(link=inverse))
    yhat=predict(regressor,test[,3:ncol(test)],type='response')
    yobs=test[,5]
    avgmse=c(avgmse,mean((yhat-yobs)^2))
    c=c+1
  }
  print(avgmse)
  return(sum(avgmse,na.rm=T)/(c-sum(is.na(avgmse))))
}

#Performance Evaluation of training set using Predict second half technique
#GLM Model
predict_second_half.glm<-function(x)
{
  avgmse=vector()
  c=0
  for(i in 1:(max(x$year)/2))
  {
    da=x[x$year>=1 & x$year<=2*i,]
    train=da[da$year<=(max(da$year)/2),]
    test=da[da$year>(max(da$year)/2),]
    regressor=glm(formula=co2emission~.,data=train[,3:ncol(train)],
                  family = Gamma(link=inverse))
    yhat=predict(regressor,test[,3:ncol(test)],type='response')
    yobs=test[,5]
    avgmse=c(avgmse,mean((yhat-yobs)^2))
    c=c+1
  }
  print(avgmse)
  return(sum(avgmse)/c)
}
```

```
# Robust Regression
predict_second_half.glmrob<-function(x)
{
  library(robustbase)
  avgmse=vector()
  c=0
  for(i in 1:(max(x$year)/2))
  {
    da=x[x$year>=1 & x$year<=2*i,]
    train=da[da$year<=(max(da$year)/2),]
    test=da[da$year>(max(da$year)/2),]
    regressor=glmrob(formula=co2emission~.,data=train[,3:ncol(train)],model=T,
                     family = Gamma(link=inverse))
    yhat=predict(regressor,test[,3:ncol(test)],type='response')
    yobs=test[,5]
    avgmse=c(avgmse,mean((yhat-yobs)^2))
    c=c+1
  }
  print(avgmse)
  return(sum(avgmse,na.rm=T)/(c-sum(is.na(avgmse))))
}

#Function for model building
#GLM Model
model.glm<-function(x)
{
  regressor<-glm(formula=co2emission~.,data=x[,3:10],family = Gamma(link=inverse))
  library(rsq)
  cat("The R squared value is ",rsq(regressor))
  return(regressor)
}
#Robust GLM Model
model.glmrob<-function(x)
{
  library(robustbase)
  regressor<-glmrob(formula=co2emission~.,data=x[,3:10],model=T,
                    family = Gamma(link=inverse))
  actual <- x$co2emission
  preds <- regressor$fitted.values
  rss <- sum((preds-actual)^ 2)
```

```
  tss <- sum((actual - mean(actual)) ^ 2)
  rsq <- 1 - rss/tss
  cat("The R squared value for the model is ",rsq)
  return(regressor)
}

#Function for graphical display
#GLM Model
draw.glm<-function(object,st)
{
  #Model adequacy checking
  #Residual plots
  library(ggplot2)
  d=fortify(object)
  g=ggplot(d,aes(x=.resid))+
    geom_histogram(aes(y=..density..),color="black",fill="black",alpha=0.6)+
    geom_density(alpha=0.3,color='black',lwd=0.75,fill='grey')+
    ylab("Density")+
    xlab("Residuals")+
    ggtitle(paste0("Histogram of residual-",st))+
    theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
      panel.background = element_rect(fill = "transparent",colour = NA),
      plot.background = element_rect(fill = "transparent",colour = NA),
      plot.title = element_text(hjust = 0.5))
  print(g)
  #QQ plot
  plot(object, which=2,col=c('red'),main=st)
}
#Robust GLM Model
draw.glmrob<-function(object,st)
{
  d=data.frame(object$residuals,object$fitted.values)
  names(d)=c('residuals','fitted values')
  g=ggplot(d,aes(x=residuals))+
    geom_histogram(aes(y=..density..),color="black",fill="black",alpha=0.6)+
    geom_density(alpha=0.3,color='black',lwd=0.75,fill='grey')+
    ylab("Density")+
    xlab("Residuals")+
    ggtitle(paste0("Histogram of residual-",st))+
    theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
```

```
        panel.background = element_rect(fill = "transparent",colour = NA),
        plot.background = element_rect(fill = "transparent",colour = NA),
        plot.title = element_text(hjust = 0.5))
  print(g)
  #QQ plot
  qqnorm(object$residuals,pch=1,col='red',main=paste0(st,'\nNormal Q-Q Plot'))
  qqline(object$residuals,lty=2, lwd = 1)
}

#Prediction error function
#GLM Model
predicterror.glm<-function(object,z)
{
  #Prediction
  pred=predict(object,z,type='response')
  obs=z$co2emission
  mean.squared.error=mean((obs-pred)^2)
  return(mean.squared.error)
}
#Robust GLM Model
predicterror.glmrob<-function(object,z)
{
  #Model Comparison
  pred=predict(object,z,type='response')
  obs=z$co2emission
  mean.squared.error=mean((obs-pred)^2)
  return(mean.squared.error)
}

# Visualizing the carbon dioxide emission
# Mean of top 40 countries with carbon dioxide emission
attach(carbon)
country_plot<-aggregate(x=co2emission,by=list(countries),FUN=mean,simplify=T)
country_plot=country_plot[order(country_plot[,2],decreasing = T),]
barplot(country_plot$x[1:40],names.arg = country_plot$Group.1[1:40],las=2,
cex.names = 0.65,cex.axis=0.7,xlab="Countries",ylab="Carbon Dioxide Emission(in Mton)",
main="Top 40 countries with mean carbon dioxide emission over the years")

#Visualizing the carbon emission using sphagetti plot
par(mfrow=c(2,2))
```

```
interaction.plot(carbon$year[1:250],carbon$country_code[1:250],
carbon$co2emission[1:250],xlab="Time",ylab="Carbon Dioxide Emission",legend=F)
interaction.plot(carbon$year[251:500],carbon$country_code[251:500],
carbon$co2emission[251:500],xlab="Time",ylab="Carbon Dioxide Emission",legend=F)
interaction.plot(carbon$year[501:750],carbon$country_code[501:750],
carbon$co2emission[501:750],xlab="Time",ylab="Carbon Dioxide Emission",legend=F)
interaction.plot(carbon$year[751:1000],carbon$country_code[751:1000],
carbon$co2emission[751:1000],xlab="Time",ylab="Carbon Dioxide Emission",legend=F)
mtext("Carbon dioxide emission vs time (each plot of 25 countries)",
      side = 3, line = -2, outer = TRUE)
dev.off()

#Histogram for carbon dioxide emission
library(ggplot2)
ggplot(carbon,aes(x=co2emission))+
  geom_histogram(aes(y=..count..),color="black",fill="grey",bins=15)+
  ylab("Frequency of the variable")+
  xlab("Carbon Dioxide Emission")+
  ggtitle("Histogram of carbon dioxide emission of complete data")+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "transparent",colour = NA),
    plot.background = element_rect(fill = "transparent",colour = NA),
    plot.title = element_text(hjust = 0.5))

# Descriptive Statistics
options(scipen=999)
summary(carbon[,4:16])

###################### Analyzing variables ##################################
##checking the categorical variables
table(ind_hdi)
table(ind_inflation)
table(ind_elec)

## Dropping the insignificant variables as per subject knowledge
##Dropping ind_elec,fuel_import,fuel_export,land_area
carbon<-carbon[,-c(7,8,12,16)]

## finding correlation between numerical variables that is left
library(psych)
```

```
pairs.panels(carbon[,3:10],method = "pearson",hist.col = "#00AFBB",
             density=TRUE, ellipses=F,
             main="Correlation and scatter plot of the data")

## looking at the correlation matrix we can drop temp_change variable
carbon<-carbon[,-10]

### Variable selection
#Standardizing the numerical variables to remove the unit problem
carbon[,4:9]<-scale(carbon[,4:9])+3
var_select<-glm(formula=co2emission~.,data=carbon[,3:11],family = Gamma(link=inverse))
##Going for stepwise regression and checking if significance of variables
step(var_select,direction = "both")

#From stepwise we see that gnipercapita can also be dropped
#Finally we get the variables for modeling as
carbon<-carbon[,-6]
var_name<-names(carbon)

#Final dataset
carbon<-carbon.copy[,var_name]
carbon.copy<-carbon.copy[,var_name]

#### Feature scaling
carbon[,4:8]<-scale(carbon[,4:8])+3

##Splitting the data into train and test
training_set=carbon[carbon$year>=1 & carbon$year<=8,]
test_set=carbon[carbon$year==9|carbon$year==10,]

#Performance Evaluation of training set using day forward technique
day_forward.glm(training_set)

#Performance Evaluation of training set using predict second half technique
predict_second_half.glm(training_set)

# Overall GLM Data Modeling
obj=model.glm(training_set)
summary(obj)
draw.glm(obj,'GLM Complete Data')
```

```
predicterror.glm(obj,test_set)

############## ROBUST REGRESSION ############################
#Performance Evaluation of training set using day forward technique
day_forward.glmrob(training_set)

#Performance Evaluation of training set using predict second half technique
predict_second_half.glmrob(training_set)

# Overall Robust GLM Data Modeling
obj2=model.glmrob(training_set)
summary(obj2)
draw.glmrob(obj2,'Robust GLM Complete Data')
predicterror.glmrob(obj2,test_set)
detach(carbon)

#################### CLUSTERING ################################################
carbon<-carbon.copy
#Converting categorical to ordinal data
library(plyr)
carbon$ind_hdi<-mapvalues(carbon$ind_hdi,from=c("low","medium","high","Very high"),
to = c(1,2,3,4))
carbon$ind_inflation<-mapvalues(carbon$ind_inflation,
from=c("deflation","creeping/low","walking/moderate","running", "galloping"),
to = c(1,2,3,4,5))
carbon$ind_hdi=as.numeric(levels(carbon$ind_hdi))[carbon$ind_hdi]
carbon$ind_inflation=as.numeric(levels(carbon$ind_inflation))[carbon$ind_inflation]

#Bringing the variables to same level through transformation
carbon[,4:8]<-log(carbon[,4:8])

#Normalizing the variables both continuous and ordinal
library(BBmisc)
carbon[,4:10]<-normalize(carbon[,4:10],method="range",range=c(0,1))

#Creating distance matrix
dd<-matrix(0,nrow=100,ncol=100)
cc<-matrix(0,nrow=100,ncol=100)
for(i in 1:10)
{
```

```
  carb_clus<-carbon[carbon$year==i,c(-1,-3)]
  cc<-as.matrix(dist(carb_clus[,-1],method='euclidean'))
  cc<-cc^2
  dd=dd+cc
}
dd=sqrt(dd)

#Elbow method
library(cluster)
sil_width <- c(NA)
for(i in 2:10)
{
  pam_fit <- pam(dd, diss = TRUE, k = i)
  sil_width[i] <- pam_fit$silinfo$avg.width
}

# Plot sihouette width (higher is better)
plot(1:10, sil_width,
     xlab = "Number of clusters",
     ylab = "Silhouette Width")
lines(1:10, sil_width)

#By looking at plot, optimal no of cluster is 3
pam_fit <- pam(dd, diss = TRUE, k = 3)
pam_cluster=pam_fit$clustering
pam_cluster

#Preparing cluster list
cluster_list<-data.frame(carbon[carbon$year==1,2],pam_cluster)
names(cluster_list)<-c("countries","cluster_no")

#Plotting cluster numbers in a graph
library(Rtsne)
tsne_obj <- Rtsne(dd, is_distance = TRUE)
library(magrittr)
# needs to be run every time you start R and want to use %>%
library(dplyr)
tsne_data <- tsne_obj$Y %>%data.frame() %>%setNames(c("X", "Y")) %>%
  mutate(cluster = factor(pam_cluster),name = cluster_list$countries)
```

```
library(ggplot2)
ggplot(aes(x = X, y = Y), data = tsne_data) +
  geom_point(aes(color = cluster),size=3)

library(rgl)
pcdf <- princomp(dd,cor=T,score=T)

#Create a 3D plot
plot3d(pcdf$scores, col=cluster_list$cluster_no,type='s',size=1,xlab='Cluster 1',
       ylab='Cluster 2',zlab='Cluster 3')

library(dplyr)
carbon_clus.submodel<-left_join(carbon.copy,cluster_list,by="countries")
carbon_clus.classify<-left_join(carbon,cluster_list,by="countries")

#### Visualization for the clusters on the world map
library(googleVis)
WorldMap=cluster_list
G5<-gvisGeoChart(WorldMap, "countries", "cluster_no",
options=list(displayMode='auto',dataMode="regions",
colorAxis="{colors:['blue', 'red','green']}",backgroundColor="lightblue",
width=800, height=400,legend="{colors:['blue','red','green']}"))
plot(G5)
#Countries not plotted-Malta,Bahrain,Barbados,St. Lucia due to small area size

country_plot<-aggregate(x=log(carbon.copy$co2emission),
                        by=list(carbon.copy$countries),FUN=sum,simplify=T)
names(country_plot)<-c('countries','sum_of_co2emission')
#Plotting for sum of co2emission over the years
G6<-gvisGeoChart(country_plot, "countries", 'sum_of_co2emission',
options=list(displayMode='auto',dataMode="regions",
colorAxis="{colors:['green','red']}",backgroundColor="lightblue",
width=800, height=400))
plot(G6)

############## WORKING ON SUB MODELS ##########################################
clustered_data<-carbon_clus.submodel
clustered_data[,4:8]<-scale(clustered_data[,4:8])+3
submodel1<-clustered_data[clustered_data$cluster_no==1,-11]
submodel2<-clustered_data[clustered_data$cluster_no==2,-11]
```

```
submodel3<-clustered_data[clustered_data$cluster_no==3,-11]

#Getting summary statistics table for clusters
t=data.frame()
for(i in 2:10)
{
  if(i==2)
  {
    t[i,1]=length(unique(submodel1[,i]))
    t[i,2]=length(unique(submodel2[,i]))
    t[i,3]=length(unique(submodel3[,i]))
  }
  if(i>=3 & i<=8)
  {
    t[i,1]=paste0("(",round(min(submodel1[,i]),4),",",round(max(submodel1[,i]),4),")")
    t[i,2]=paste0("(",round(min(submodel2[,i]),4),",",round(max(submodel2[,i]),4),")")
    t[i,3]=paste0("(",round(min(submodel3[,i]),4),",",round(max(submodel3[,i]),4),")")
  }
  if(i==9)
  {
    z=cbind(paste(rownames(table(submodel1[,9])),table(submodel1[,9])),
            paste(rownames(table(submodel2[,9])),table(submodel2[,9])),
            paste(rownames(table(submodel3[,9])),table(submodel3[,9])))
    t=rbind(t,z)
  }
  if(i==10)
  {
    z=cbind(paste(rownames(table(submodel1[,10])),table(submodel1[,10])),
            paste(rownames(table(submodel2[,10])),table(submodel2[,10])),
            paste(rownames(table(submodel3[,10])),table(submodel3[,10])))
    t=rbind(t,z)
  }
}
t=t[-1,]
level_info1=c("ind_hdi(low+medium)","ind_hdi(low+medium)",
              "ind_hdi(Very high+high+medium)")
level_info2=c("ind_inflation(galloping+running","no change",
              "ind_inflation(galloping+running)")
t=rbind(t,level_info1,level_info2)
rownames(t)<-c("No of countries","year","pop_total","co2emission","gdp",
```

```
"forest_area","agri_area","ind_hdi1","ind_hdi2","ind_hdi3","ind_hdi4",
"ind_inflation1","ind_inflation2","ind_inflation3","ind_inflation4","ind_inflation5",
"ind_hdi change","ind_inflation change")
colnames(t)<-c("cluster1","cluster2","cluster3")

############ Merging some levels in the clustered data ####################
mergelevel<-function(submodel,choice)
{
#In 1st cluster:ind_hdi==>low level+medium level,ind_inflation==>deflation+low,
#ind_inflation==>galloping+running
library(plyr)
if(choice=='1')
{
submodel$ind_hdi<-mapvalues(submodel$ind_hdi,from=c("low","medium","high","Very high"),
to = c('low+medium','low+medium','high','Very high'))
submodel$ind_inflation<-mapvalues(submodel$ind_inflation,
from=c("deflation","creeping/low","walking/moderate","running", "galloping"),
to = c("deflation+low","deflation+low","walking/moderate",'running+galloping',
        'running+galloping'))
}
if(choice=='2')
{
#In 2nd Cluster:ind_hdi==>low+medium
submodel$ind_hdi<-mapvalues(submodel$ind_hdi,from=c("low","medium","high","Very high"),
to = c('low+medium','low+medium','high','Very high'))
}
if(choice=='3')
{
#In 3rd cluster:ind_hdi==>Very high+high+medium,ind_inflation==>deflation+low,
#ind_inflation==>galloping+running
submodel$ind_hdi<-mapvalues(submodel$ind_hdi,from=c("low","medium","high","Very high"),
to = c('low','medium+high+veryhigh','medium+high+veryhigh','medium+high+veryhigh'))
submodel$ind_inflation<-mapvalues(submodel$ind_inflation,
from=c("deflation","creeping/low","walking/moderate","running", "galloping"),
to = c("deflation+low","deflation+low","walking/moderate",'running+galloping',
        'running+galloping'))
}
  return(submodel)
}
#################### SUB MODEL 1 ########################################
```

```
#Histogram for carbon dioxide emission
library(ggplot2)
ggplot(submodel1,aes(x=co2emission))+
  geom_histogram(aes(y=..count..),color="black",fill="grey",alpha=0.6)+
  ylab("Frequency of the variable")+
  xlab("Carbon Dioxide Emission")+
  ggtitle("Histogram of carbon dioxide emission for submodel 1")+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "transparent",colour = NA),
    plot.background = element_rect(fill = "transparent",colour = NA),
    plot.title = element_text(hjust = 0.5))

#making conversion of categorical variables
submodel1=mergelevel(submodel1,'1')

#Splitting the data into training and test set
training_set=submodel1[submodel1$year>=1 & submodel1$year<=8,]
test_set=submodel1[submodel1$year==9|submodel1$year==10,]

#Performance Evaluation of training set using day forward technique
day_forward.glm(training_set)

#Predict second half GLM validation
predict_second_half.glm(training_set)

#creating basic glm model
subobj1=model.glm(training_set)
summary(subobj1)
draw.glm(subobj1,'GLM Sub-Model 1')
predicterror.glm(subobj1,test_set)

############## ROBUST REGRESSION FOR SUB-MODEL 1 ############################
#Performance Evaluation of training set using day forward technique
day_forward.glmrob(training_set)

#Predict second half GLM validation
predict_second_half.glmrob(training_set)

#robust regression of glm sub-model 1
```

```
subobj1.2=model.glmrob(training_set)
summary(subobj1.2)
draw.glmrob(subobj1.2,'Robust GLM Sub-Model 1')
predicterror.glmrob(subobj1.2,test_set)

#################### SUB MODEL 2 ######################################
#Histogram for carbon dioxide emission
library(ggplot2)
ggplot(submodel2,aes(x=co2emission))+
  geom_histogram(aes(y=..count..),color="black",fill="grey",alpha=0.6,bins=20)+
  ylab("Frequency of the variable")+
  xlab("Carbon Dioxide Emission")+
  ggtitle("Histogram of carbon dioxide emission for Sub-Model 2")+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "transparent",colour = NA),
    plot.background = element_rect(fill = "transparent",colour = NA),
    plot.title = element_text(hjust = 0.5))

#making conversion of categorical variables
submodel2=mergelevel(submodel2,'2')

#Splitting the data into training and test set
training_set=submodel2[submodel2$year>=1 & submodel2$year<=8,]
test_set=submodel2[submodel2$year==9|submodel2$year==10,]

#Performance Evaluation of training set using day forward technique
day_forward.glm(training_set)

#Predict second half GLM validation
predict_second_half.glm(training_set)

#creating basic glm model
subobj2=model.glm(training_set)
summary(subobj2)
draw.glm(subobj2,'GLM Sub-Model 2')
predicterror.glm(subobj2,test_set)

############## ROBUST REGRESSION FOR SUB-MODEL 2 ############################
#Performance Evaluation of training set using day forward technique
day_forward.glmrob(training_set)
```

```
#Predict second half Robust GLM validation
predict_second_half.glmrob(training_set)

#robust regression of glm sub-model 2
subobj2.2=model.glmrob(training_set)
summary(subobj2.2)
draw.glmrob(subobj2.2,'Robust GLM Sub-Model 2')
predicterror.glmrob(subobj2.2,test_set)

#################### SUB MODEL 3 ########################################
#Histogram for carbon dioxide emission
library(ggplot2)
ggplot(submodel3,aes(x=co2emission))+
  geom_histogram(aes(y=..count..),color="black",fill="grey",alpha=0.6)+
  ylab("Frequency of the variable")+
  xlab("Carbon Dioxide Emission")+
  ggtitle("Histogram of carbon dioxide emission for sub-model 3")+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
    panel.background = element_rect(fill = "transparent",colour = NA),
    plot.background = element_rect(fill = "transparent",colour = NA),
    plot.title = element_text(hjust = 0.5))

#making conversion of categorical variables
submodel3=mergelevel(submodel3,'3')

#Splitting the data into training and test set
training_set=submodel3[submodel3$year>=1 & submodel3$year<=8,]
test_set=submodel3[submodel3$year==9|submodel3$year==10,]

#Performance Evaluation of training set using day forward technique
day_forward.glm(training_set)

#Predict second half GLM validation
predict_second_half.glm(training_set)

#creating basic glm model
subobj3=model.glm(training_set)
summary(subobj3)
draw.glm(subobj3,'GLM Sub-Model 3')
```

```
predicterror.glm(subobj3,test_set)

############## ROBUST REGRESSION FOR SUB-MODEL 3 ############################
#Performance Evaluation of training set using day forward technique
day_forward.glmrob(training_set)

#Predict second half GLM validation
predict_second_half.glmrob(training_set)

#robust regression of glm sub-model 3
subobj3.2=model.glmrob(training_set)
summary(subobj3.2)
draw.glmrob(subobj3.2,'Robust GLM Sub-Model 3')
predicterror.glmrob(subobj3.2,test_set)

###################### CLASSIFICATION ########################################
data<-carbon_clus.classify
data$cluster_no<-as.factor(data$cluster_no)

#Splitting the data into training and test based on earlier criteria
training_set=data[data$year>=1&data$year<=8,-c(1,2,3)]
test_set=data[data$year==9|data$year==10,-c(1,2,3)]

###################### SVM ########################################
library(e1071)
classobj1<-svm(formula=cluster_no~.,data = training_set,kernel='radial')
summary(classobj1)
pred=predict(classobj1,test_set[,-8])
t1<-as.matrix(table(pred,test_set[,8]))
rownames(t1)=c("","Pedicted","")
colnames(t1)=c("","Actual","")
t1
cat("Accuracy Score: ",sum(diag(t1))/nrow(test_set)*100,"%\n")

######################### DECISION TREE ###################################
#Decision trees dont need any preprocessing
#But still we will scale the continuous variables since values are large
data<-carbon_clus.submodel
data[,4:8]<-scale(data[,4:8])+3
data$cluster_no<-as.factor(data$cluster_no)
```

```
#Splitting the data into training and test based on earlier criteria
training_set=data[data$year>=1&data$year<=8,-c(1,2,3)]
test_set=data[data$year==9|data$year==10,-c(1,2,3)]

library(rpart)
classobj2<-rpart(formula=cluster_no~.,data=training_set,
                 control = rpart.control(minsplit = 1))
pred=predict(classobj2,test_set[,-8],type='class')
t2<-as.matrix(table(pred,test_set[,8]))
rownames(t2)=c("","Pedicted","")
colnames(t2)=c("","Actual","")
t2
cat("Accuracy Score: ",sum(diag(t2))/nrow(test_set)*100,"%\n")
library(rpart.plot)
rpart.plot(classobj2, box.palette="RdBu",shadow.col="gray", nn=TRUE)

################ FINAL STATUS ############################
#Function for categorizing carbon emission
carbon_category<-function(x)
{
  if(x>=4.13)
  {
    status='5.Extremely High(>1500 Mt)'
  }else{
    if(x>=3.66 & x<4.13)
    {
      status='4.Very High(1000-1500 Mt)'
    }else
    {
      if(x>=3.19 & x<3.66)
      {
        status='3.High(500-1000 Mt)'
      }else{
        if(x>=2.82 & x<3.19)
        {
          status='2.Moderate(100-500 Mt)'
        }else{
            status='1.Low(<100 Mt)'
        }}}}
```

```
  return(status)
}
emission_status=vector()
carbon=carbon.copy
carbon[,4:8]=scale(carbon[,4:8])+3
for(i in 1:nrow(carbon))
  emission_status[i]=carbon_category(carbon[i,5])
carbon_map=data.frame(carbon[,c(2,3,5)],emission_status)

######################## Prediction Result for our study ##########################
#Ready to classify-using SVM
studydata1=carbon_clus.classify[carbon_clus.classify$year==9|
                                  carbon_clus.classify$year==10,-c(1,2,3,11)]
#Predicting the class
predclass=predict(classobj1,studydata1)
c1=which(predclass=='1')
c2=which(predclass=='2')
c3=which(predclass=='3')

#Predicting carbon emission after preprocessing
studydata2=carbon[carbon$year==9|carbon$year==10,]
#Use submodel1 for cluster1,complete data model for cluster2,submodel3 for cluster3
data1=mergelevel(studydata2[c1,],'1')
data2=studydata2[c2,]
data3=mergelevel(studydata2[c3,],'3')
predval1=predict(subobj1,data1,type='response')
predval2=predict(obj,data2,type='response')
predval3=predict(subobj3,data3,type='response')
predcarbon=data.frame(rbind(data1[,2:3],data2[,2:3],data3[,2:3]),
                      c(predval1,predval2,predval3))
names(predcarbon)=c('countries','year','pred_emission')

########### Study carbon status history ##########################
#Checking status of predicted value
predemission_status=vector()
for(i in 1:nrow(predcarbon))
  predemission_status[i]=carbon_category(predcarbon[i,3])
predcarbon=cbind(predcarbon,predemission_status)
library(dplyr)
carbon_map=cbind(paste0(carbon_map[,1],carbon_map[,2]),carbon_map)
```

```
names(carbon_map)<-c('countries_year','countries','year','co2emission',
                     'emission_status')
countries_year=paste0(predcarbon[,1],predcarbon[,2])
predcarbon=cbind(countries_year,predcarbon)
predcarbon=predcarbon[,-c(2,3)]
histstudy=left_join(carbon_map,predcarbon,by="countries_year")[,-1]
t0=table(histstudy[histstudy$year==9|histstudy$year==10,4],
      histstudy[histstudy$year==9|histstudy$year==10,6])
accuracy=(sum(diag(t0))/200)*100
accuracy
t1=table(histstudy[histstudy$year==1,4],histstudy[histstudy$year==8,4])
t2=table(histstudy[histstudy$year==1,4],histstudy[histstudy$year==9,6])
cat("Row vs Column:Year 1 vs Year 8\n")
t1
cat("Row vs Column:Year 1 vs Year 9\n")
t2
cat("Year 1 vs Year 8: Transition of change(Count)",100-sum(diag(t1)),"\n")
cat("Year 1 vs Year 9: Transition of change(Count)",100-sum(diag(t2)),"\n")
```

# Chapter 8

# Conclusion

In our study, we have dealt with an approach to predict carbon dioxide emission based on the classification of the countries. The study can help to understand the dynamics of climate change in a more better way than just going for a descriptive analysis. In our study, we first classify the data based on the features and based on the label predicted we choose the model to predict the carbon dioxide emission. On prediction of the carbon dioxide emission, we know the status of carbon emission through the categories created and we study the historical emissions of the countries and see the jump in the status of the countries.

In our analysis, we see that R squared measure cannot be taken as a criteria for judgment of models since as variation in the response variable reduces drastically, the R squared value deteriorates although the prediction power measured through MSE is good.By looking at the validation table, we see that sub-model 1 and sub-model 3 are the best models with not much variation between Robust GLM and GLM models. Sub-model 2 deteriorates year wise and hence it is not appropriate for prediction. Instead we can use complete model in place of sub-model 2. It has been observed that most of the time GLM model proves to be better that robust GLM model but for countries with high values or extreme values, Robust GLM is the best choice. It was noticed that due to extreme values there is deviation in the assumptions in the tail ends as visible in model adequacy checking.

In the classification problem, we see that SVM performs better than decision trees and hence SVM was used in our predictions irrespective of the year since we have made classifier considering data to be independent of time. But based on subject knowledge, decision trees are supposed to be better classifier for longitudinal data. So, this thing can be kept in mind while performing classification. After classification we perform prediction based on the models selected for the groups. Predicted values are categorized to see the transition that

has occurred over the years by the 100 countries.

After the prediction and categorization we see that there is an increase in the number of countries which has shown transition between year 1 and year 9 from year 1 and year 8. This analysis can be more refined by analyzing individual countries but currently this is not part of our study. Hence, we conclude that even with limitation of data(due to smoothed data by secondary sources) we can say that this approach is very helpful in understanding the dynamics of carbon emission by countries over the years.

Future studies can involve multivariate longitudinal analysis, multivariate time series forecasting and machine learning as techniques to get better results. A minor study was made with some machine learning model(not included in the study) and it was seen that it performed better than statistical models. Also considering the complex nature of the topic and various complexities in the analysis and limitation of data, machine learning proves to be an apt technique in this field. This study was done to explain the approach in predicting carbon emission and making a useful analysis that can help in building a sustainable world in future. It is just a small contribution in our fight to reduce carbon emission and save our planet.

# REFERENCE


[1] Climate Change Performance Index 2019: https://www.climate-change-performance-index.org/

[2] IPCC Special Report: Global Warming of 1.5$^o$C. https://www.ipcc.ch/sr15/

[3] Gujarati, D.N. (2003). *Basic Econometrics, International Edition - 4th ed.* McGraw-Hill Higher Education

[4] Fitzmaurice, G.M., Laird, N.M. and Ware, J.H. (2004). *Applied LongitudinalAnalysi-1 st ed.* Wiley Series in Probability and Statistics

[5] Gibbons, R. D. and Hedeker D. (2006). *Longitudinal Data Analysis-1st ed.* Wiley Series in Probability and Statistics. pp 1-4

[6] Davidian, M. *Course notes on Applied Longitudinal Data Analysis:* https://www4.stat.ncsu.edu/ davidian/st732_sp2007/

[7] Petrinco, M. & Barbati, Giulia & Meylan, D. & Pagano, Eva & Gregori, Dario & Merletti, Franco & Marazzi, Alfio. (2012). *Robust Gamma regression models for the analysis of health care cost data.* Model Assisted Statistics and Applications. 7.115-124.10.3233/MAS-2011-0218.

[8] Montgomery, D.C., Peck, E.A. and Vining, G.G. (2012). *Introduction to Linear Regression Analysis-5th ed.* Wiley Series in Probability and Statistics

[9] Weiss, R.E.(2005). *Modeling Longitudinal Data-1st ed.* Springer Publication

[10] Zheng, B., Li, S. (2014). *Multivariable panel data cluster analysis and its application.* Computer Modeling & New Technologies pp 553–557

[11] Kaufman, L. and Rousseauw, P. J. (1990) *Finding Groups in Data: An introduction to cluster analysis-9th ed.* A John Wiley and Sons Inc. pp 68-122

[12] Han, J., Kamber, M. and Pei J. (2012) *Data Mining: Concepts and Techniques-3rd ed.* The Morgan Kaufmann Series in Data Management Systems pp 327-442

[13] C. Bergmeir and J. M. Benítez. *On the use of cross-validation for time series predictor evaluation.* Inf. Sci., 191:192–213

[14] Hastie, T., Tibshirani, R. and Friedman, J. (2008). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction-2nd ed.* Springer Pub. pp 215-223

[15] Cochrane,C.(2018). *Time Series Nested Cross Validation:* https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9