SOIL ANALYSIS AND CROP YIELD PREDICTION

Data Visualization
CSE3020

Winter Semester – 2022-2023

Soil Analysis And Crop Yield Prediction

A PROJECT REPORT

Submitted by:

Nandini Singh	20BCE1171
Prashant Kumar Sharma	20BCE1709

CSE3020 – Data Visualization

Project Guide

Dr. Parvathi R
Professor and Associate Dean (Academics)
School of Computer Science and Engineering (SCOPE)

B.Tech. (Computer Science and Engineering)
IN
School of Computer Science and Engineering (SCOPE)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Winter Semester 2022-23

DECLARATION BY THE STUDENT

We Nandini Singh (20BCE1171) and Prashant Kumar Sharma (20BCE1709) hereby declare that this project report entitled **“Soil Analysis And Crop Yield Prediction”** has been prepared by us towards the partial fulfillment of the requirement for the course of **CSE3020 – Data Visualization** under the guidance of Dr. Parvathi R.

We also declare that this project report is my original work and has not been previously submitted for the award of any degree, diploma, fellowship, or other similar titles.

DATE: 10/04/2023

BONAFIDE CERTIFICATE

Certified that this project report titled **"Soil Analysis And Crop Yield Prediction"** is the bonafide work of Nandini Singh (20BCE1171) and Prashant Kumar Sharma (20BCE1709) who carried out the project work under my supervision in the partial fulfillment of the requirements for the course of CSE3020 – Data Visualization.

Dr. Parvathi R

(Name & Signature of the Course Faculty)

ACKNOWLEDGEMENT

First and foremost, we would like to thank our Data Visualization faculty **Dr. Parvathi R.**, who guided us throughout the project. She provided us with invaluable advice and helped us whenever we needed it. Her motivation and help contributed tremendously to the successful completion of the project.

Besides, we would like to thank our institution VIT, without which we wouldn’t have been able to move forward with this project at the first step.

Also, we would like to thank our family and friends for their support. Without that support, we couldn’t have succeeded in completing this project.

Last but not least, we would like to thank everyone who helped and motivated us to work on this project.

TABLE OF CONTENTS

ABSTRACT

Crop production is a highly complicated feature that is influenced by several variables, including genotype, environment, and their interactions, and India has faced significant problems with it over the years. A thorough knowledge of the functional link between yield and various interaction components is essential for accurate yield prediction, and revealing this relationship calls for both large datasets and effective algorithms. The highlighted issue is that the algorithms only pay attention to one or two of the elements affecting crop productivity. The crop's dependence on several elements must be carefully considered. Increases in agricultural yield and economic yield per unit of land will be required, not only in ideal growing environments but also in environments with climatic, water-accessible, and soil-quality restrictions.

Crop output and quality can be increased and improved in a variety of ways. The production of crop yields can also be predicted with the help of data analysis. Data analysis is, in general, the process of looking at data from many angles and distilling it into meaningful information. All these data's patterns, correlations, and interactions may include information.

In this study, data analysis will be used to forecast the suitability of the soil for agricultural planting, assisting farmers in making better decisions and increasing crop production.

PROBLEM STATEMENT AND OBJECTIVES

Although data analytics has already started playing vital roles in the agriculture sector but one of the main problem that has been identified is that the algorithms that have been used only focus on one or two of the factors concerned with the productivity of the crops but especially in India, where farmers do not have great availability of external resources and depend a lot on the natural resources of the Indian weather and India's location, the crop yield depends on numerous factors including those like climate, water availability, and soil quality. Hence in order to solve this gap we are using data analytics in an extended manner to identify the factors on which this production depends and thus, help the farmers in crop choice and the steps to be taken after that.

INTRODUCTION

From the ancient period, agriculture is considered as the main and the most important practice in India. The greenish goods produced in the land which have been taken by the creature leads to a healthy and welfare life. Since the invention of new innovative technologies and techniques the agriculture field is slowly degrading.

Due to these abundant inventions people have been concentrated on cultivating artificial products that are hybrid products where there leads to an unhealthy life. Nowadays, modern people do not have awareness about the cultivation of the crops at the right time and in the right place. Because of these cultivating techniques the seasonal climatic conditions are also being changed against the fundamental assets like soil, water and air which lead to insecurity of food.

Overall, this will benefit the farmers, seed, policy makers in making informed decisions with the help of the predictions made by the models. Being a totally software solution, it does not allow maintenance factor to be considered much. Also, the accuracy level would be high as compared to hardware-based solutions, because components like soil composition, soil type, pH value, weather conditions all come into picture during the prediction process.

The project aims to perform soil sample analysis by normalizing the data sets followed by k-means testing, and visualizing the data by plotting cluster dendrograms and silhouette plots using R, which will be of great help to the farmers as it provides detailed information about the soil type which can be used for selection of suitable crops.

To be precise and accurate in predicting crops, the project analyses the nutrients present in the soil and the crop productivity based on location.

LITERATURE SURVEY

Normalization based K means Clustering Algorithm by Deepali Virmani, Shweta Taneja, Geetika Malhotra

The paper "Normalization based K means Clustering Algorithm" proposes a modified version of the K-means clustering algorithm that uses z-score normalization to address issues of scale and variable correlation in the clustering process. The authors evaluate the algorithm's performance on various datasets and compare it with traditional K-means clustering algorithms. The results show that the proposed algorithm outperforms traditional algorithms in terms of accuracy and convergence rate. The authors conclude that the normalization-based K-means clustering algorithm can be a useful tool for various data mining applications.

Crop forecasting: Its importance, current approaches, ongoing evolution and organizational aspects by Yakob M.Seid

Crop forecasting is crucial in predicting the yield, production, and supply of agricultural crops, enabling farmers, traders, and policymakers to make informed decisions. Current approaches include advanced techniques like remote sensing, machine learning, and simulation models, allowing for more accurate and timely predictions. Organizational aspects involve collaboration between government agencies, research institutions, and private sector players. The ongoing evolution of crop forecasting methods and the emphasis on data sharing will continue to improve the accuracy of crop forecasts and support decision-making in the agricultural industry.

Soil Data Analysis and Crop Yield Prediction in Data Mining using R Programming by K. Samundeeswari, K. Srinivasan

The paper "Soil Data Analysis and Crop Yield Prediction in Data Mining using R-Programming" proposes a methodology for analyzing soil data and predicting crop yield using data mining techniques and R programming. The authors collect soil data from various regions and preprocess it to remove noise and missing values. They then use various data mining techniques, such as correlation analysis and decision tree algorithms, to identify significant factors affecting crop yield. The results show that the proposed methodology can accurately predict crop yield, and the identified significant factors are consistent with existing literature. The authors conclude that the proposed methodology can be a useful tool for agricultural practitioners to make informed decisions regarding crop yield and soil management.

Forecasting of demand using ARIMA model by Jamal Fattah, Latifa Ezzine, Zineb, Aman, Haj El Moussami, and Abdeslam Lachhab

The paper "Forecasting of demand using ARIMA model" by Jamal Fattah, Latifa Ezzine, Zineb Aman, Haj El Moussami, and Abdeslam Lachhab discusses the application of ARIMA models for demand forecasting using real-world data from a Moroccan bakery. The study compares the performance of different ARIMA models and evaluates their accuracy using statistical measures. The results demonstrate the effectiveness of ARIMA models in forecasting demand, and provide practical guidelines for businesses and organizations that rely on demand forecasting for decision making.

What is K-means Clustering in Machine Learning? by Neelam Tyagi

This study states that K-means clustering is an unsupervised learning algorithm used to group a set of data points into K clusters based on their similarities. The algorithm starts by randomly selecting K cluster centers and assigning data points to their nearest centroid based on Euclidean distance. Centroids are then updated to the mean of all data points assigned to them until convergence or a predefined number of iterations. K-means clustering is a popular algorithm for identifying structure in large datasets, but it has limitations such as sensitivity to the initial choice of centroids and number of clusters needed. Extensions to K-means include hierarchical clustering and fuzzy clustering.

Crop prediction using predictive analytics by P. S. Vijayabaskar; R. Sreemathi; E. Keertanaa

The article discusses the use of predictive analytics in crop prediction to help farmers make informed decisions about their crops. The authors explain that predictive analytics involves using historical data and machine learning algorithms to predict future outcomes. They use weather data, soil data, and crop data to predict the yield of crops in a particular area. The authors describe the process of data collection, pre-processing, and analysis to create a model that predicts the yield of crops. The article concludes that predictive analytics can be an effective tool for farmers to optimize their crop yield and minimize losses. The authors suggest that further research is needed to improve the accuracy of the models and to make them more accessible to farmers in developing countries.

Crop Yield Prediction based on Indian Agriculture using Machine Learning by Potnuru Sai Nishant; Pinapa Sai Venkat; Bollu Lakshmi Avinash; B. Jabber

The article presents a machine learning-based approach for crop yield prediction in Indian agriculture. The authors explain that crop yield prediction is an essential task for farmers and policymakers to ensure food security and sustainability. They use historical crop yield data, weather data, and soil data to train a machine learning model. The authors compare the performance of different machine learning algorithms, including decision tree, random forest, and neural network, to select the best model. The article also discusses the importance of feature selection and hyperparameter tuning to improve the accuracy of the model. The authors conclude that machine learning can be an effective tool for crop yield prediction in Indian agriculture, which can help farmers make informed decisions and improve crop productivity.

Analysis of Soil Samples for its Physico-Chemical Parameters from Kadi City by Chandak Nisha, Maiti Barnali, Pathan Shabana and Desai Meena

The article presents an analysis of soil samples collected from Kadi city to determine its physico-chemical parameters. The authors explain that soil analysis is crucial for understanding soil quality, which is essential for sustainable agriculture. The article discusses the process of sample collection and analysis, including the measurement of soil pH, organic matter, nitrogen, phosphorus, and potassium content. The authors compare their findings with the standard values recommended for agricultural use to assess the soil quality in Kadi city. The article concludes that the soil in Kadi city is generally alkaline with a high content of organic matter and nitrogen, but low in phosphorus and potassium. The authors suggest that farmers in the region should use appropriate fertilizers and other soil amendments to improve soil quality and crop productivity. The study provides valuable insights into soil quality in Kadi city and highlights the importance of soil analysis for sustainable agriculture.

MODULE DESCRIPTION

1. **Normalization of our dataset:**

   Normalization is used to eliminate redundant data and ensures that good quality clusters are generated which can improve the efficiency of clustering algorithms. So, it becomes an essential step before clustering as Euclidean distance is very sensitive to the changes in the differences.

   Data Analysis can generate effective results if normalization is applied to the dataset. It is a process used to standardize all the attributes of the dataset and give them equal weight so that redundant or noisy objects can be eliminated and there is valid and reliable data which enhances the accuracy of the result.

2. **Silhouette plot analysis:**

   Silhouette analysis allows you to calculate how similar each observation is with the cluster it is assigned relative to other clusters. This metric (silhouette width) ranges from -1 to 1 for each observation in your data and can be interpreted as follows:

   - Values close to 1 suggest that the observation is well matched to the assigned cluster.

   - Values close to 0 suggest that the observation is borderline matched between two clusters.

   - Values close to -1 suggest that the observations may be assigned to the wrong cluster.

   In our project the final silhouette plot tells us about the best fitted and worst fitted samples in the three clusters, the observations with values close to 1 are the soil samples that are most suitable for the crop plantation, the observation with values close to 0 are the soil samples which are less suitable for plantation, whereas the observations with negative values are not suitable for plantation at all.

3. K-means Clustering:

By specifying the value of k, you are informing the algorithm of how many means or centers you are looking for. Again repeating, if k is equal to 3, the algorithm accounts it for 3 clusters.

Following are the steps for working of the k-means algorithm:

- K-centers are modeled randomly in accordance with the present value of K like it has been implemented in the testing dataset in our project. Value of K has been given as 3.

- K-means assigns each data point in the dataset to the adjacent center and attempts to curtail Euclidean distance between data points. Data points are assumed to be present in the peculiar cluster as if it is closer to the center of that cluster than any other cluster center. We have calculated the same in the project which gives us the similarity matrix, to be used for further calculations.

- After that, k-means determines the center by accounting the mean of all data points referred to that cluster center. It reduces the complete variance of the intra-clusters with respect to the prior step. Here, the "means" defines the average of data points and identifies a new center in the method of k-means clustering. An appropriate number of iterations is attained, no variation in the value of cluster center or no change in the cluster due to data points.

4. Hierarchical Clustering:

Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly like each other.

Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps:

- Identify the two clusters that are closest together, and
- Merge the two most similar clusters. This iterative process continues until all the clusters are merged.

In our project the clustering has been done with both complete linkage and center linkage in our project to determine which gives better results.

5. Time Series Analysis (ARIMA Model):

Using ARIMA model, you can forecast a time series using the series past values.

ARIMA, short for ‘Auto Regressive Integrated Moving Average’ is a class of models that ‘explains’ a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.

Any ‘non-seasonal’ time series that exhibits patterns and is not a random white noise can be modeled with ARIMA models.

An ARIMA model is characterized by 3 terms: p, d, q
where,

- p is the order of the AR term
- q is the order of the MA term
- d is the number of differencing required to make the time series station

ALGORITHM USED

1. **K-Means Clustering:**

   K-means clustering is a popular unsupervised learning algorithm used in machine learning and data mining. It is used to partition a given dataset into K clusters, where K is a user-specified number of clusters. The algorithm works by first selecting K initial centroids randomly from the dataset. Each data point is then assigned to the nearest centroid based on the Euclidean distance between the data point and the centroid.

   Once all data points have been assigned to clusters, the centroids are re-calculated as the mean of all data points in the cluster. This process of re-assigning data points to clusters based on new centroids and re-calculating centroids continues until the algorithm converges, i.e., the centroids stop moving significantly.

   K-means clustering has several advantages, including its simplicity and efficiency. It is also a scalable algorithm and can handle large datasets with a large number of variables. However, the algorithm requires the user to specify the number of clusters beforehand, which can be a challenge when the optimal number of clusters is not known. Additionally, K-means clustering can produce different results depending on the initial random selection of centroids.

2. **Hierarchical Clustering:**

   Hierarchical clustering is a popular unsupervised learning technique used in data analysis and machine learning. It is used to group similar data points together based on their similarity or distance metrics. In this approach, the data is represented in a tree-like structure or a dendrogram.

   There are two main types of hierarchical clustering: agglomerative and divisive. Agglomerative clustering starts with individual data points and then iteratively merges similar clusters until only one cluster is left. Divisive clustering, on the other hand, starts with all data points in one cluster and then recursively divides the cluster into smaller and more distinct sub-clusters.

The result of hierarchical clustering is a dendrogram that shows the hierarchical relationships between the data points. The branches of the dendrogram represent the clusters at different levels of granularity, with the leaves representing individual data points.

Hierarchical clustering has several advantages over other clustering algorithms. It does not require the number of clusters to be predefined, making it more flexible. It also provides a visual representation of the data, allowing for easier interpretation and analysis. However, it can be computationally expensive, especially for large datasets, and the choice of distance metric and linkage method can significantly affect the results.

3. **ARIMA Model**:

ARIMA stands for Autoregressive Integrated Moving Average, which is a statistical model used for time series analysis and forecasting. It is a combination of three components: autoregression (AR), integration (I), and moving average (MA).

The AR component of ARIMA refers to the use of past observations to predict future values. In this component, the model assumes that the future value of the series is a linear combination of its past values. The number of past values to include in the model is determined by the order of the autoregressive parameter (p).

The I component refers to the use of differencing to remove any trends or seasonality in the data. This involves subtracting the current value of the series from the previous value, which helps to stabilize the variance of the series. The order of differencing (d) is determined by the number of times the differencing is performed.

The MA component refers to the use of past errors to predict future values. In this component, the model assumes that the future value of the series is a linear combination of its past errors. The order of the moving average parameter (q) determines the number of past errors to include in the model.

The ARIMA model is typically denoted as ARIMA (p, d, q). It is widely used in financial forecasting, economics, and other fields where trends and seasonality in time series data need to be identified and analysed. The model can be further extended to include exogenous variables, resulting in the ARIMAX model, which is useful when external factors are known to affect the time series.

DETAILED ARCHITECTURE

The architecture of our project involves several steps and components, including-

1. Data collection: This involves collecting data on various soil and environmental factors, such as pH, EC, X.O.C, Available Nitrogen, phosphorous and potassium, copper, iron, manganese, and sulphur. Time series data is also collected so that we can apply ARIMA model on it and make future predictions.

2. Data pre-processing: The collected data is pre-processed to remove any errors, missing values, or outliers. This step involves data cleaning, normalization, and transformation.

3. Feature selection: Relevant features are selected from the pre-processed data in our project, based on their importance in predicting crop yield.

4. Machine learning models: Various machine learning models are used to predict crop yield based on the selected features. These models include k-means clustering, hierarchical clustering and ARIMA model as well.

5. Model evaluation: The performance of the machine learning models is evaluated using various metrics, such as mean absolute error, mean squared error, and coefficient of determination.

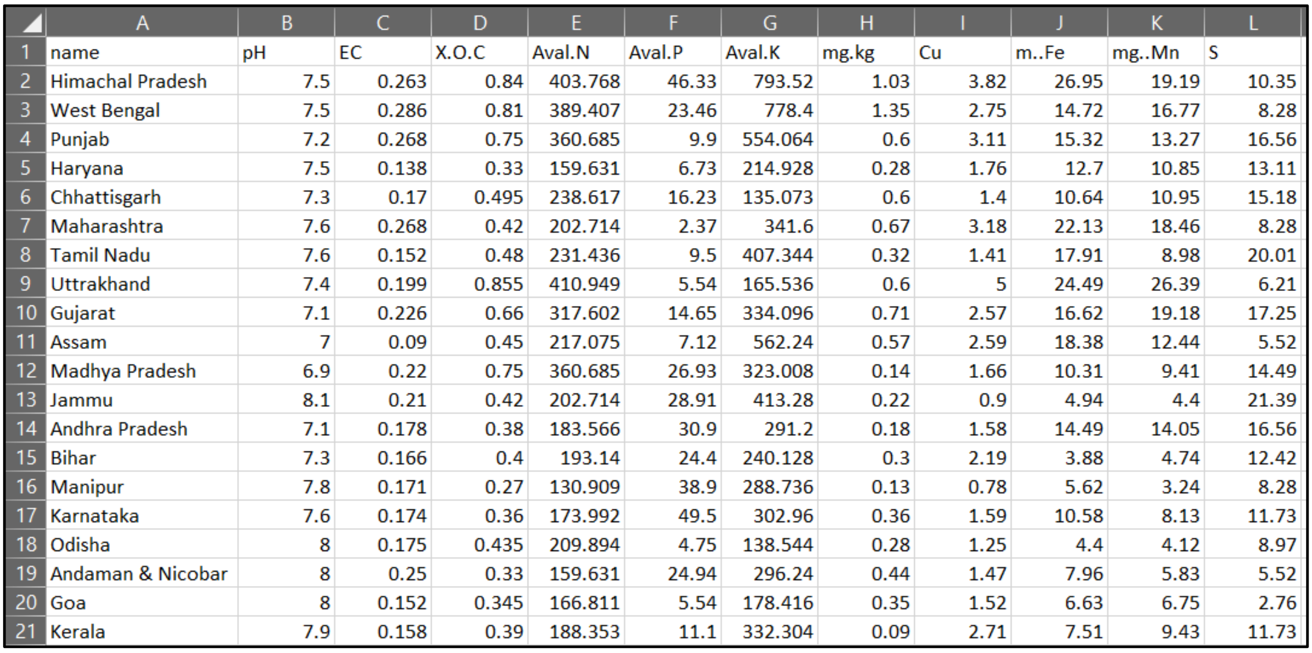
6. Crop yield prediction: The selected model is used to predict the crop yield based on the input features, such as soil and environmental factors.

7. Result visualization: The predicted crop yield can be visualized using graphs or charts to provide insights into the relationships between soil and environmental factors and crop yield. Silhouette plot analysis is used as well.

Overall, the architecture of our project: soil analysis and crop yield prediction involve collecting data, pre-processing, and selecting features, building machine learning models, evaluating model performance, predicting crop yield, and visualizing results.

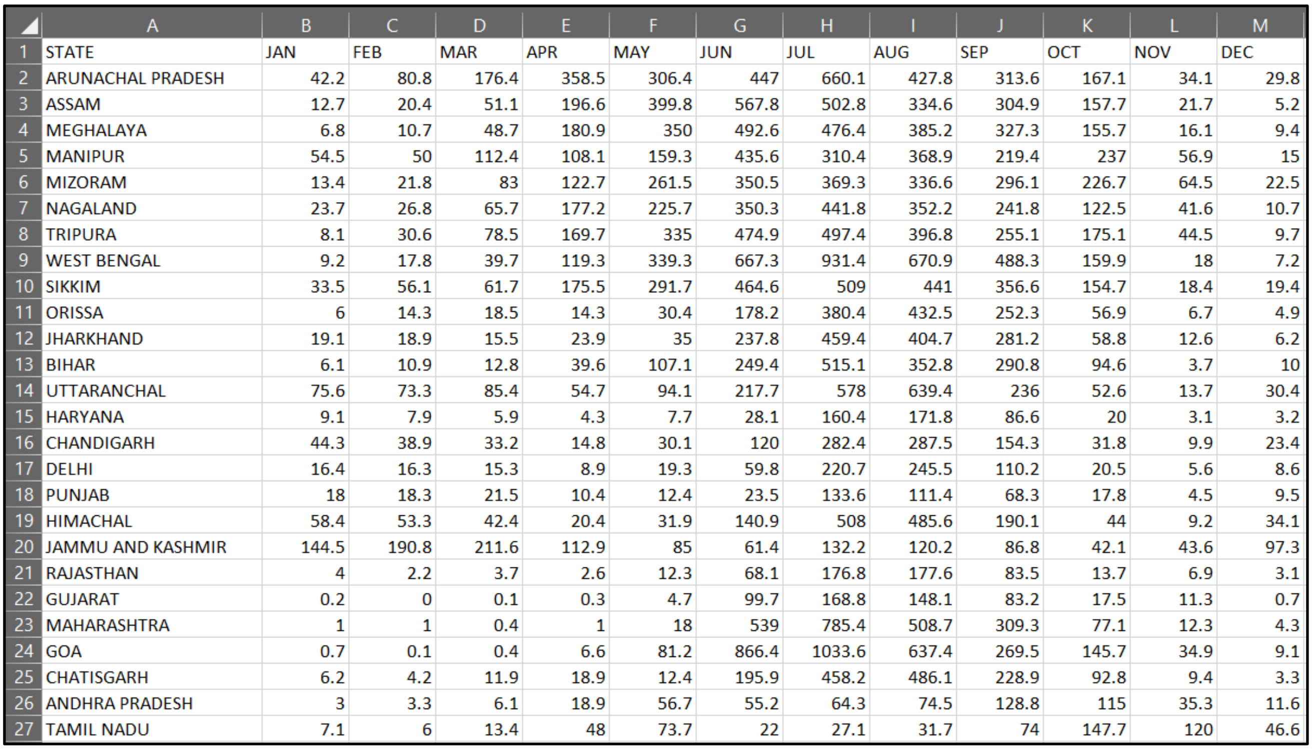
DATASETS

Dataset 1: State-wise dataset with information of nutrient content of soil



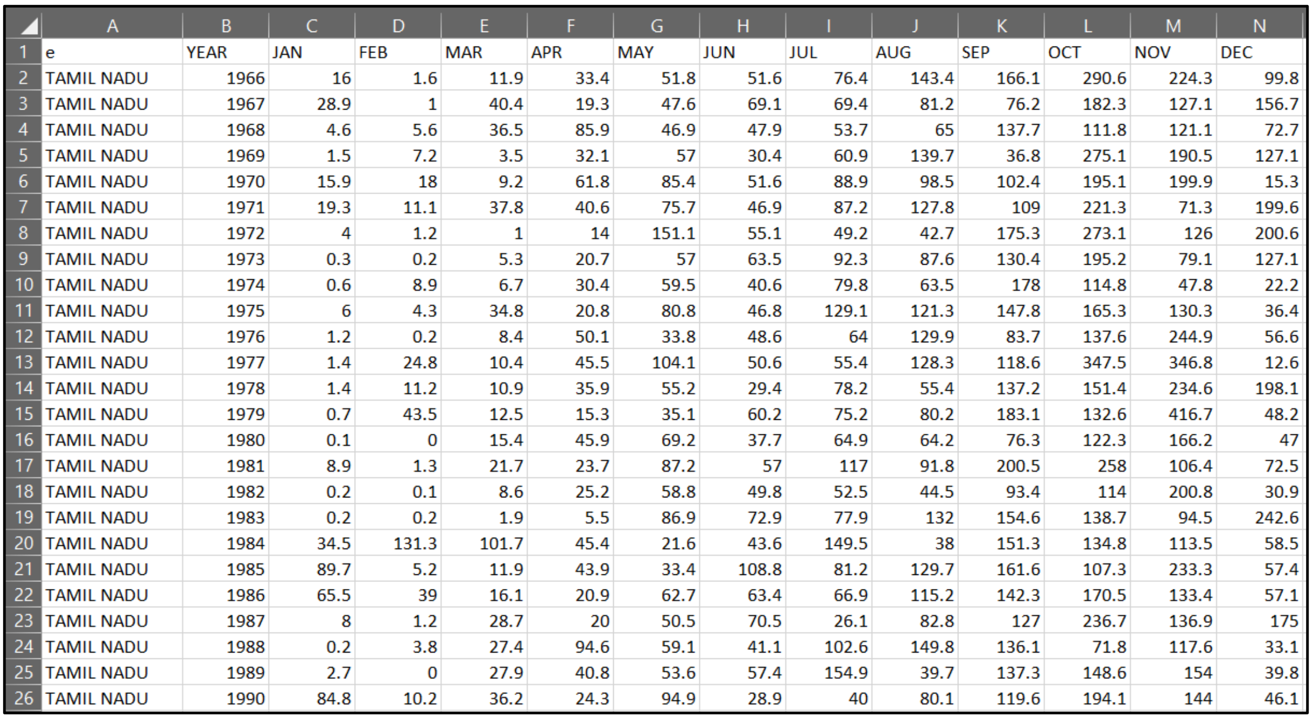
	A	B	C	D	E	F	G	H	I	J	K	L
1	name	pH	EC	X.O.C	Aval.N	Aval.P	Aval.K	mg.kg	Cu	m..Fe	mg..Mn	S
2	Himachal Pradesh	7.5	0.263	0.84	403.768	46.33	793.52	1.03	3.82	26.95	19.19	10.35
3	West Bengal	7.5	0.286	0.81	389.407	23.46	778.4	1.35	2.75	14.72	16.77	8.28
4	Punjab	7.2	0.268	0.75	360.685	9.9	554.064	0.6	3.11	15.32	13.27	16.56
5	Haryana	7.5	0.138	0.33	159.631	6.73	214.928	0.28	1.76	12.7	10.85	13.11
6	Chhattisgarh	7.3	0.17	0.495	238.617	16.23	135.073	0.6	1.4	10.64	10.95	15.18
7	Maharashtra	7.6	0.268	0.42	202.714	2.37	341.6	0.67	3.18	22.13	18.46	8.28
8	Tamil Nadu	7.6	0.152	0.48	231.436	9.5	407.344	0.32	1.41	17.91	8.98	20.01
9	Uttrakhand	7.4	0.199	0.855	410.949	5.54	165.536	0.6	5	24.49	26.39	6.21
10	Gujarat	7.1	0.226	0.66	317.602	14.65	334.096	0.71	2.57	16.62	19.18	17.25
11	Assam	7	0.09	0.45	217.075	7.12	562.24	0.57	2.59	18.38	12.44	5.52
12	Madhya Pradesh	6.9	0.22	0.75	360.685	26.93	323.008	0.14	1.66	10.31	9.41	14.49
13	Jammu	8.1	0.21	0.42	202.714	28.91	413.28	0.22	0.9	4.94	4.4	21.39
14	Andhra Pradesh	7.1	0.178	0.38	183.566	30.9	291.2	0.18	1.58	14.49	14.05	16.56
15	Bihar	7.3	0.166	0.4	193.14	24.4	240.128	0.3	2.19	3.88	4.74	12.42
16	Manipur	7.8	0.171	0.27	130.909	38.9	288.736	0.13	0.78	5.62	3.24	8.28
17	Karnataka	7.6	0.174	0.36	173.992	49.5	302.96	0.36	1.59	10.58	8.13	11.73
18	Odisha	8	0.175	0.435	209.894	4.75	138.544	0.28	1.25	4.4	4.12	8.97
19	Andaman & Nicobar	8	0.25	0.33	159.631	24.94	296.24	0.44	1.47	7.96	5.83	5.52
20	Goa	8	0.152	0.345	166.811	5.54	178.416	0.35	1.52	6.63	6.75	2.76
21	Kerala	7.9	0.158	0.39	188.353	11.1	332.304	0.09	2.71	7.51	9.43	11.73

Dataset 2: State-wise dataset with information of rainfall throughout the year



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	STATE	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
2	ARUNACHAL PRADESH	42.2	80.8	176.4	358.5	306.4	447	660.1	427.8	313.6	167.1	34.1	29.8
3	ASSAM	12.7	20.4	51.1	196.6	399.8	567.8	502.8	334.6	304.9	157.7	21.7	5.2
4	MEGHALAYA	6.8	10.7	48.7	180.9	350	492.6	476.4	385.2	327.3	155.7	16.1	9.4
5	MANIPUR	54.5	50	112.4	108.1	159.3	435.6	310.4	368.9	219.4	237	56.9	15
6	MIZORAM	13.4	21.8	83	122.7	261.5	350.5	369.3	336.6	296.1	226.7	64.5	22.5
7	NAGALAND	23.7	26.8	65.7	177.2	225.7	350.3	441.8	352.2	241.8	122.5	41.6	10.7
8	TRIPURA	8.1	30.6	78.5	169.7	335	474.9	497.4	396.8	255.1	175.1	44.5	9.7
9	WEST BENGAL	9.2	17.8	39.7	119.3	339.3	667.3	931.4	670.9	488.3	159.9	18	7.2
10	SIKKIM	33.5	56.1	61.7	175.5	291.7	464.6	509	441	356.6	154.7	18.4	19.4
11	ORISSA	6	14.3	18.5	14.3	30.4	178.2	380.4	432.5	252.3	56.9	6.7	4.9
12	JHARKHAND	19.1	18.9	15.5	23.9	35	237.8	459.4	404.7	281.2	58.8	12.6	6.2
13	BIHAR	6.1	10.9	12.8	39.6	107.1	249.4	515.1	352.8	290.8	94.6	3.7	10
14	UTTARANCHAL	75.6	73.3	85.4	54.7	94.1	217.7	578	639.4	236	52.6	13.7	30.4
15	HARYANA	9.1	7.9	5.9	4.3	7.7	28.1	160.4	171.8	86.6	20	3.1	3.2
16	CHANDIGARH	44.3	38.9	33.2	14.8	30.1	120	282.4	287.5	154.3	31.8	9.9	23.4
17	DELHI	16.4	16.3	15.3	8.9	19.3	59.8	220.7	245.5	110.2	20.5	5.6	8.6
18	PUNJAB	18	18.3	21.5	10.4	12.4	23.5	133.6	111.4	68.3	17.8	4.5	9.5
19	HIMACHAL	58.4	53.3	42.4	20.4	31.9	140.9	508	485.6	190.1	44	9.2	34.1
20	JAMMU AND KASHMIR	144.5	190.8	211.6	112.9	85	61.4	132.2	120.2	86.8	42.1	43.6	97.3
21	RAJASTHAN	4	2.2	3.7	2.6	12.3	68.1	176.8	177.6	83.5	13.7	6.9	3.1
22	GUJARAT	0.2	0	0.1	0.3	4.7	99.7	168.8	148.1	83.2	17.5	11.3	0.7
23	MAHARASHTRA	1	1	0.4	1	18	539	785.4	508.7	309.3	77.1	12.3	4.3
24	GOA	0.7	0.1	0.4	6.6	81.2	866.4	1033.6	637.4	269.5	145.7	34.9	9.1
25	CHATISGARH	6.2	4.2	11.9	18.9	12.4	195.9	458.2	486.1	228.9	92.8	9.4	3.3
26	ANDHRA PRADESH	3	3.3	6.1	18.9	56.7	55.2	64.3	74.5	128.8	115	35.3	11.6
27	TAMIL NADU	7.1	6	13.4	48	73.7	22	27.1	31.7	74	147.7	120	46.6

Dataset 3: Dataset with information of rainfall throughout the year in Tamil Nadu

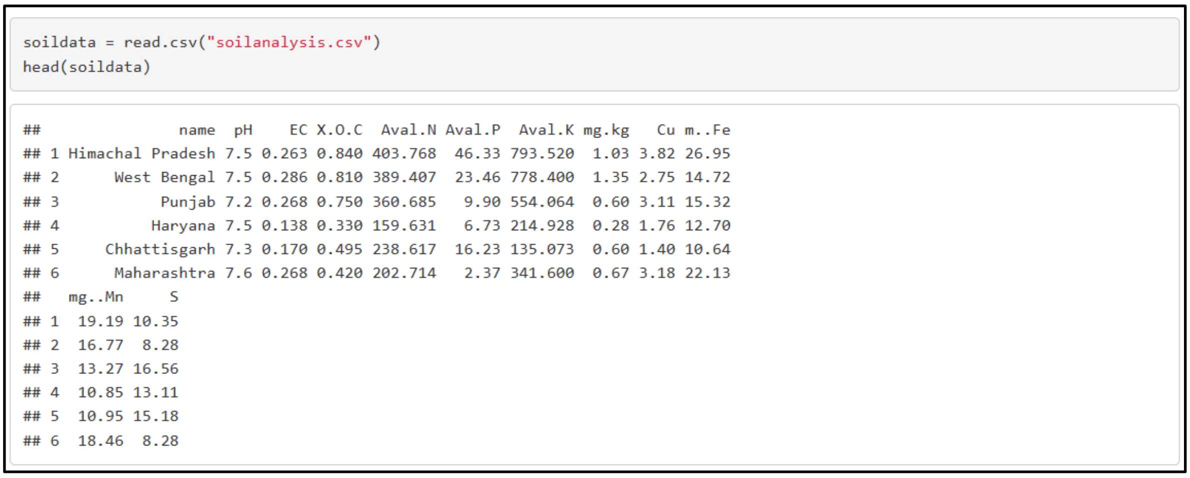


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	e	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
2	TAMIL NADU	1966	16	1.6	11.9	33.4	51.8	51.6	76.4	143.4	166.1	290.6	224.3	99.8
3	TAMIL NADU	1967	28.9	1	40.4	19.3	47.6	69.1	69.4	81.2	76.2	182.3	127.1	156.7
4	TAMIL NADU	1968	4.6	5.6	36.5	85.9	46.9	47.9	53.7	65	137.7	111.8	121.1	72.7
5	TAMIL NADU	1969	1.5	7.2	3.5	32.1	57	30.4	60.9	139.7	36.8	275.1	190.5	127.1
6	TAMIL NADU	1970	15.9	18	9.2	61.8	85.4	51.6	88.9	98.5	102.4	195.1	199.9	15.3
7	TAMIL NADU	1971	19.3	11.1	37.8	40.6	75.7	46.9	87.2	127.8	109	221.3	71.3	199.6
8	TAMIL NADU	1972	4	1.2	1	14	151.1	55.1	49.2	42.7	175.3	273.1	126	200.6
9	TAMIL NADU	1973	0.3	0.2	5.3	20.7	57	63.5	92.3	87.6	130.4	195.2	79.1	127.1
10	TAMIL NADU	1974	0.6	8.9	6.7	30.4	59.5	40.6	79.8	63.5	178	114.8	47.8	22.2
11	TAMIL NADU	1975	6	4.3	34.8	20.8	80.8	46.8	129.1	121.3	147.8	165.3	130.3	36.4
12	TAMIL NADU	1976	1.2	0.2	8.4	50.1	33.8	48.6	64	129.9	83.7	137.6	244.9	56.6
13	TAMIL NADU	1977	1.4	24.8	10.4	45.5	104.1	50.6	55.4	128.3	118.6	347.5	346.8	12.6
14	TAMIL NADU	1978	1.4	11.2	10.9	35.9	55.2	29.4	78.2	55.4	137.2	151.4	234.6	198.1
15	TAMIL NADU	1979	0.7	43.5	12.5	15.3	35.1	60.2	75.2	80.2	183.1	132.6	416.7	48.2
16	TAMIL NADU	1980	0.1	0	15.4	45.9	69.2	37.7	64.9	64.2	76.3	122.3	166.2	47
17	TAMIL NADU	1981	8.9	1.3	21.7	23.7	87.2	57	117	91.8	200.5	258	106.4	72.5
18	TAMIL NADU	1982	0.2	0.1	8.6	25.2	58.8	49.8	52.5	44.5	93.4	114	200.8	30.9
19	TAMIL NADU	1983	0.2	0.2	1.9	5.5	86.9	72.9	77.9	132	154.6	138.7	94.5	242.6
20	TAMIL NADU	1984	34.5	131.3	101.7	45.4	21.6	43.6	149.5	38	151.3	134.8	113.5	58.5
21	TAMIL NADU	1985	89.7	5.2	11.9	43.9	33.4	108.8	81.2	129.7	161.6	107.3	233.3	57.4
22	TAMIL NADU	1986	65.5	39	16.1	20.9	62.7	63.4	66.9	115.2	142.3	170.5	133.4	57.1
23	TAMIL NADU	1987	8	1.2	28.7	20	50.5	70.5	26.1	82.8	127	236.7	136.9	175
24	TAMIL NADU	1988	0.2	3.8	27.4	94.6	59.1	41.1	102.6	149.8	136.1	71.8	117.6	33.1
25	TAMIL NADU	1989	2.7	0	27.9	40.8	53.6	57.4	154.9	39.7	137.3	148.6	154	39.8
26	TAMIL NADU	1990	84.8	10.2	36.2	24.3	94.9	28.9	40	80.1	119.6	194.1	144	46.1

RESULTS AND DISCUSSIONS

Output with description and inference:

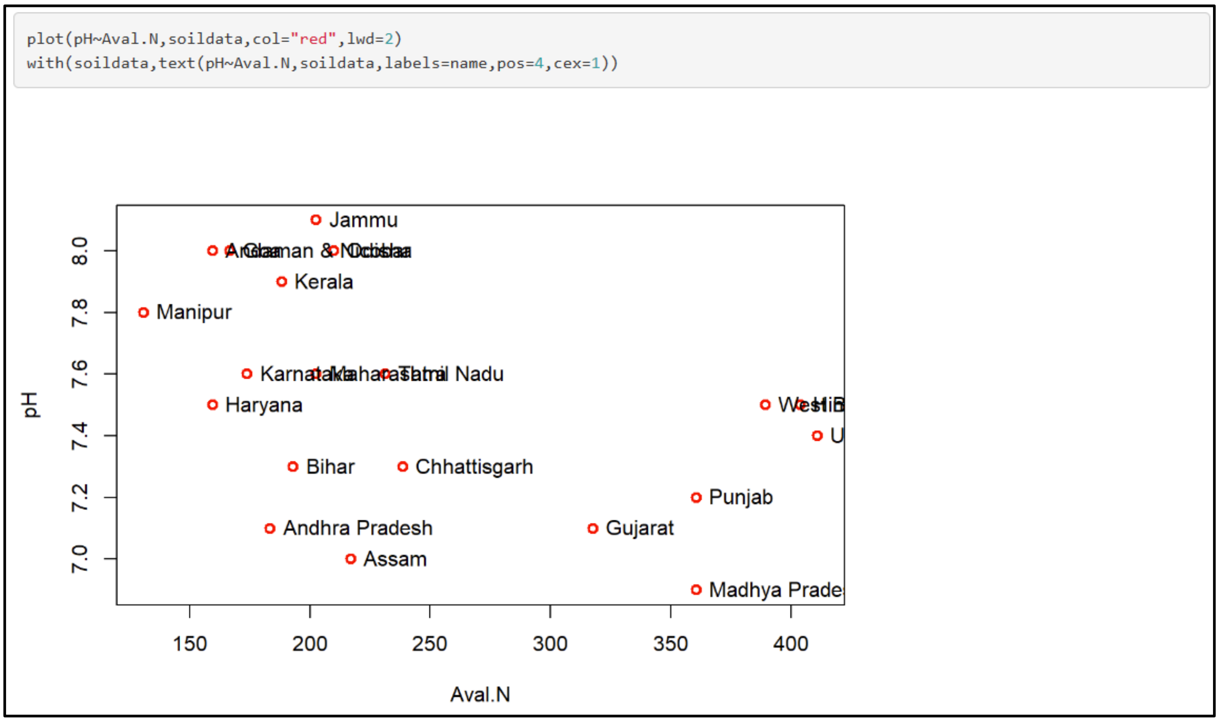
- **R outputs**



```
soildata = read.csv("soilanalysis.csv")
head(soildata)
```

```
##               name  pH    EC X.O.C  Aval.N Aval.P  Aval.K mg.kg   Cu m..Fe
## 1 Himachal Pradesh 7.5 0.263 0.840 403.768  46.33 793.520  1.03 3.82 26.95
## 2      West Bengal 7.5 0.286 0.810 389.407  23.46 778.400  1.35 2.75 14.72
## 3           Punjab 7.2 0.268 0.750 360.685   9.90 554.064  0.60 3.11 15.32
## 4          Haryana 7.5 0.138 0.330 159.631   6.73 214.928  0.28 1.76 12.70
## 5     Chhattisgarh 7.3 0.170 0.495 238.617  16.23 135.073  0.60 1.40 10.64
## 6      Maharashtra 7.6 0.268 0.420 202.714   2.37 341.600  0.67 3.18 22.13
##   mg..Mn     S
## 1  19.19 10.35
## 2  16.77  8.28
## 3  13.27 16.56
## 4  10.85 13.11
## 5  10.95 15.18
## 6  18.46  8.28
```

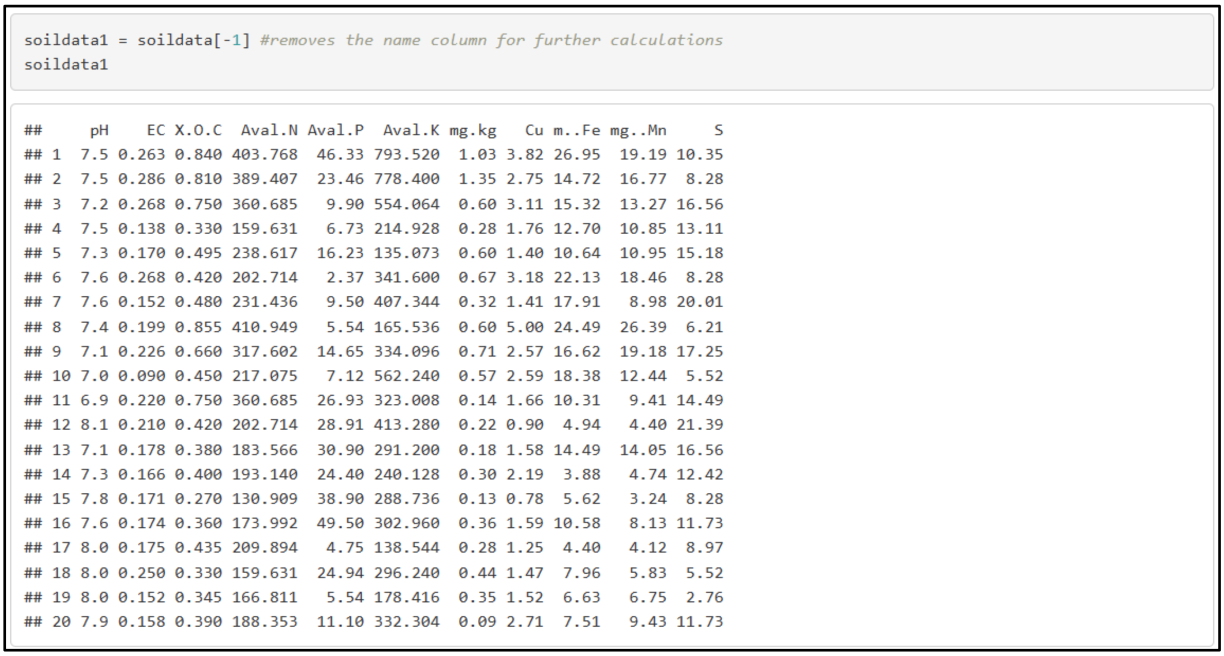
The above picture shows the head of our soil dataset and the various features of the soil that we will be using for the analysis.



```
plot(pH~Aval.N,soildata,col="red",lwd=2)
with(soildata,text(pH~Aval.N,soildata,labels=name,pos=4,cex=1))
```

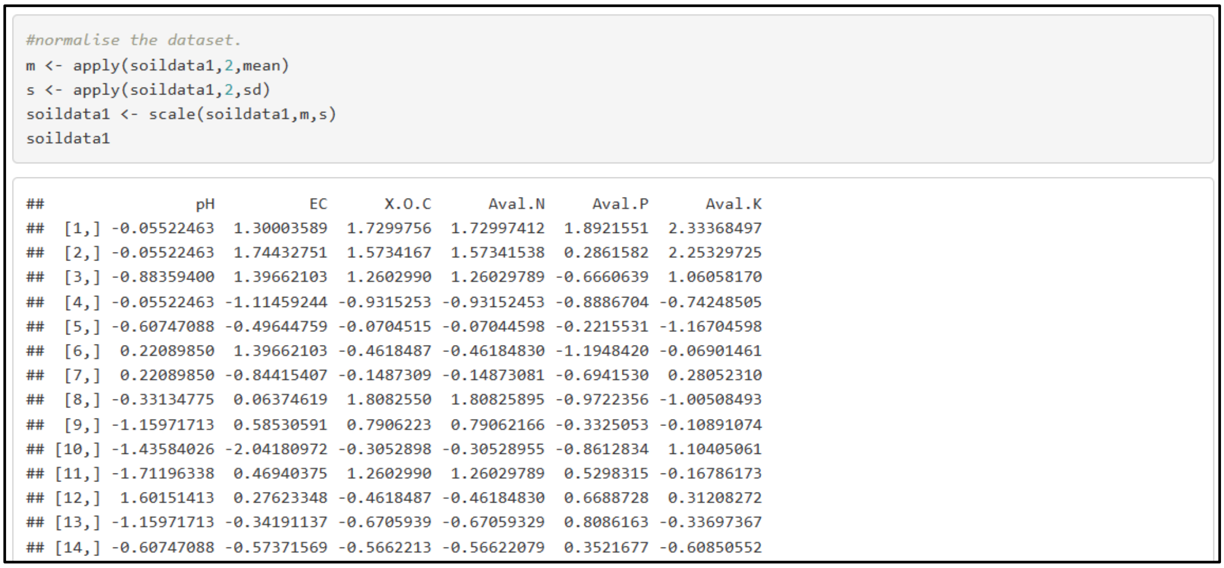
In the above plot, we have selected the feature "available nitrogen" and "pH value" of the soil and displayed their relation for all the states in the dataset. This is important as the more the nitrogen value available in a soil higher is the soil fertility. Hence, we can determine which soil can grow which kind of crops.

Next, we have removed the first column of the dataset, i.e., the name of the state, so that only the numerical values of the dataset are present. Then, we have normalised the dataset so that we can perform our further calculations.



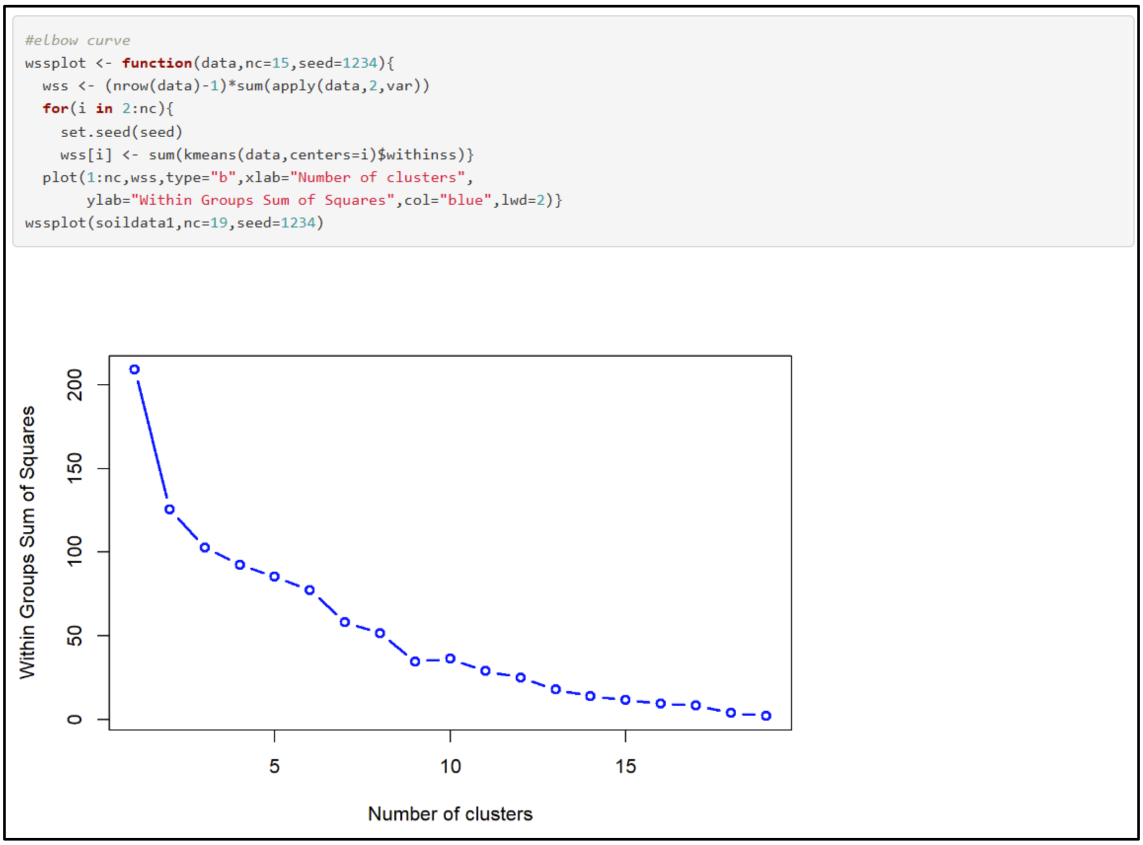
```
soildata1 = soildata[-1] #removes the name column for further calculations
soildata1
```

```
##     pH    EC X.O.C  Aval.N Aval.P  Aval.K mg.kg   Cu m..Fe mg..Mn     S
## 1  7.5 0.263 0.840 403.768  46.33 793.520  1.03 3.82 26.95  19.19 10.35
## 2  7.5 0.286 0.810 389.407  23.46 778.400  1.35 2.75 14.72  16.77  8.28
## 3  7.2 0.268 0.750 360.685   9.90 554.064  0.60 3.11 15.32  13.27 16.56
## 4  7.5 0.138 0.330 159.631   6.73 214.928  0.28 1.76 12.70  10.85 13.11
## 5  7.3 0.170 0.495 238.617  16.23 135.073  0.60 1.40 10.64  10.95 15.18
## 6  7.6 0.268 0.420 202.714   2.37 341.600  0.67 3.18 22.13  18.46  8.28
## 7  7.6 0.152 0.480 231.436   9.50 407.344  0.32 1.41 17.91   8.98 20.01
## 8  7.4 0.199 0.855 410.949   5.54 165.536  0.60 5.00 24.49  26.39  6.21
## 9  7.1 0.226 0.660 317.602  14.65 334.096  0.71 2.57 16.62  19.18 17.25
## 10 7.0 0.090 0.450 217.075   7.12 562.240  0.57 2.59 18.38  12.44  5.52
## 11 6.9 0.220 0.750 360.685  26.93 323.008  0.14 1.66 10.31   9.41 14.49
## 12 8.1 0.210 0.420 202.714  28.91 413.280  0.22 0.90  4.94   4.40 21.39
## 13 7.1 0.178 0.380 183.566  30.90 291.200  0.18 1.58 14.49  14.05 16.56
## 14 7.3 0.166 0.400 193.140  24.40 240.128  0.30 2.19  3.88   4.74 12.42
## 15 7.8 0.171 0.270 130.909  38.90 288.736  0.13 0.78  5.62   3.24  8.28
## 16 7.6 0.174 0.360 173.992  49.50 302.960  0.36 1.59 10.58   8.13 11.73
## 17 8.0 0.175 0.435 209.894   4.75 138.544  0.28 1.25  4.40   4.12  8.97
## 18 8.0 0.250 0.330 159.631  24.94 296.240  0.44 1.47  7.96   5.83  5.52
## 19 8.0 0.152 0.345 166.811   5.54 178.416  0.35 1.52  6.63   6.75  2.76
## 20 7.9 0.158 0.390 188.353  11.10 332.304  0.09 2.71  7.51   9.43 11.73
```



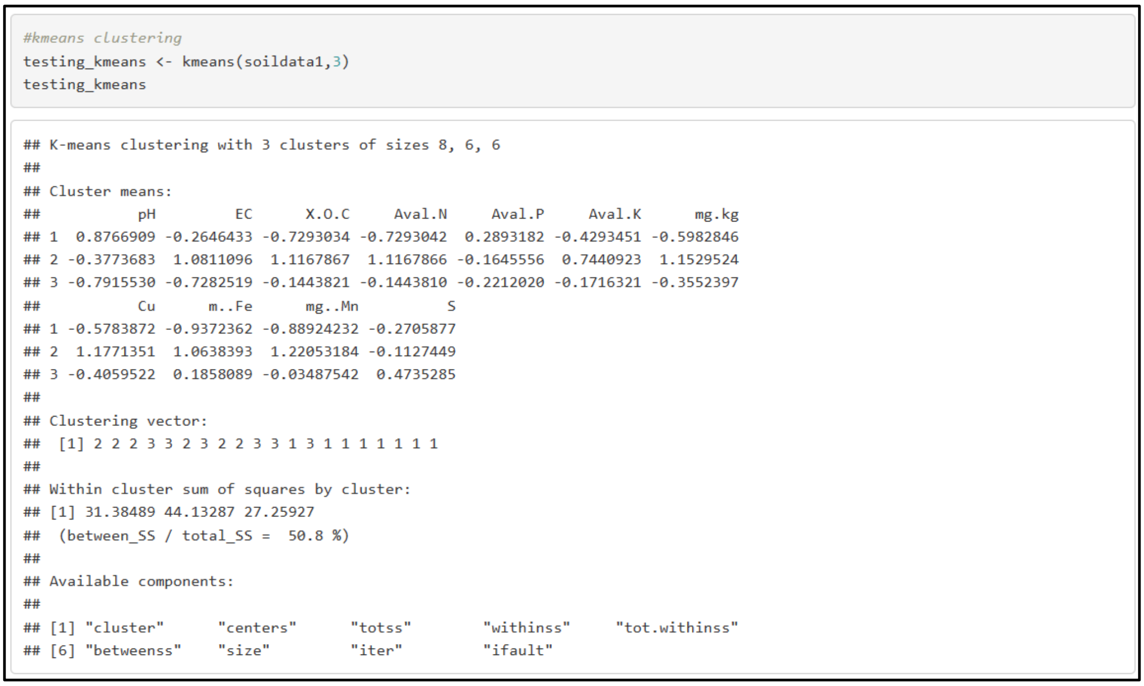
```
#normalise the dataset.
m <- apply(soildata1,2,mean)
s <- apply(soildata1,2,sd)
soildata1 <- scale(soildata1,m,s)
soildata1
```

```
##                pH          EC      X.O.C      Aval.N     Aval.P      Aval.K
##  [1,] -0.05522463  1.30003589  1.7299756  1.72997412  1.8921551  2.33368497
##  [2,] -0.05522463  1.74432751  1.5734167  1.57341538  0.2861582  2.25329725
##  [3,] -0.88359400  1.39662103  1.2602990  1.26029789 -0.6660639  1.06058170
##  [4,] -0.05522463 -1.11459244 -0.9315253 -0.93152453 -0.8886704 -0.74248505
##  [5,] -0.60747088 -0.49644759 -0.0704515 -0.07044598 -0.2215531 -1.16704598
##  [6,]  0.22089850  1.39662103 -0.4618487 -0.46184830 -1.1948420 -0.06901461
##  [7,]  0.22089850 -0.84415407 -0.1487309 -0.14873081 -0.6941530  0.28052310
##  [8,] -0.33134775  0.06374619  1.8082550  1.80825895 -0.9722356 -1.00508493
##  [9,] -1.15971713  0.58530591  0.7906223  0.79062166 -0.3325053 -0.10891074
## [10,] -1.43584026 -2.04180972 -0.3052898 -0.30528955 -0.8612834  1.10405061
## [11,] -1.71196338  0.46940375  1.2602990  1.26029789  0.5298315 -0.16786173
## [12,]  1.60151413  0.27623348 -0.4618487 -0.46184830  0.6688728  0.31208272
## [13,] -1.15971713 -0.34191137 -0.6705939 -0.67059329  0.8086163 -0.33697367
## [14,] -0.60747088 -0.57371569 -0.5662213 -0.56622079  0.3521677 -0.60850552
```



```
#elbow curve
wssplot <- function(data,nc=15,seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for(i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data,centers=i)$withinss)}
  plot(1:nc,wss,type="b",xlab="Number of clusters",
       ylab="Within Groups Sum of Squares",col="blue",lwd=2)}
wssplot(soildata1,nc=19,seed=1234)
```

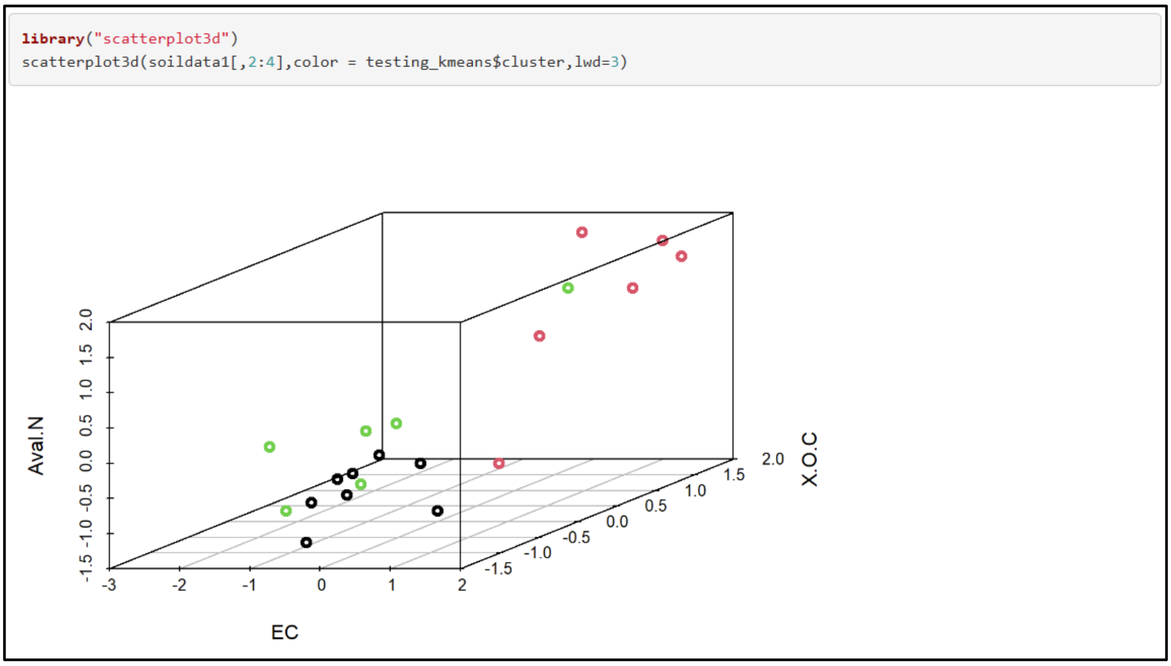
In the above diagram, we have plotted the WCSS values of the dataset with respect to the number of clusters. From this plot, we have inferred that the optimal number of clusters for our dataset is three.



```
#kmeans clustering
testing_kmeans <- kmeans(soildata1,3)
testing_kmeans
```

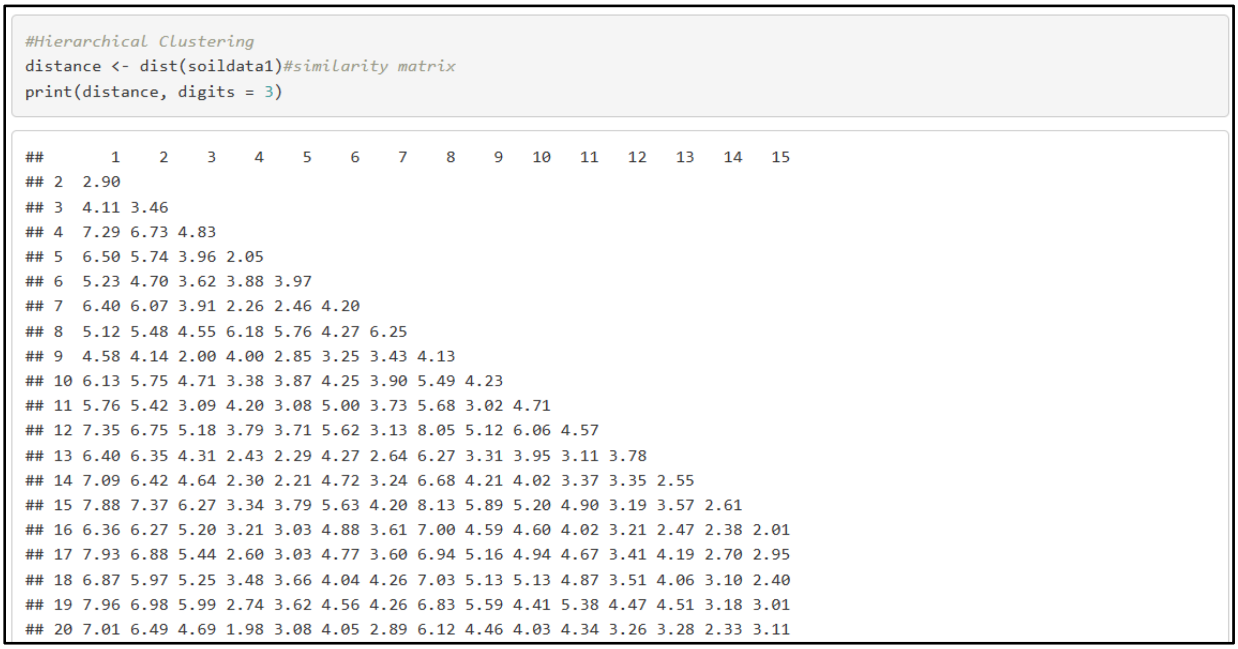
```
## K-means clustering with 3 clusters of sizes 8, 6, 6
##
## Cluster means:
##           pH         EC      X.O.C     Aval.N     Aval.P     Aval.K      mg.kg
## 1  0.8766909 -0.2646433 -0.7293034 -0.7293042  0.2893182 -0.4293451 -0.5982846
## 2 -0.3773683  1.0811096  1.1167867  1.1167866 -0.1645556  0.7440923  1.1529524
## 3 -0.7915530 -0.7282519 -0.1443821 -0.1443810 -0.2212020 -0.1716321 -0.3552397
##           Cu      m..Fe      mg..Mn          S
## 1 -0.5783872 -0.9372362 -0.88924232 -0.2705877
## 2  1.1771351  1.0638393  1.22053184 -0.1127449
## 3 -0.4059522  0.1858089 -0.03487542  0.4735285
##
## Clustering vector:
##  [1] 2 2 2 3 3 2 3 2 2 3 3 1 3 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 31.38489 44.13287 27.25927
##  (between_SS / total_SS =  50.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Then, we have performed K-means algorithm on the dataset choosing the number of clusters as 3 which we had found out in the previous step. As our result, each of the states are assigned to any one cluster and the WCSS value is also displayed. The WCSS value for the three clusters is 31.38489, 44.13287 and 27.25927 respectively. The ratio of the between sum of square of errors to the total sum of square of errors is also displayed and has come out to be 50.8%.



```
library("scatterplot3d")
scatterplot3d(soildata1[,2:4],color = testing_kmeans$cluster,lwd=3)
```

In the above diagram, we have plotted the clusters using scatterplot in 3 dimensions. The clusters are visible in greed, red and black colours respectively.

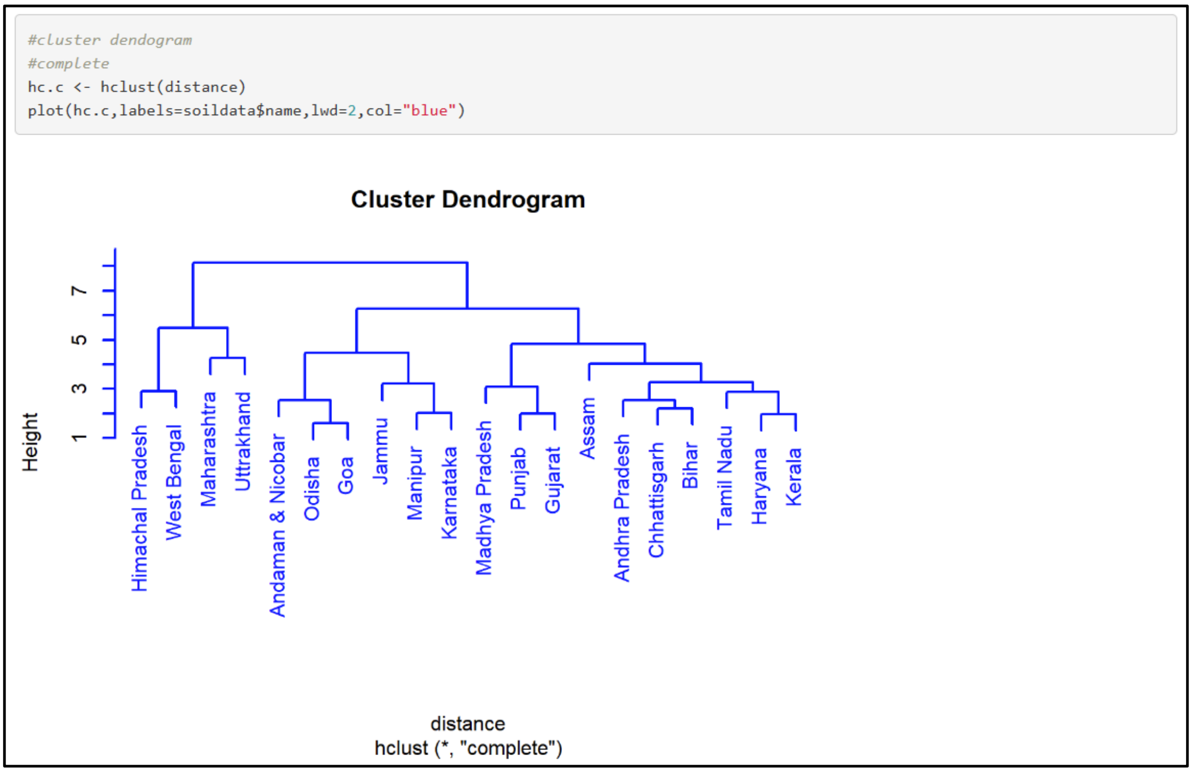


```
#Hierarchical Clustering
distance <- dist(soildata1)#similarity matrix
print(distance, digits = 3)
```

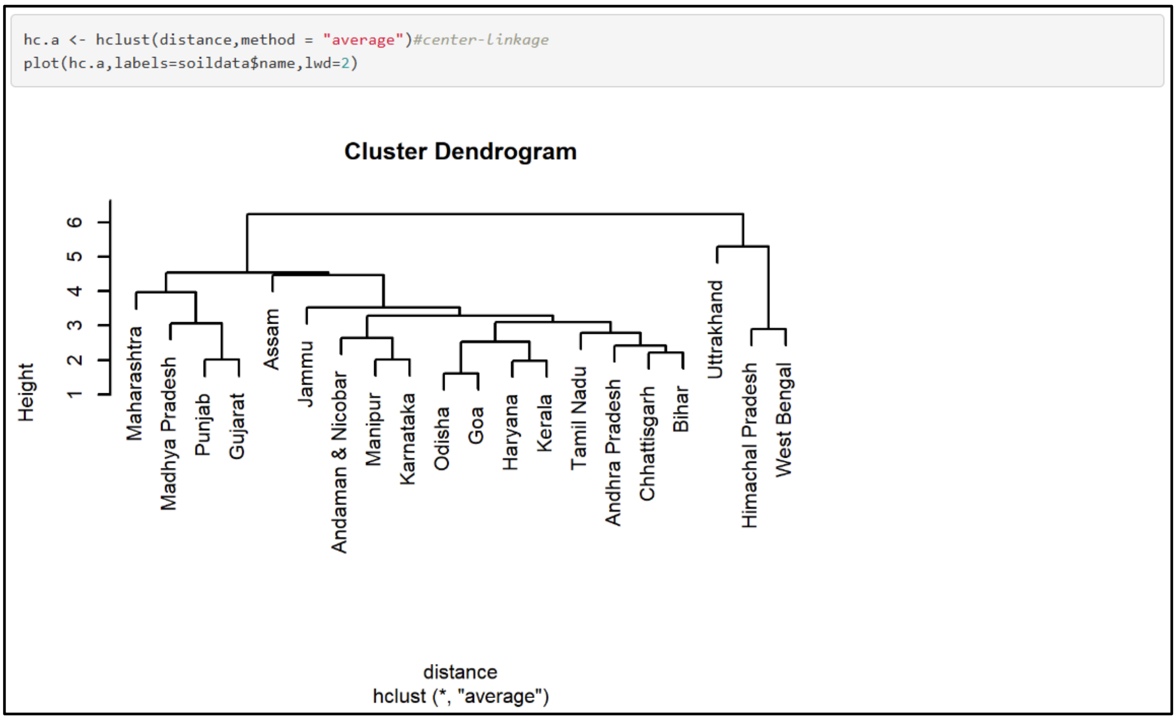
```
##       1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 2  2.90
## 3  4.11 3.46
## 4  7.29 6.73 4.83
## 5  6.50 5.74 3.96 2.05
## 6  5.23 4.70 3.62 3.88 3.97
## 7  6.40 6.07 3.91 2.26 2.46 4.20
## 8  5.12 5.48 4.55 6.18 5.76 4.27 6.25
## 9  4.58 4.14 2.00 4.00 2.85 3.25 3.43 4.13
## 10 6.13 5.75 4.71 3.38 3.87 4.25 3.90 5.49 4.23
## 11 5.76 5.42 3.09 4.20 3.08 5.00 3.73 5.68 3.02 4.71
## 12 7.35 6.75 5.18 3.79 3.71 5.62 3.13 8.05 5.12 6.06 4.57
## 13 6.40 6.35 4.31 2.43 2.29 4.27 2.64 6.27 3.31 3.95 3.11 3.78
## 14 7.09 6.42 4.64 2.30 2.21 4.72 3.24 6.68 4.21 4.02 3.37 3.35 2.55
## 15 7.88 7.37 6.27 3.34 3.79 5.63 4.20 8.13 5.89 5.20 4.90 3.19 3.57 2.61
## 16 6.36 6.27 5.20 3.21 3.03 4.88 3.61 7.00 4.59 4.60 4.02 3.21 2.47 2.38 2.01
## 17 7.93 6.88 5.44 2.60 3.03 4.77 3.60 6.94 5.16 4.94 4.67 3.41 4.19 2.70 2.95
## 18 6.87 5.97 5.25 3.48 3.66 4.04 4.26 7.03 5.13 5.13 4.87 3.51 4.06 3.10 2.40
## 19 7.96 6.98 5.99 2.74 3.62 4.56 4.26 6.83 5.59 4.41 5.38 4.47 4.51 3.18 3.01
## 20 7.01 6.49 4.69 1.98 3.08 4.05 2.89 6.12 4.46 4.03 4.34 3.26 3.28 2.33 3.11
```

After K-means clustering, we have performed hierarchical clustering. Hence, we have calculated the distance matrix for our dataset and scaled it two decimal places as you can see in the above picture.

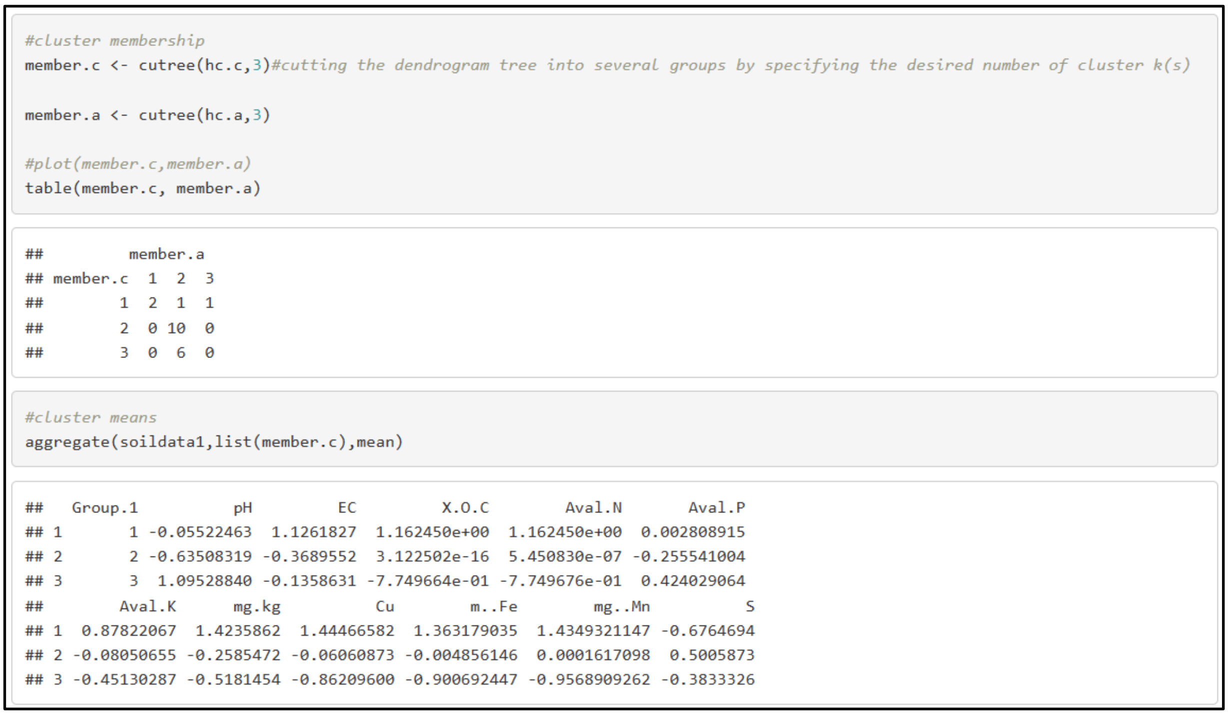
We have performed hierarchical clustering using two ways complete linkage and single linkage. The first plot is of the dendrogram using complete linkage and the second plot is of the dendrogram using average linkage.



```
#cluster dendogram
#complete
hc.c <- hclust(distance)
plot(hc.c,labels=soildata$name,lwd=2,col="blue")
```



```
hc.a <- hclust(distance,method = "average")#center-linkage
plot(hc.a,labels=soildata$name,lwd=2)
```



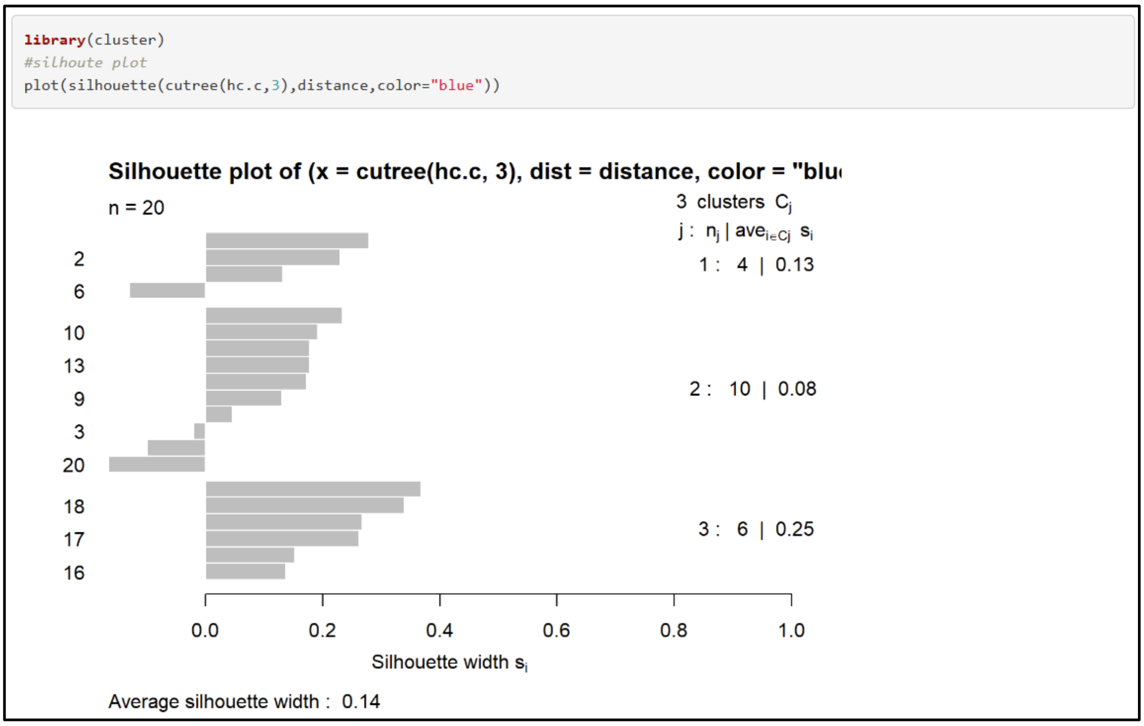
```
#cluster membership
member.c <- cutree(hc.c,3)#cutting the dendrogram tree into several groups by specifying the desired number of cluster k(s)

member.a <- cutree(hc.a,3)

#plot(member.c,member.a)
table(member.c, member.a)
```

```
##          member.a
## member.c  1  2  3
##         1  2  1  1
##         2  0 10  0
##         3  0  6  0
```

```
#cluster means
aggregate(soildata1,list(member.c),mean)
```

```
##   Group.1          pH         EC           X.O.C        Aval.N       Aval.P
## 1       1 -0.05522463  1.1261827  1.162450e+00  1.162450e+00  0.002808915
## 2       2 -0.63508319 -0.3689552  3.122502e-16  5.450830e-07 -0.255541004
## 3       3  1.09528840 -0.1358631 -7.749664e-01 -7.749676e-01  0.424029064
##        Aval.K      mg.kg          Cu         m..Fe         mg..Mn          S
## 1  0.87822067  1.4235862  1.44466582  1.363179035  1.4349321147 -0.6764694
## 2 -0.08050655 -0.2585472 -0.06060873 -0.004856146  0.0001617098  0.5005873
## 3 -0.45130287 -0.5181454 -0.86209600 -0.900692447 -0.9568909262 -0.3833326
```

To explain the dendrogram and its optimal clusters we have displayed a table that showcases that the first cluster has 2 states, the second has 17 and the last one has only 1 state.

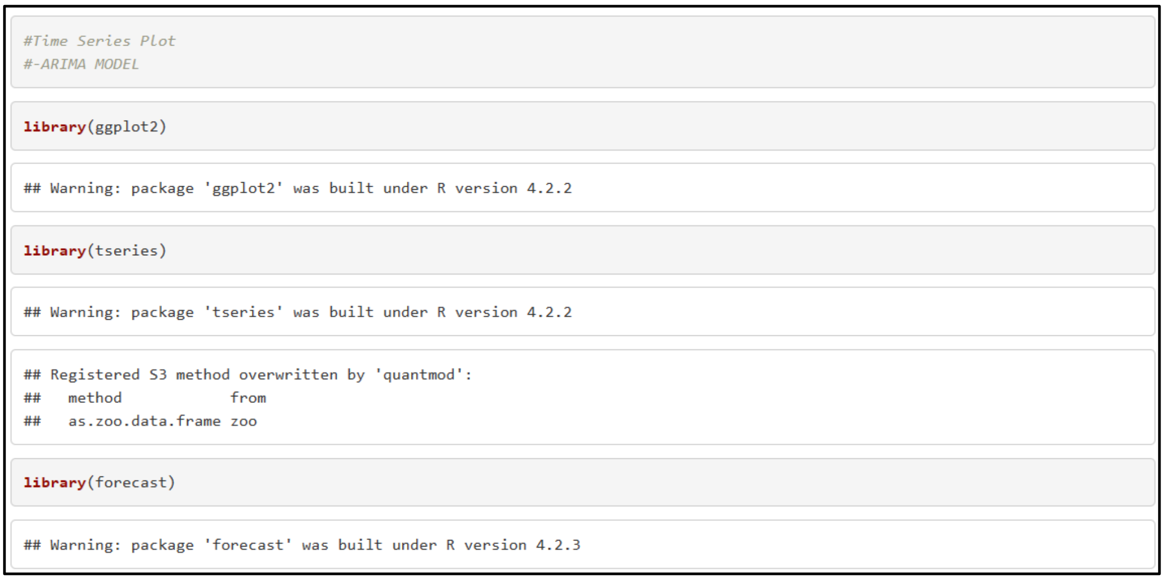


```
library(cluster)
#silhoute plot
plot(silhouette(cutree(hc.c,3),distance,color="blue"))
```

Silhouette plot of (x = cutree(hc.c, 3), dist = distance, color = "blu
n = 20
3 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 4 | 0.13
2 : 10 | 0.08
3 : 6 | 0.25
2
6
10
13
9
3
20
18
17
16
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.14

The plot above is a silhouette plot that determines the relationship between the clusters and its members. The average silhouette width is 0.14 and the average silhouette value for each of the clusters is 0.13, 0.08 and 0.25 respectively.

In the below pictures, we have loaded the important libraries necessary for the second part of our project that is Soil Moisture Prediction. We have also loaded the dataset and plotted a simple bar chart that displays the rainfall for all the states of India given.



```
#Time Series Plot
#-ARIMA MODEL
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

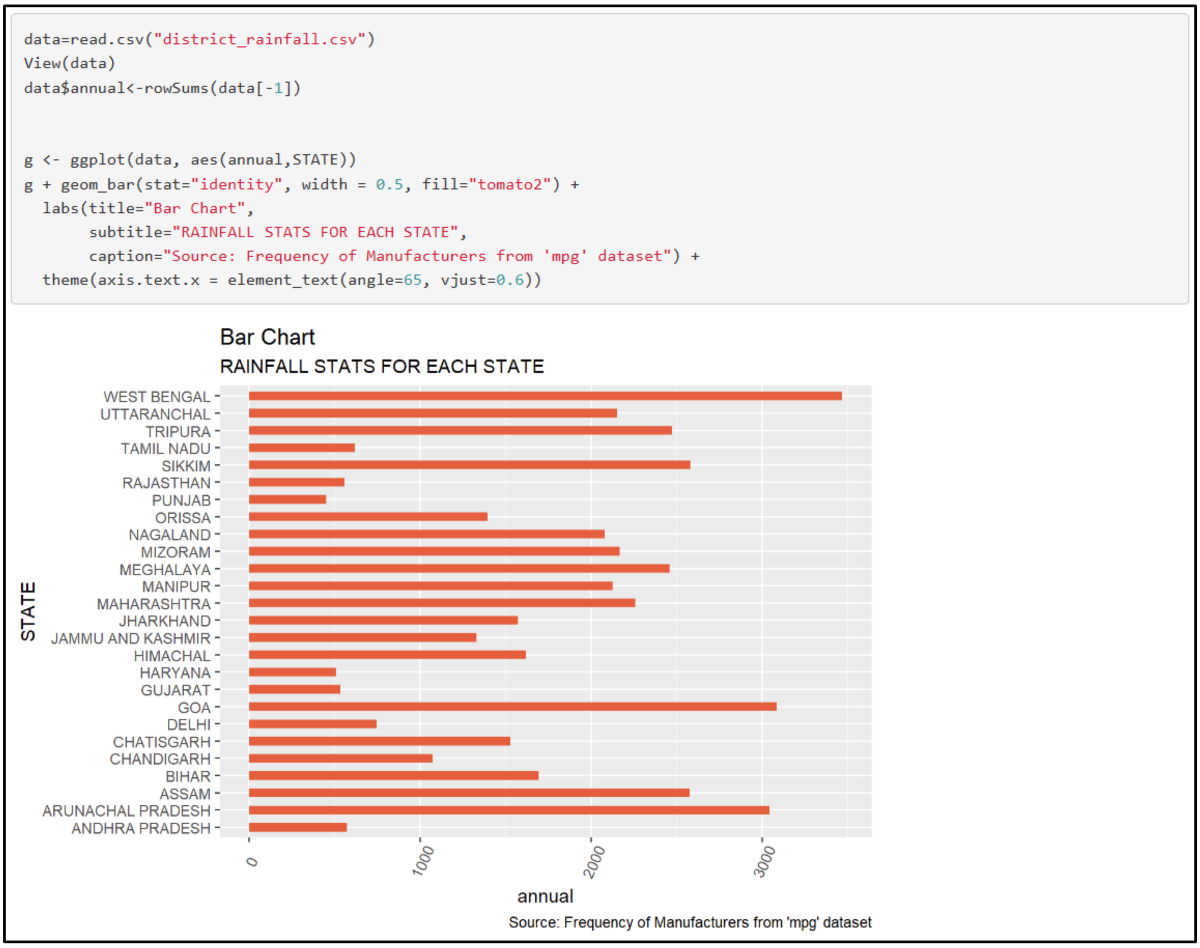
```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.2.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

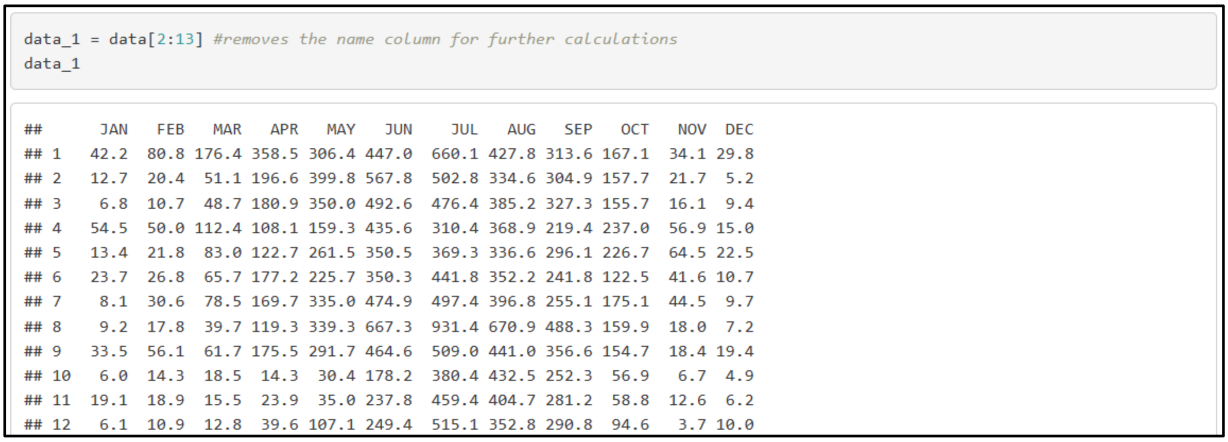
```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.2.3
```



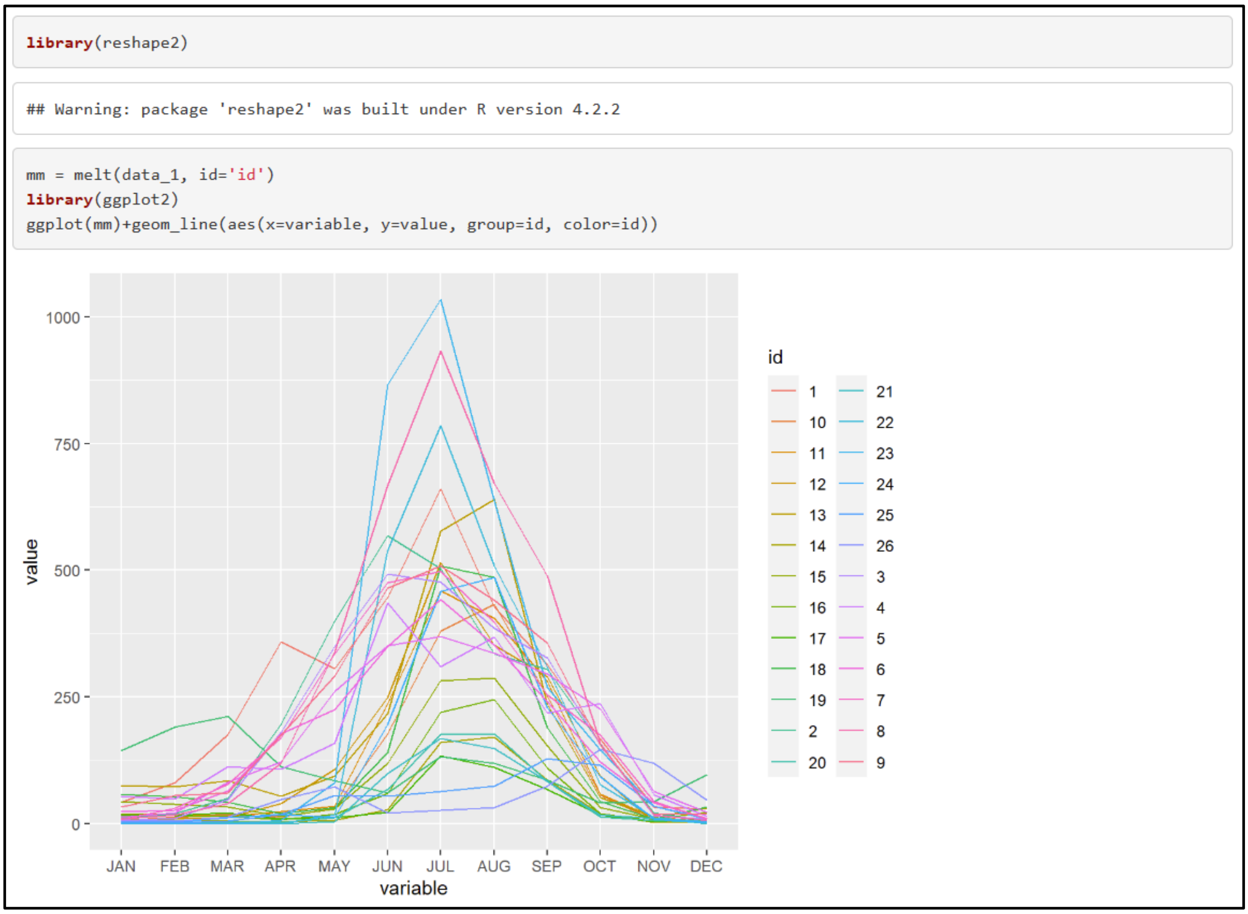
```
data=read.csv("district_rainfall.csv")
View(data)
data$annual<-rowSums(data[-1])


g <- ggplot(data, aes(annual,STATE))
g + geom_bar(stat="identity", width = 0.5, fill="tomato2") +
  labs(title="Bar Chart",
       subtitle="RAINFALL STATS FOR EACH STATE",
       caption="Source: Frequency of Manufacturers from 'mpg' dataset") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```

We have then, manipulated the data and plotted a geom plot that displays the annual rainfall of all the states. Each state is assigned a unique colour and the colours are mentioned in the legend of the plot.



```
data_1 = data[2:13] #removes the name column for further calculations
data_1
```

```
##      JAN   FEB   MAR   APR   MAY   JUN    JUL   AUG   SEP   OCT   NOV  DEC
## 1   42.2  80.8 176.4 358.5 306.4 447.0  660.1 427.8 313.6 167.1  34.1 29.8
## 2   12.7  20.4  51.1 196.6 399.8 567.8  502.8 334.6 304.9 157.7  21.7  5.2
## 3    6.8  10.7  48.7 180.9 350.0 492.6  476.4 385.2 327.3 155.7  16.1  9.4
## 4   54.5  50.0 112.4 108.1 159.3 435.6  310.4 368.9 219.4 237.0  56.9 15.0
## 5   13.4  21.8  83.0 122.7 261.5 350.5  369.3 336.6 296.1 226.7  64.5 22.5
## 6   23.7  26.8  65.7 177.2 225.7 350.3  441.8 352.2 241.8 122.5  41.6 10.7
## 7    8.1  30.6  78.5 169.7 335.0 474.9  497.4 396.8 255.1 175.1  44.5  9.7
## 8    9.2  17.8  39.7 119.3 339.3 667.3  931.4 670.9 488.3 159.9  18.0  7.2
## 9   33.5  56.1  61.7 175.5 291.7 464.6  509.0 441.0 356.6 154.7  18.4 19.4
## 10   6.0  14.3  18.5  14.3  30.4 178.2  380.4 432.5 252.3  56.9   6.7  4.9
## 11  19.1  18.9  15.5  23.9  35.0 237.8  459.4 404.7 281.2  58.8  12.6  6.2
## 12   6.1  10.9  12.8  39.6 107.1 249.4  515.1 352.8 290.8  94.6   3.7 10.0
```



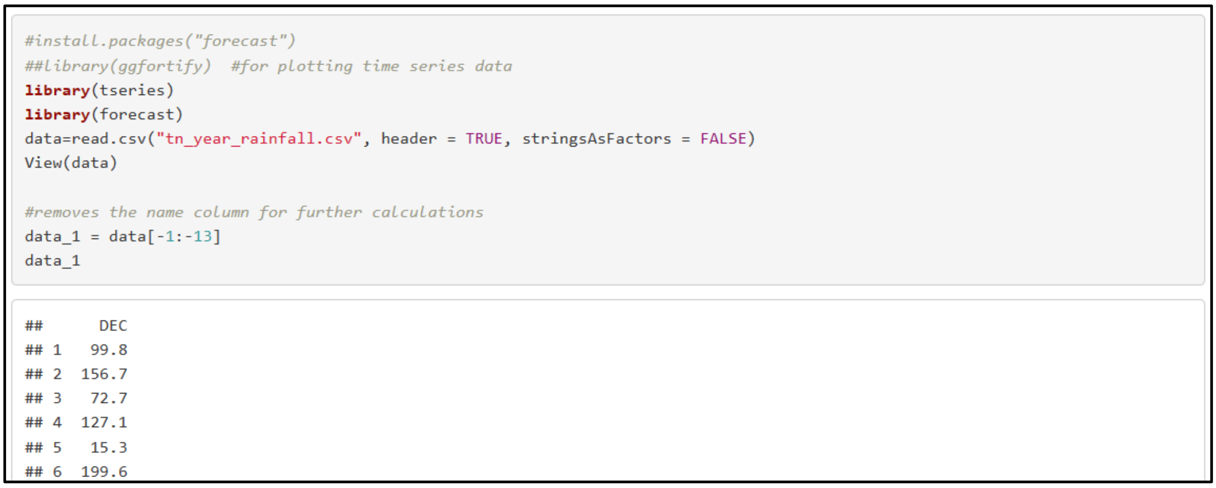
```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.2.2
```

```
mm = melt(data_1, id='id')
library(ggplot2)
ggplot(mm)+geom_line(aes(x=variable, y=value, group=id, color=id))
```

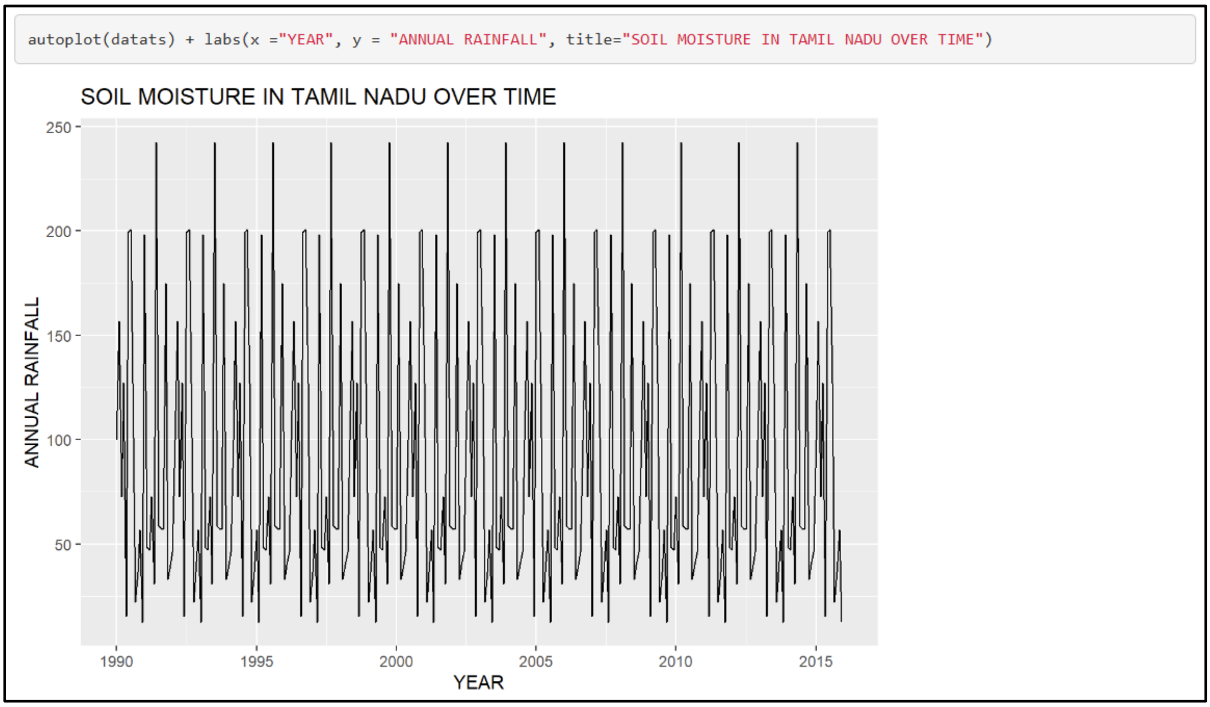
To showcase the soil moisture prediction, we have selected the state Tamil Nadu and taken a new dataset that consists of the rainfall values of Tamil Nadu over the year from 1990 to 2015. After loading this dataset and the required libraries we have plotted an auto plot that depicts the soil moisture of Tamil Nadu over the years.

From the plot you can notice that, the maximum moisture value is nearly 250 and the minimum value is nearly 10.



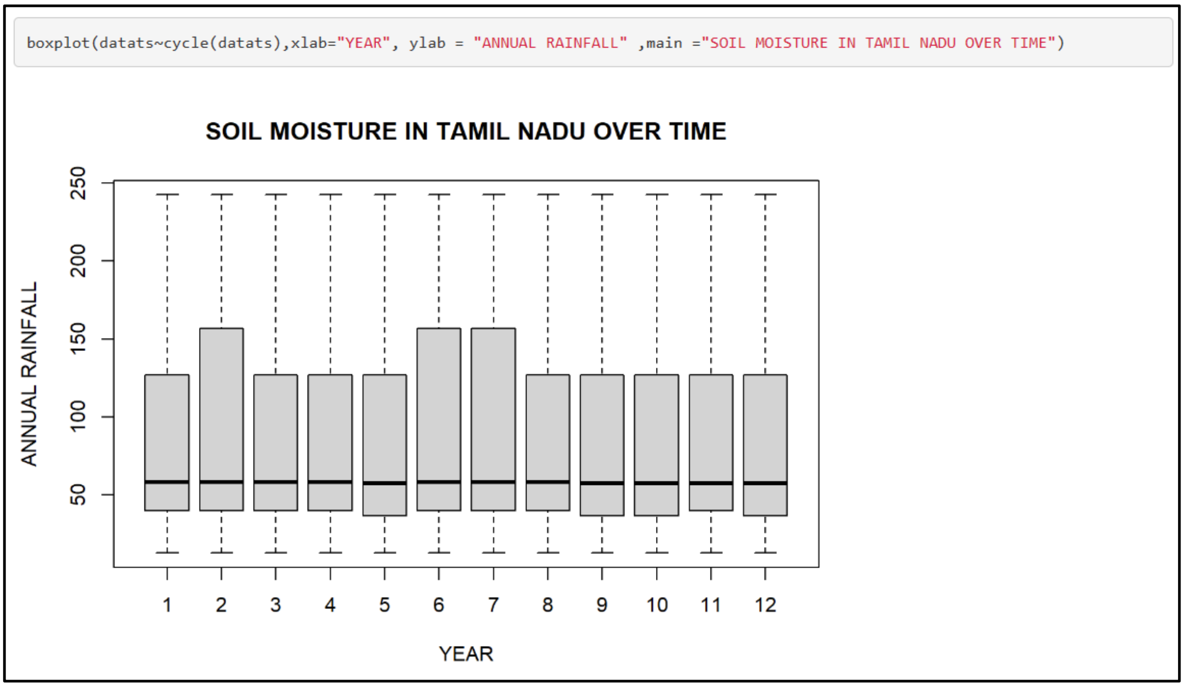
```
#install.packages("forecast")
##library(ggfortify)  #for plotting time series data
library(tseries)
library(forecast)
data=read.csv("tn_year_rainfall.csv", header = TRUE, stringsAsFactors = FALSE)
View(data)

#removes the name column for further calculations
data_1 = data[-1:-13]
data_1
```

```
##      DEC
## 1   99.8
## 2  156.7
## 3   72.7
## 4  127.1
## 5   15.3
## 6  199.6
```

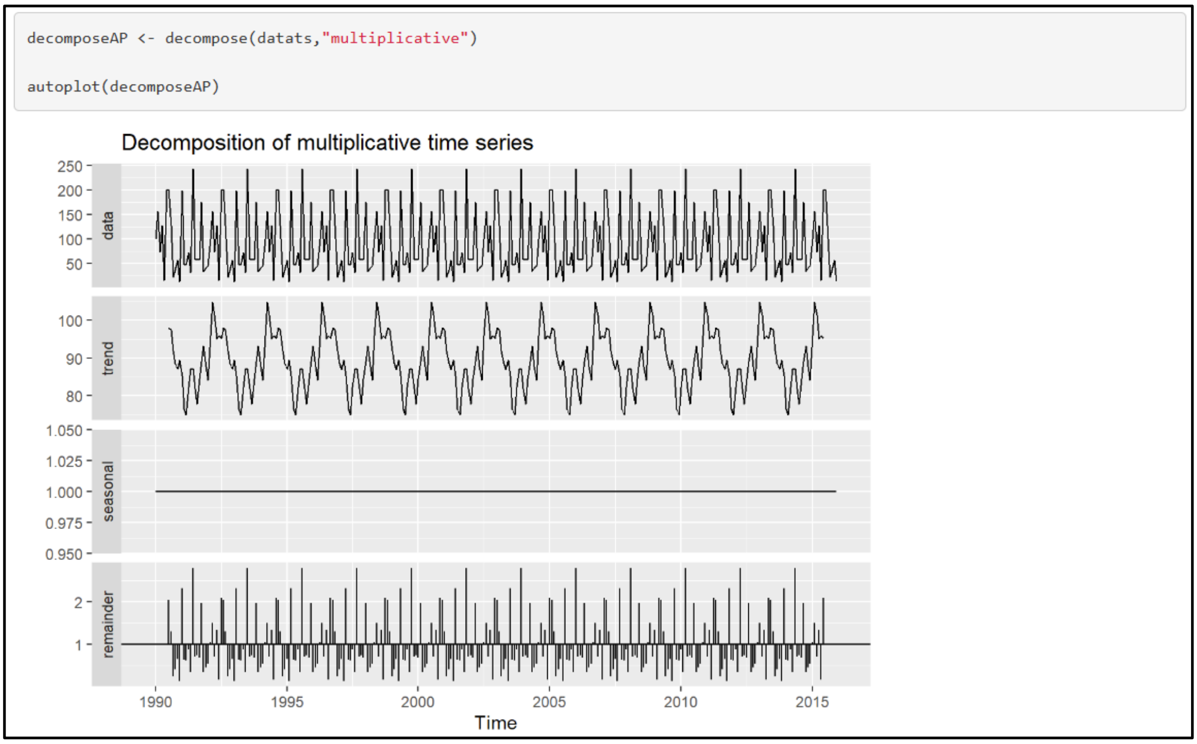


```
autoplot(datats) + labs(x ="YEAR", y = "ANNUAL RAINFALL", title="SOIL MOISTURE IN TAMIL NADU OVER TIME")
```

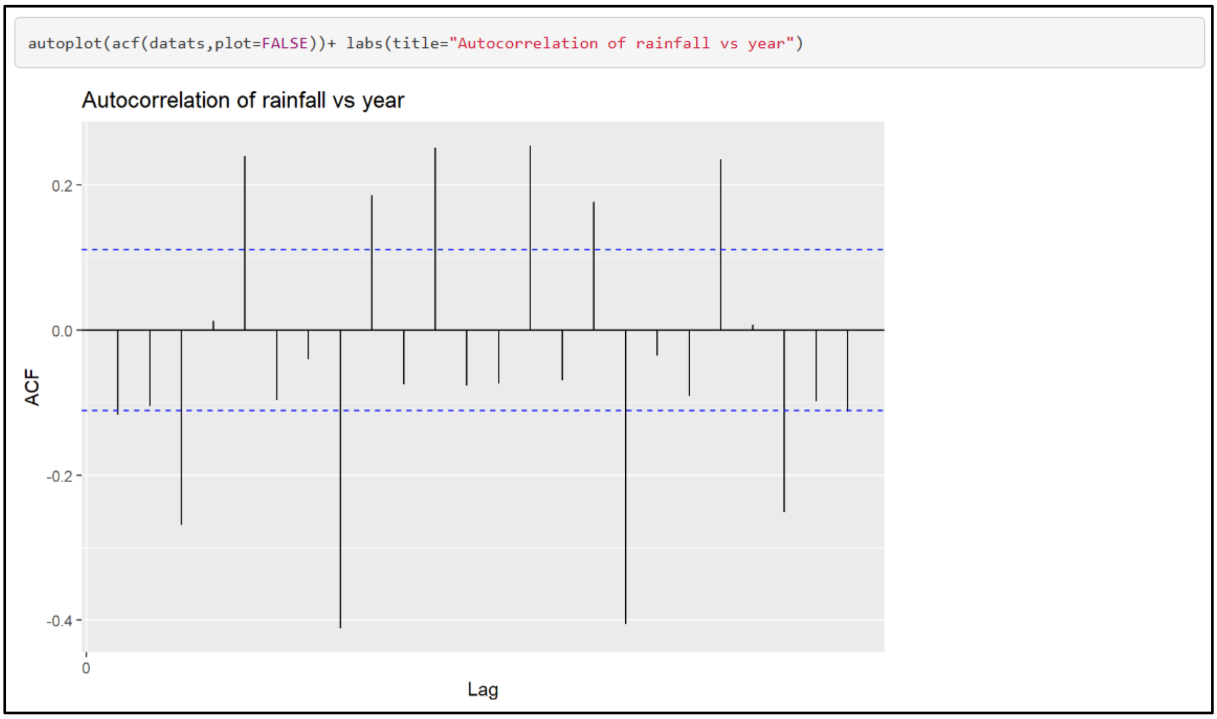
After plotting the boxplot for the same, we have displayed a multiplicative time series plot that shows the pattern of the rainfall in Tamil Nadu. The plot consists of four sections data, trend, seasonal and remainder.



```
boxplot(datats~cycle(datats),xlab="YEAR", ylab = "ANNUAL RAINFALL" ,main ="SOIL MOISTURE IN TAMIL NADU OVER TIME")
```

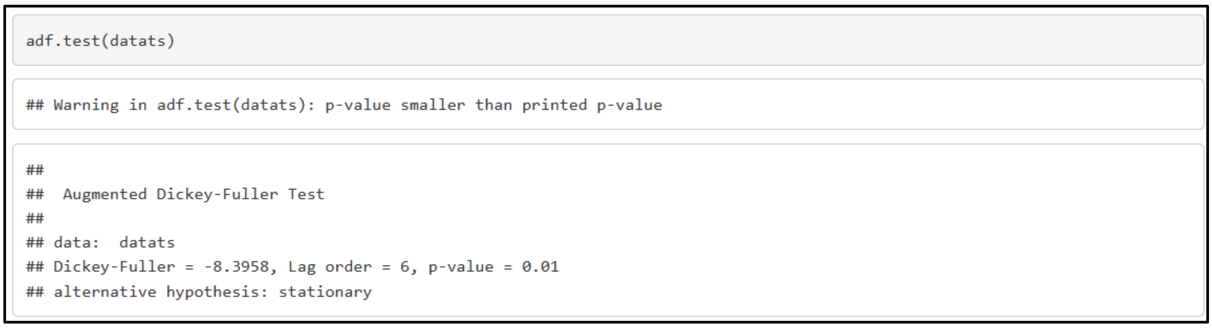


```
decomposeAP <- decompose(datats,"multiplicative")

autoplot(decomposeAP)
```



```
autoplot(acf(datats,plot=FALSE))+ labs(title="Autocorrelation of rainfall vs year")
```

The above picture displays the autocorrelation plot for our dataset. As you can see from the plot, the peak values are nearly above 0.25.

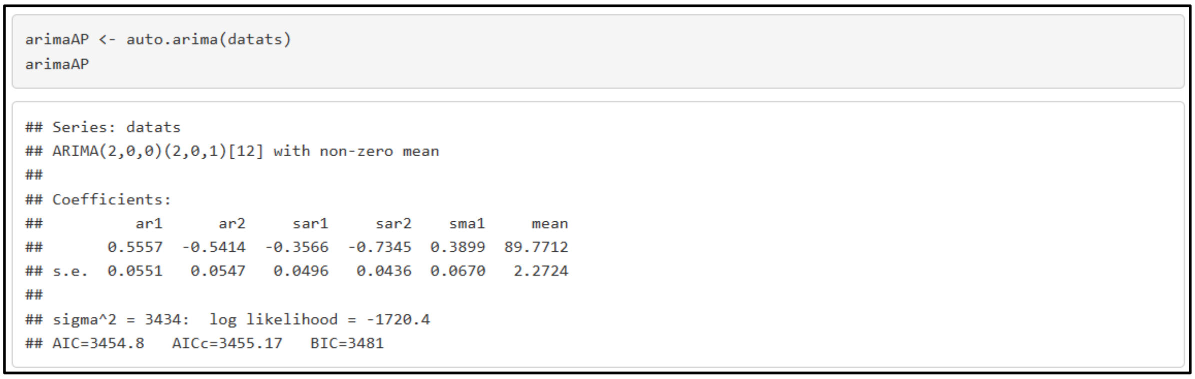


```
adf.test(datats)
```

```
## Warning in adf.test(datats): p-value smaller than printed p-value
```

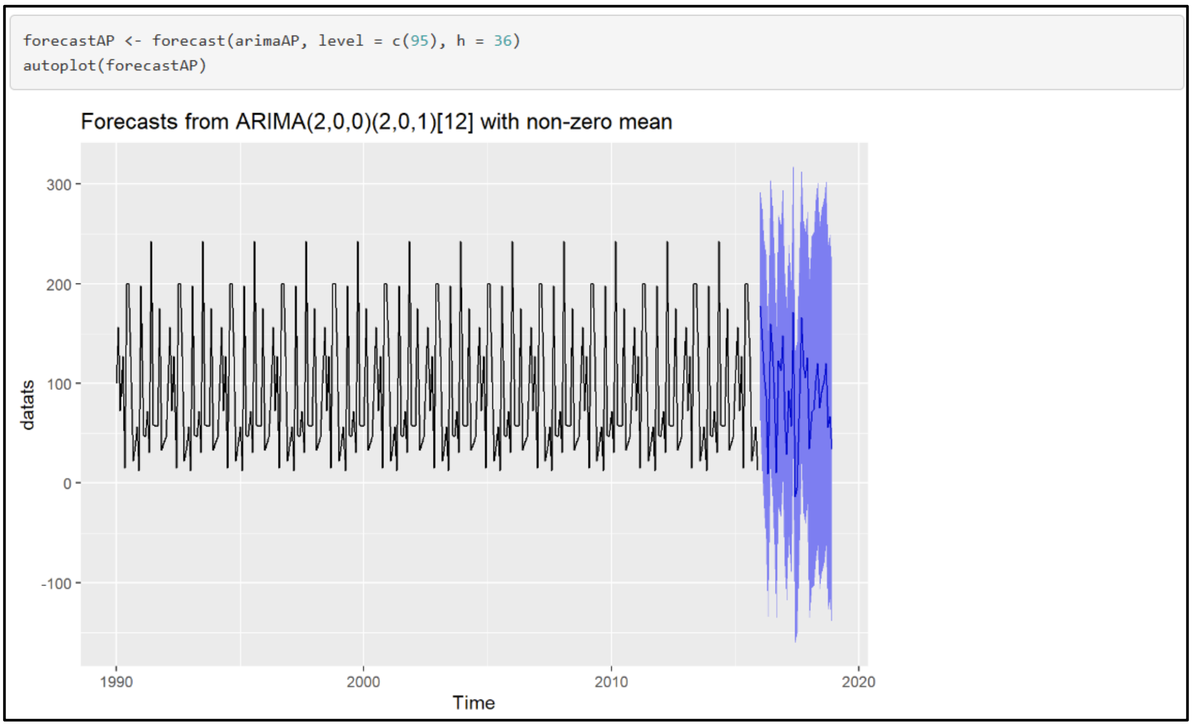
```
##
##  Augmented Dickey-Fuller Test
##
## data:  datats
## Dickey-Fuller = -8.3958, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

We have applied the ARIMA model on our given dataset and displayed its results. The values of the coefficients are displayed. The standard error values are also displayed which are 0.051,0.0547,0.0496, 0.0670 and 2.274 respectively. The log likelihood value is -1720.4



```
arimaAP <- auto.arima(datats)
arimaAP
```

```
## Series: datats
## ARIMA(2,0,0)(2,0,1)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2     sar1     sar2    sma1     mean
##       0.5557  -0.5414  -0.3566  -0.7345  0.3899  89.7712
## s.e.  0.0551   0.0547   0.0496   0.0436  0.0670   2.2724
##
## sigma^2 = 3434:  log likelihood = -1720.4
## AIC=3454.8   AICc=3455.17   BIC=3481
```

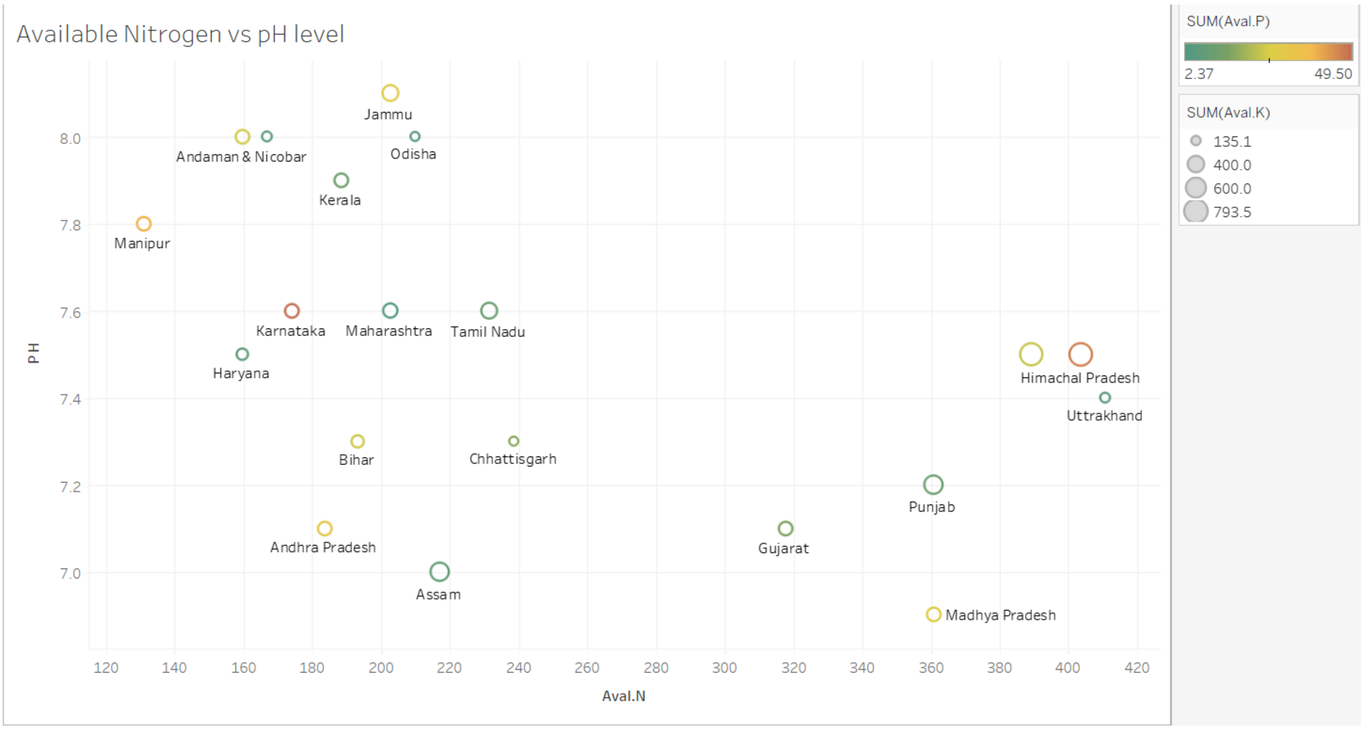


```
forecastAP <- forecast(arimaAP, level = c(95), h = 36)
autoplot(forecastAP)
```

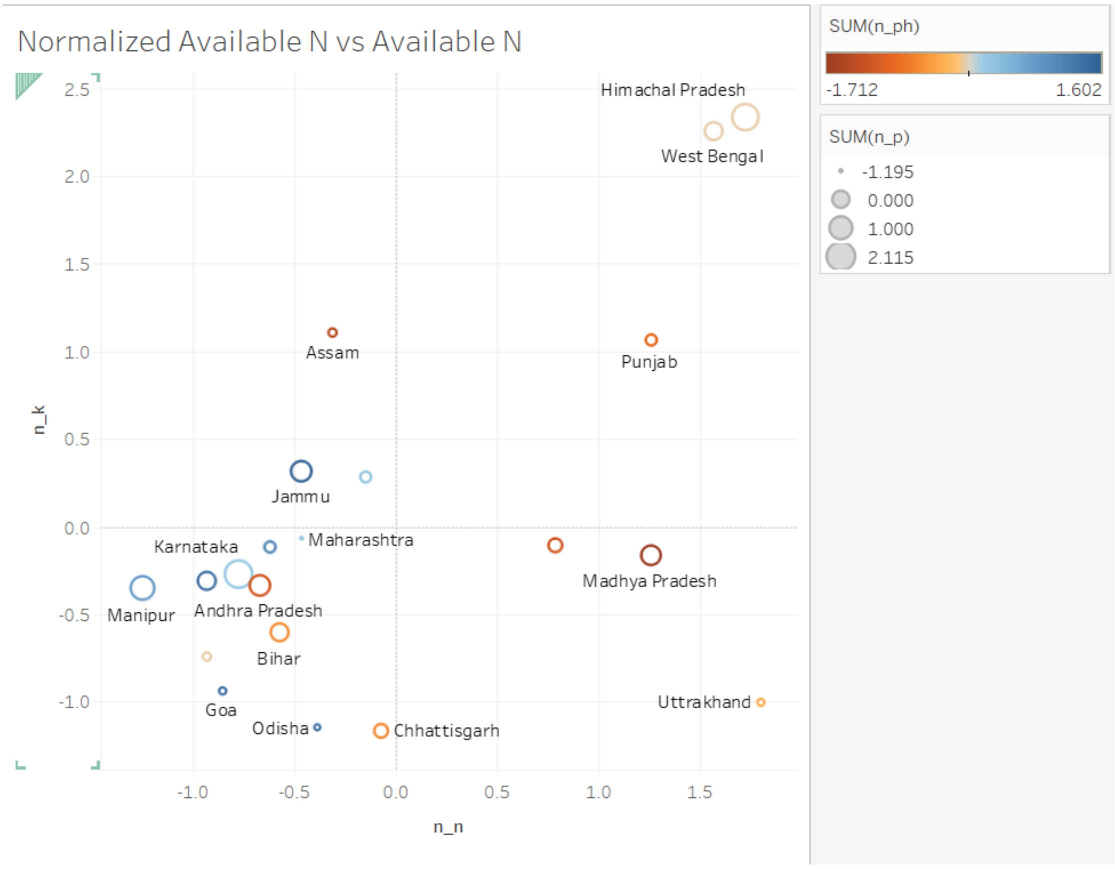
Lastly, using the ARIMA model we have displayed an auto plot for the forecasts of the soil moisture in Tamil Nadu. In our dataset, the values are given till 2015 but we have forecasted the values for the next 5 years which is displayed in blue colour.

- # Tableau outputs

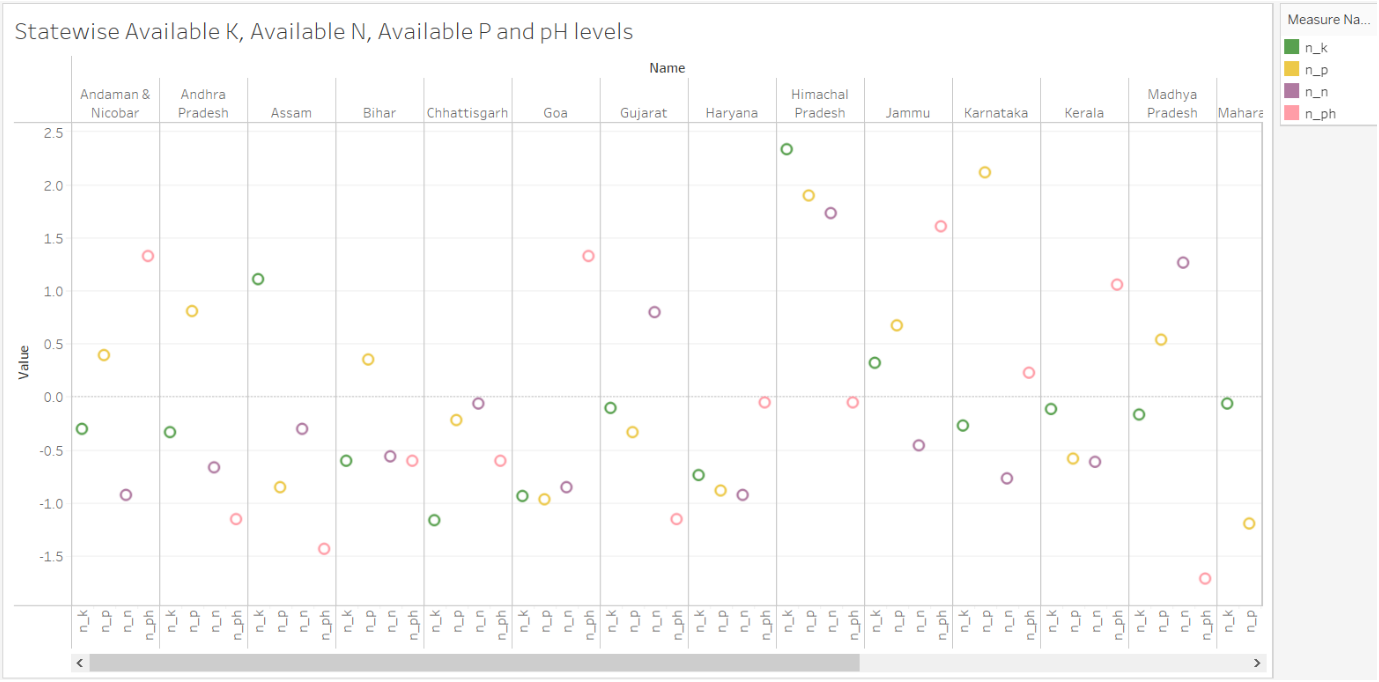
1. Available N vs pH levels (Size – Available K)



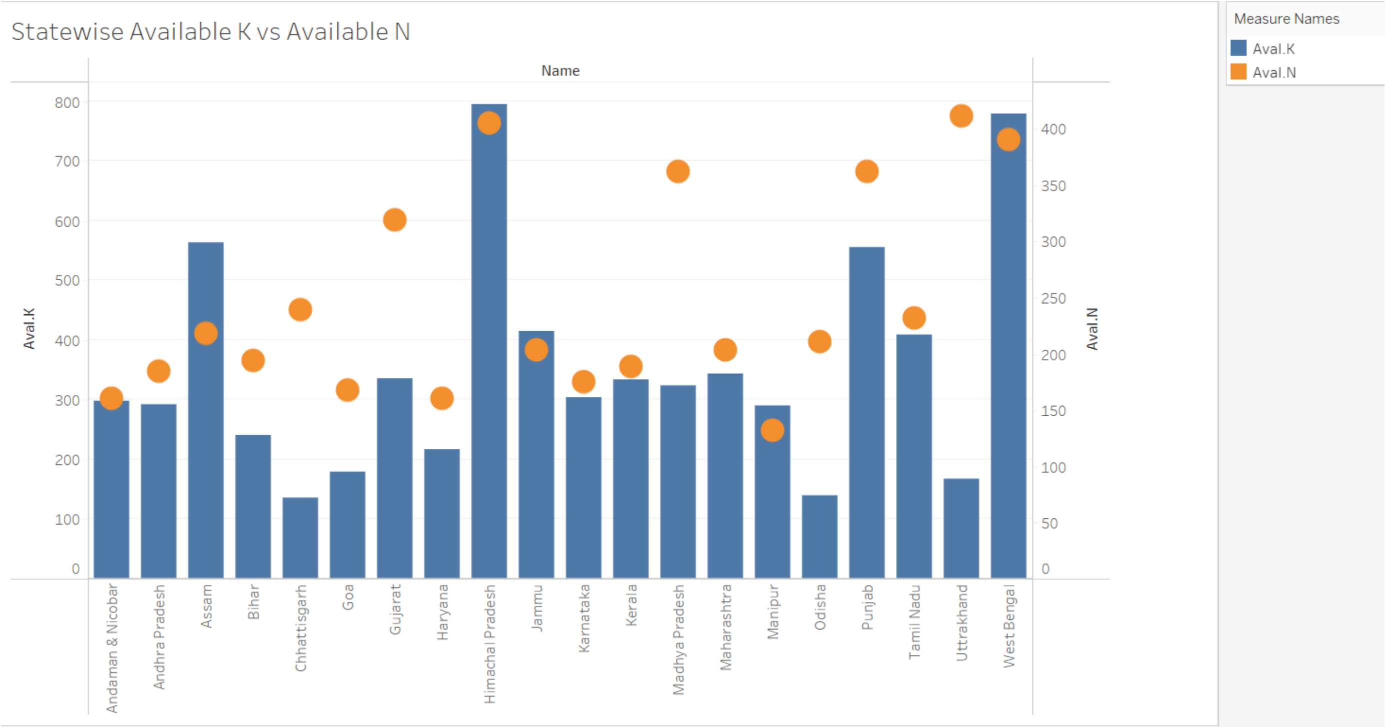
2. Normalized N vs normalized K (Size – normalized P, Color – normalized pH levels)



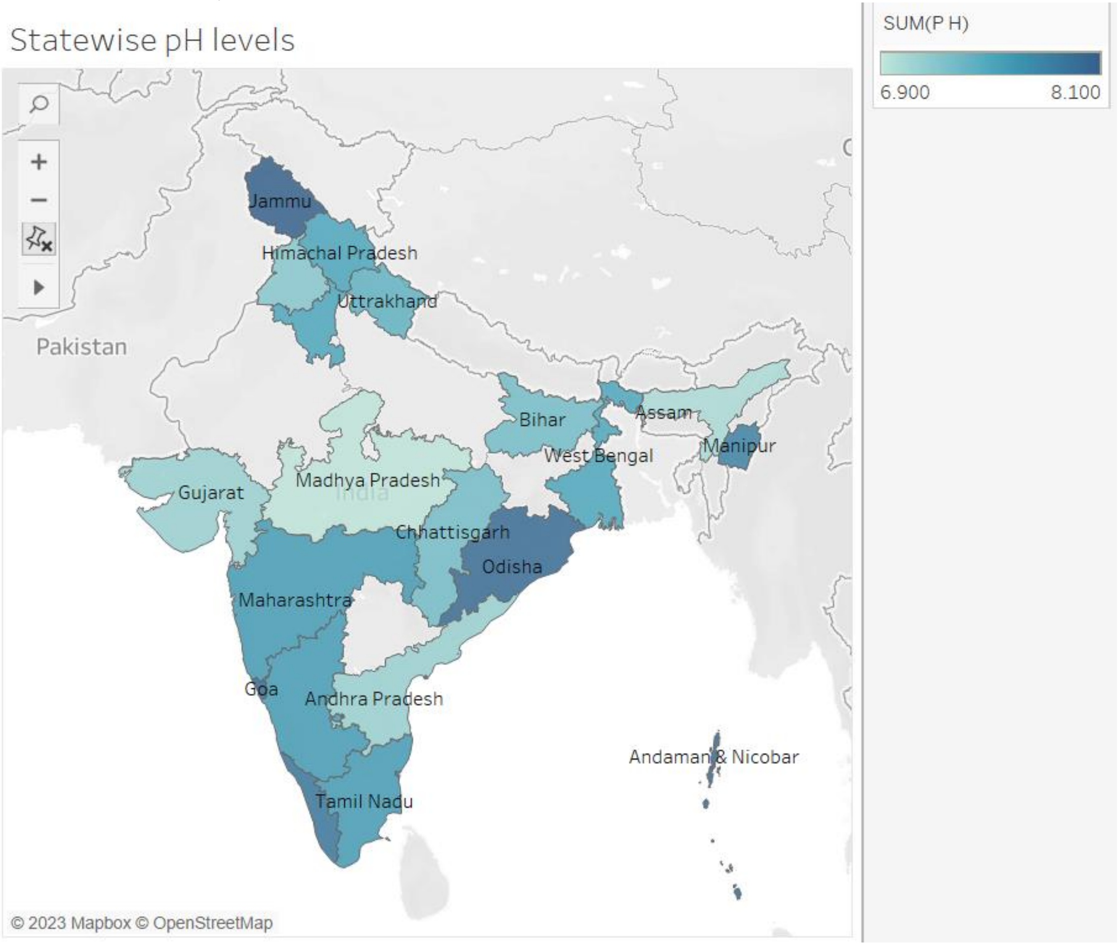
3. Normalized K vs P vs N vs pH statewise



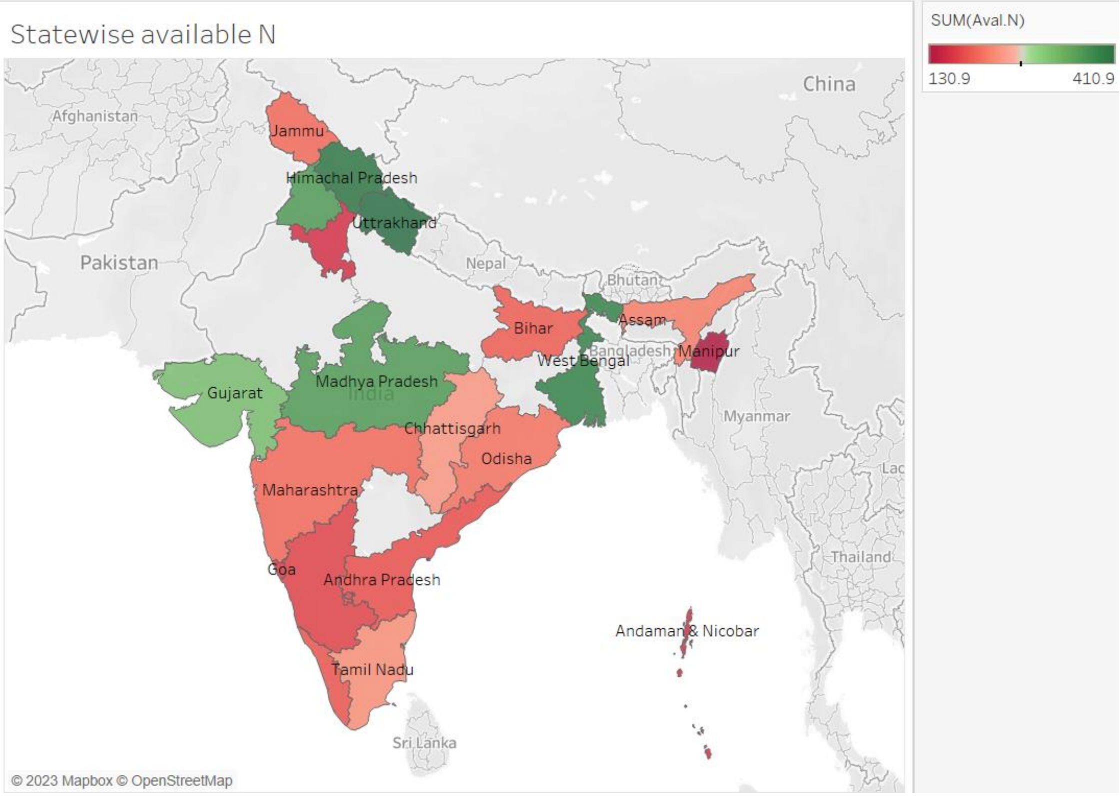
4. Statewise available K vs available N



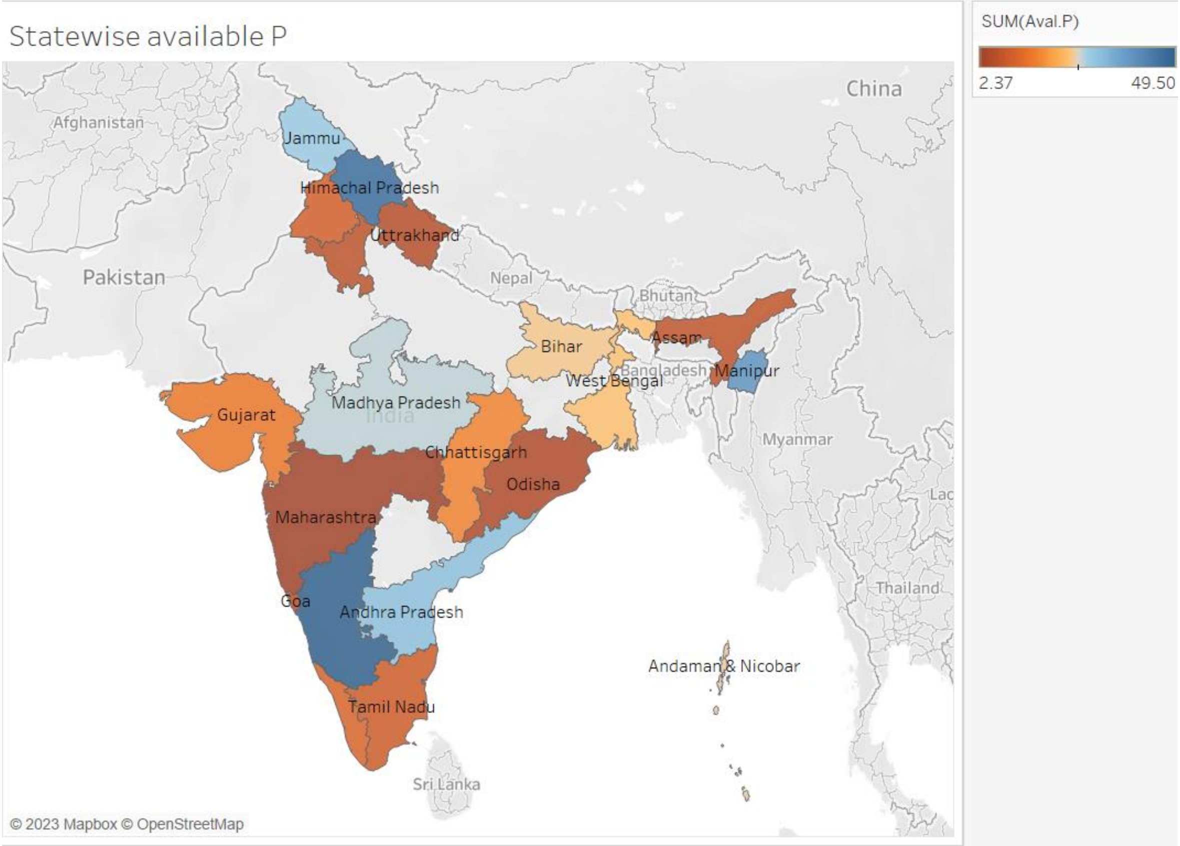
5. Statewise pH levels



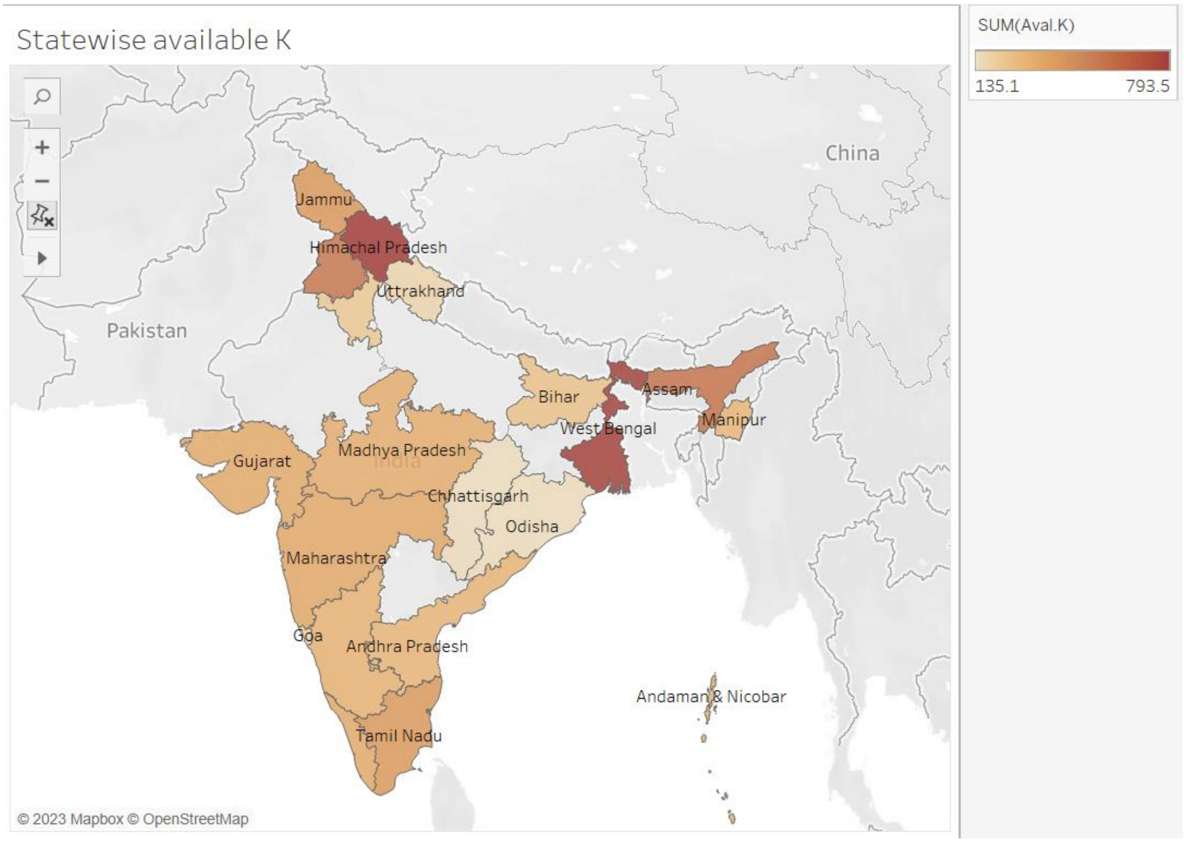
6. Statewise available N



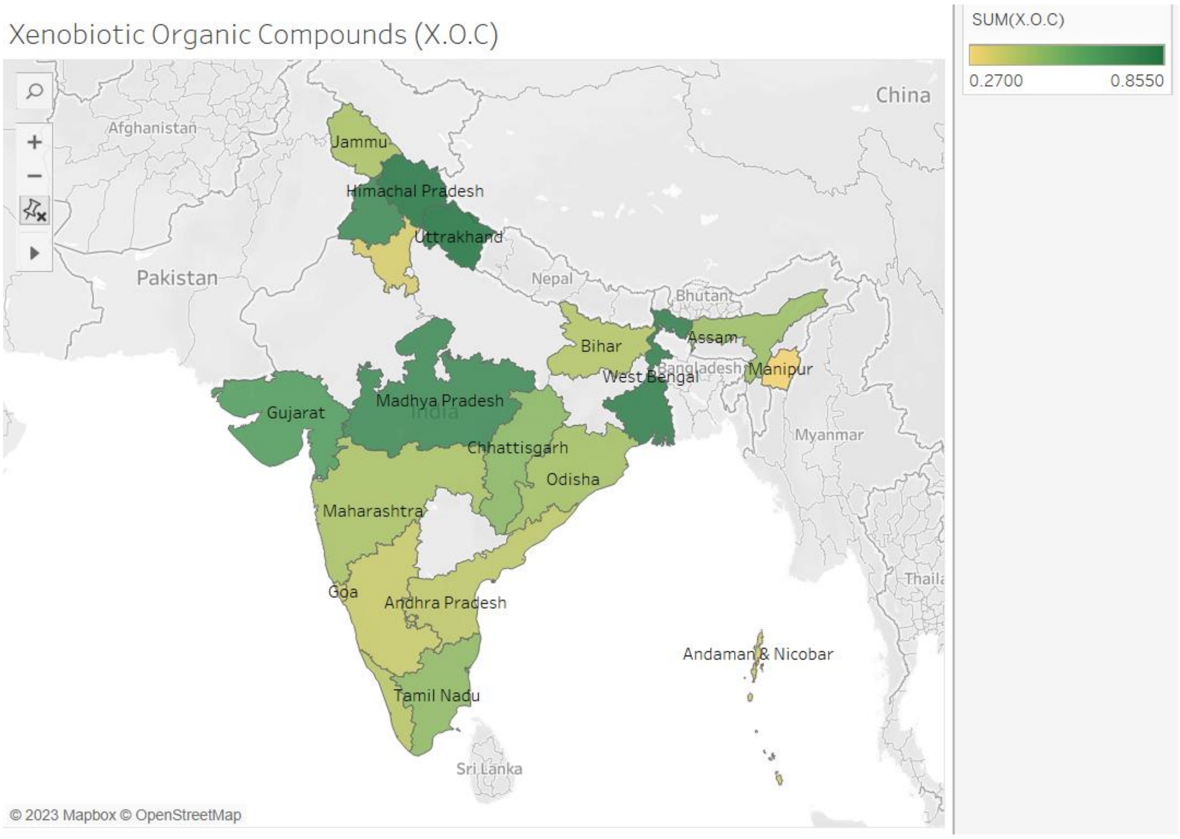
7. Statewise available P



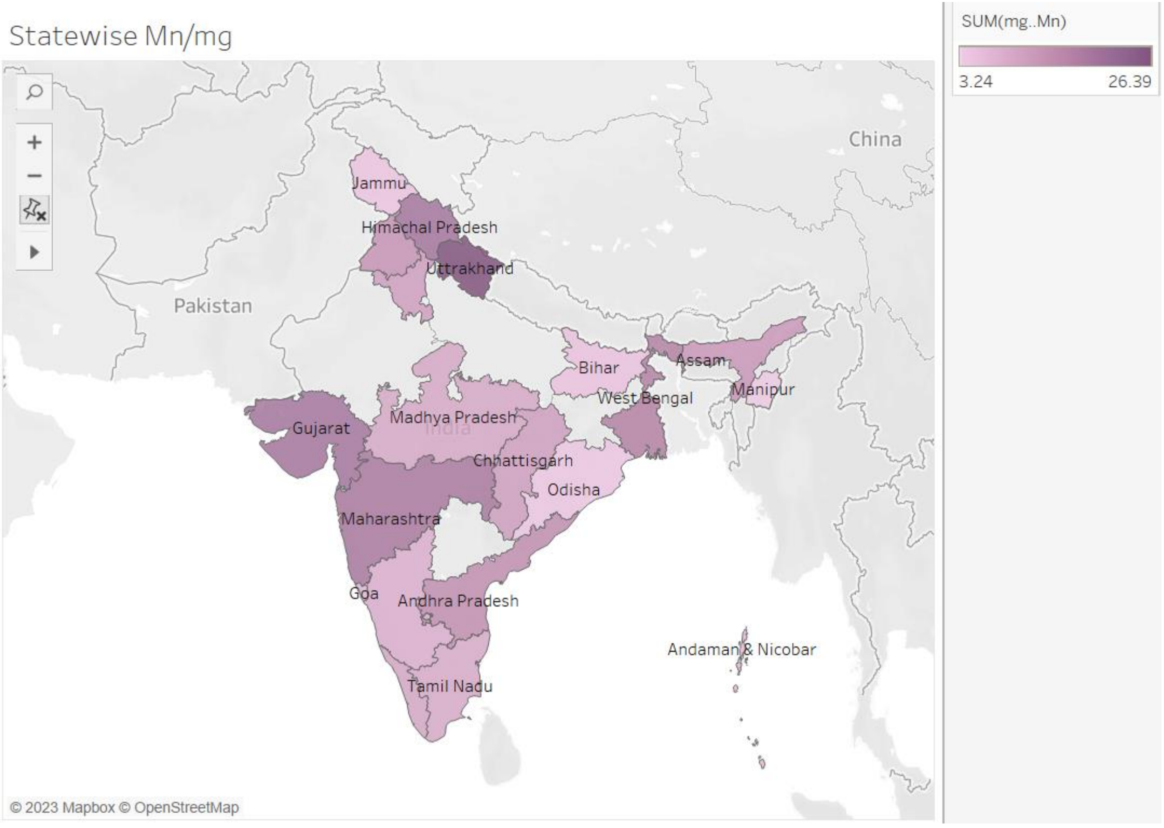
8. Statewise available K



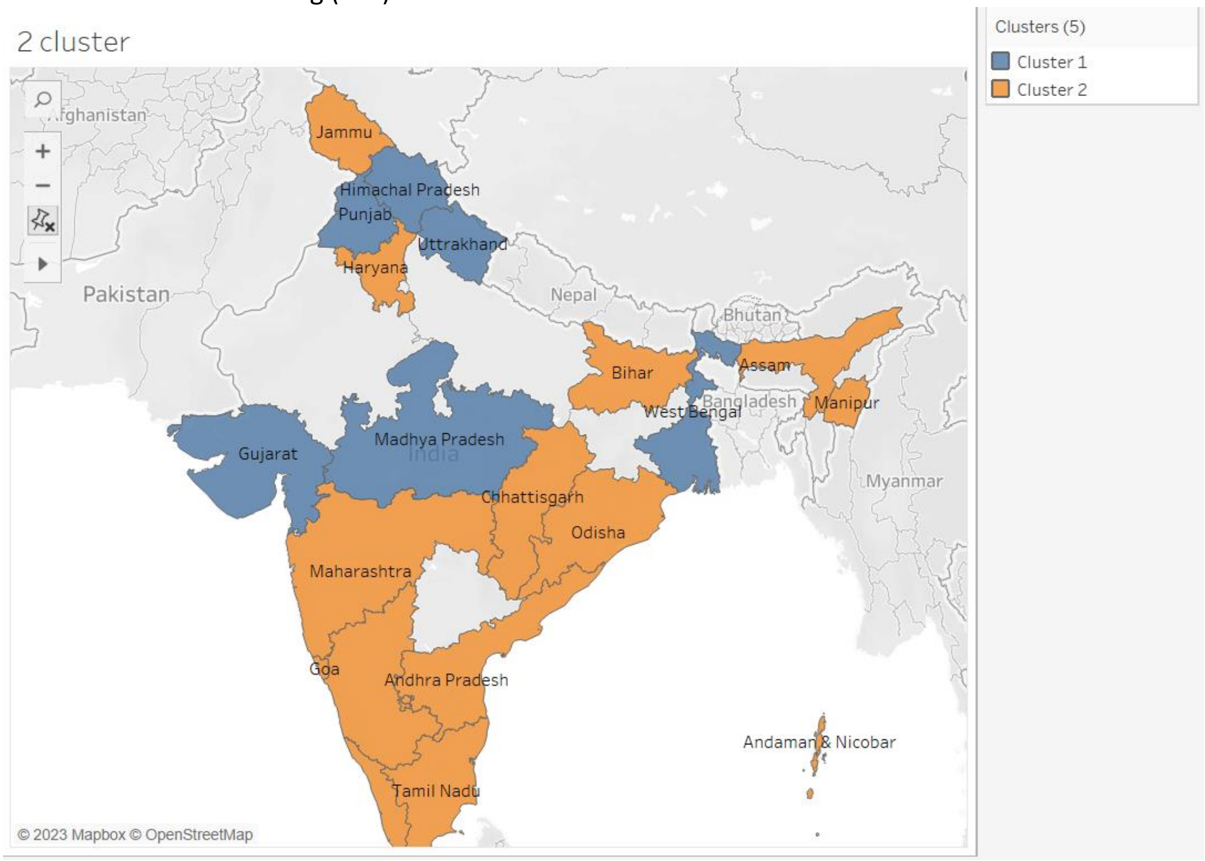
9. Statewise X.O.C.



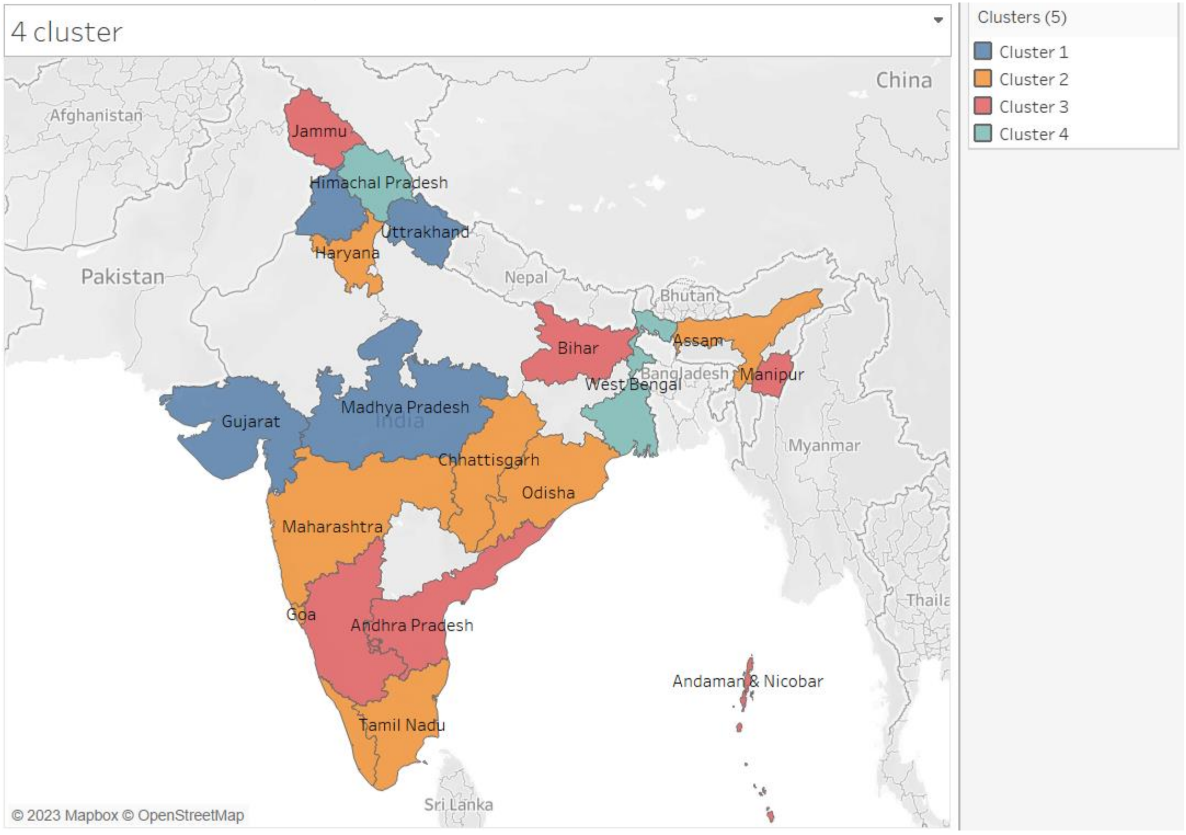
10. Statewise Mn/mg



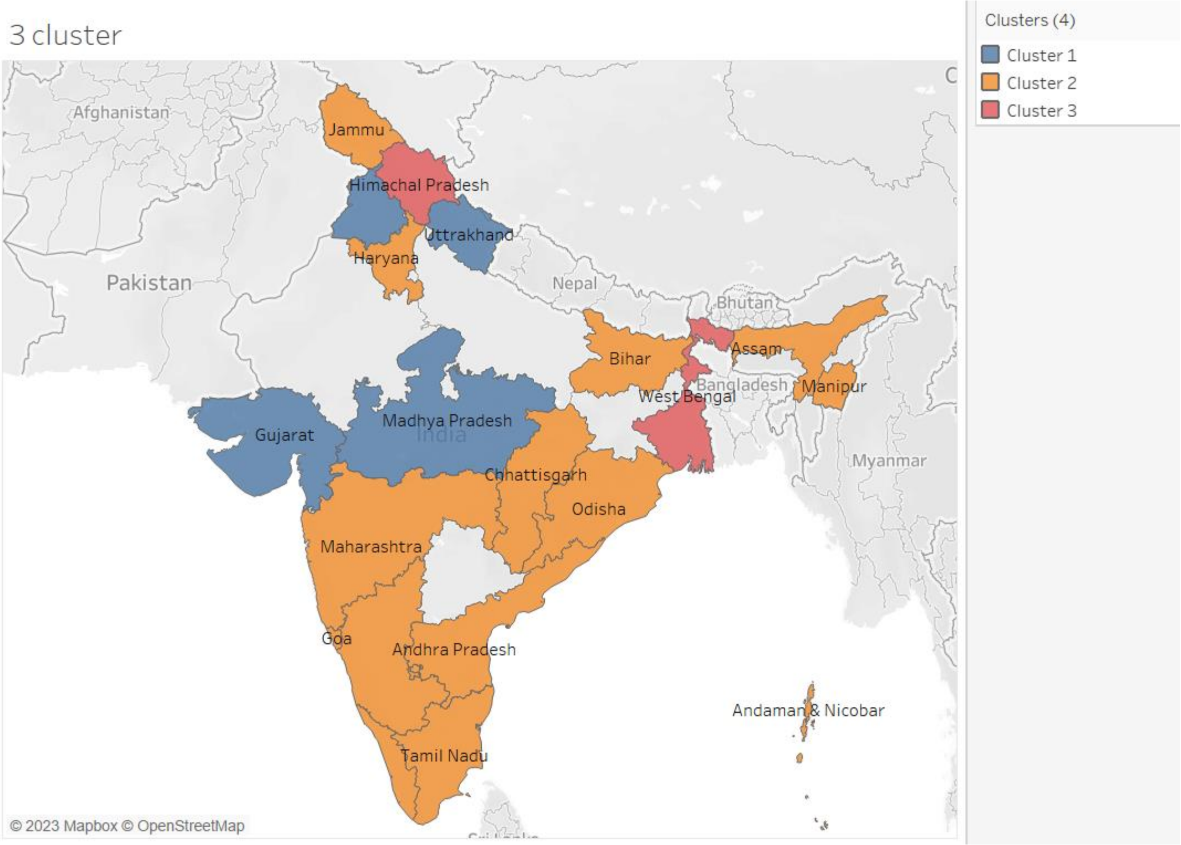
11. K means clustering (k=2)



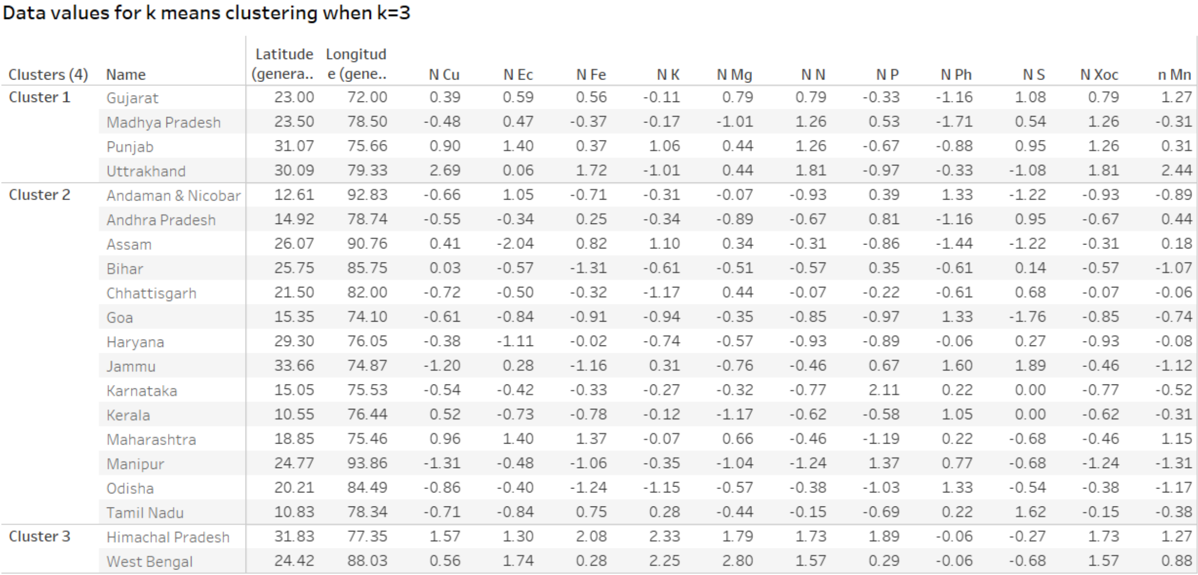
12. K means clustering (k=4)



13. K means clustering (k=3)



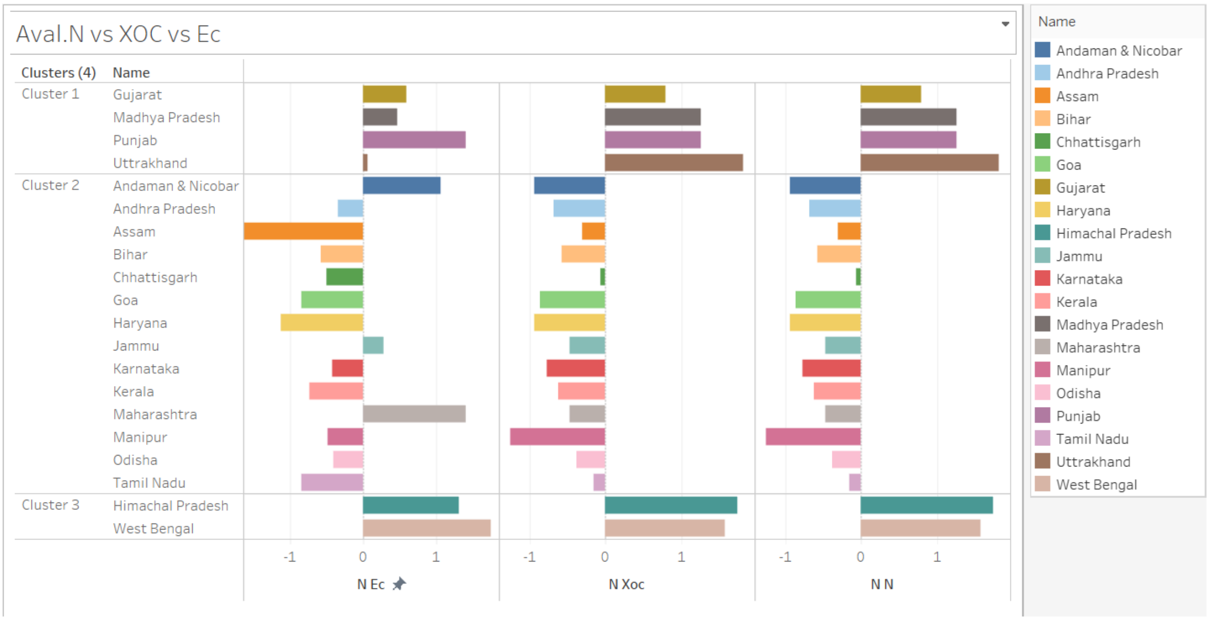
14. Data values for k means clustering when k=3



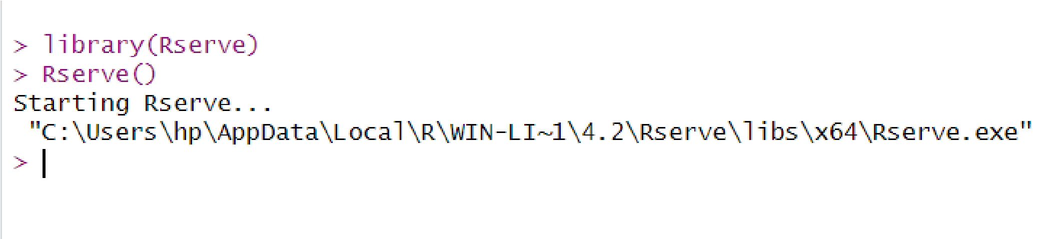
Data values for k means clustering when k=3

Clusters (4)	Name	Latitude (genera..	Longitud e (gene..	N Cu	N Ec	N Fe	N K	N Mg	N N	N P	N Ph	N S	N Xoc	n Mn
Cluster 1	Gujarat	23.00	72.00	0.39	0.59	0.56	-0.11	0.79	0.79	-0.33	-1.16	1.08	0.79	1.27
	Madhya Pradesh	23.50	78.50	-0.48	0.47	-0.37	-0.17	-1.01	1.26	0.53	-1.71	0.54	1.26	-0.31
	Punjab	31.07	75.66	0.90	1.40	0.37	1.06	0.44	1.26	-0.67	-0.88	0.95	1.26	0.31
	Uttrakhand	30.09	79.33	2.69	0.06	1.72	-1.01	0.44	1.81	-0.97	-0.33	-1.08	1.81	2.44
Cluster 2	Andaman & Nicobar	12.61	92.83	-0.66	1.05	-0.71	-0.31	-0.07	-0.93	0.39	1.33	-1.22	-0.93	-0.89
	Andhra Pradesh	14.92	78.74	-0.55	-0.34	0.25	-0.34	-0.89	-0.67	0.81	-1.16	0.95	-0.67	0.44
	Assam	26.07	90.76	0.41	-2.04	0.82	1.10	0.34	-0.31	-0.86	-1.44	-1.22	-0.31	0.18
	Bihar	25.75	85.75	0.03	-0.57	-1.31	-0.61	-0.51	-0.57	0.35	-0.61	0.14	-0.57	-1.07
	Chhattisgarh	21.50	82.00	-0.72	-0.50	-0.32	-1.17	0.44	-0.07	-0.22	-0.61	0.68	-0.07	-0.06
	Goa	15.35	74.10	-0.61	-0.84	-0.91	-0.94	-0.35	-0.85	-0.97	1.33	-1.76	-0.85	-0.74
	Haryana	29.30	76.05	-0.38	-1.11	-0.02	-0.74	-0.57	-0.93	-0.89	-0.06	0.27	-0.93	-0.08
	Jammu	33.66	74.87	-1.20	0.28	-1.16	0.31	-0.76	-0.46	0.67	1.60	1.89	-0.46	-1.12
	Karnataka	15.05	75.53	-0.54	-0.42	-0.33	-0.27	-0.32	-0.77	2.11	0.22	0.00	-0.77	-0.52
	Kerala	10.55	76.44	0.52	-0.73	-0.78	-0.12	-1.17	-0.62	-0.58	1.05	0.00	-0.62	-0.31
	Maharashtra	18.85	75.46	0.96	1.40	1.37	-0.07	0.66	-0.46	-1.19	0.22	-0.68	-0.46	1.15
	Manipur	24.77	93.86	-1.31	-0.48	-1.06	-0.35	-1.04	-1.24	1.37	0.77	-0.68	-1.24	-1.31
	Odisha	20.21	84.49	-0.86	-0.40	-1.24	-1.15	-0.57	-0.38	-1.03	1.33	-0.54	-0.38	-1.17
	Tamil Nadu	10.83	78.34	-0.71	-0.84	0.75	0.28	-0.44	-0.15	-0.69	0.22	1.62	-0.15	-0.38
Cluster 3	Himachal Pradesh	31.83	77.35	1.57	1.30	2.08	2.33	1.79	1.73	1.89	-0.06	-0.27	1.73	1.27
	West Bengal	24.42	88.03	0.56	1.74	0.28	2.25	2.80	1.57	0.29	-0.06	-0.68	1.57	0.88

15. Clusterwise nutrient analysis

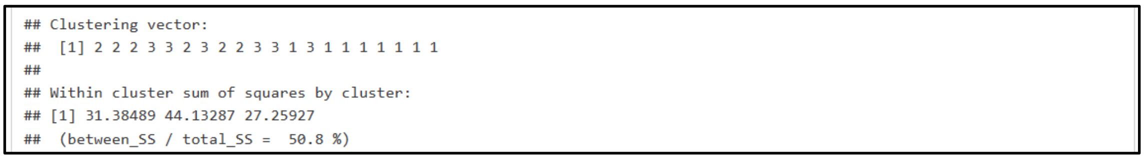


16. Rserve connection in R



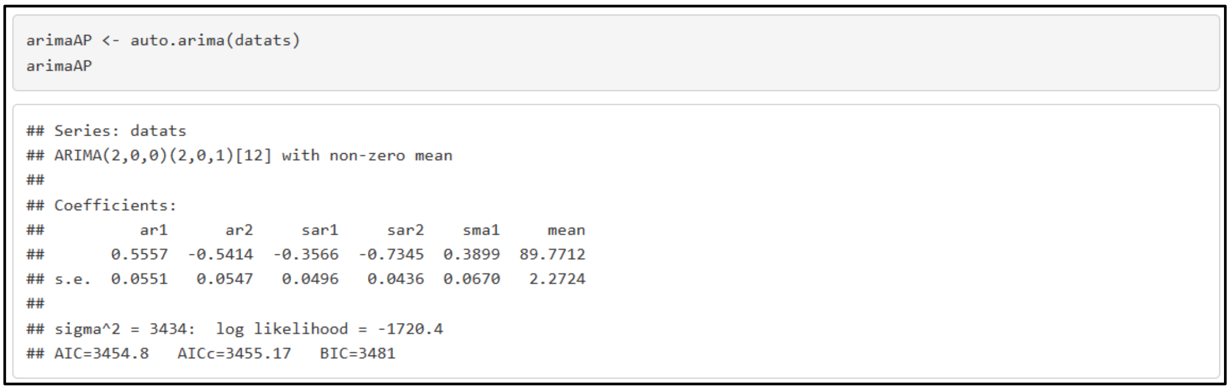
```
> library(Rserve)
> Rserve()
Starting Rserve...
 "C:\Users\hp\AppData\Local\R\WIN-LI~1\4.2\Rserve\libs\x64\Rserve.exe"
> |
```

TESTING AND PERFORMANCE EVALUATION



```
## Clustering vector:
##  [1] 2 2 2 3 3 2 3 2 2 3 3 1 3 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 31.38489 44.13287 27.25927
##  (between_SS / total_SS =  50.8 %)
```

The optimal number of clusters for K-means clustering obtained from the elbow method is 3. The value of WCSS is when the data is clustered into 3 cluster is 31.384+44.132+27.259= 102.775. The value of WCSS must be minimal to obtained the optimal number of clusters. As the optimal number of clusters obtained from the elbow method is 3, the value of WCSS will be minimum for k=3.



```
arimaAP <- auto.arima(datats)
arimaAP
```

```
## Series: datats
## ARIMA(2,0,0)(2,0,1)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2     sar1     sar2    sma1     mean
##       0.5557  -0.5414  -0.3566  -0.7345  0.3899  89.7712
## s.e.  0.0551   0.0547   0.0496   0.0436  0.0670   2.2724
##
## sigma^2 = 3434:  log likelihood = -1720.4
## AIC=3454.8   AICc=3455.17   BIC=3481
```

When we implement ARIMA method on a dataset, AIC criterion allows us to compare the fit of different models if the models are adequate. The smaller the AIC criterion, the better the model. As you can see in the image above, the AIC values are 3454.8 and 3455.17. Therefore, choosing the smaller value i.e., 3454.8 will be the appropriate criterion for our dataset.

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion:

In recent years, there has been a growing interest in using data analytics to analyse soil properties and predict soil moisture content in agriculture. Soil analysis and soil moisture prediction are essential components of precision agriculture, which aims to optimize crop yields while minimizing environmental impact.

It is important to note that soil analysis and soil moisture prediction models are not always accurate and require validation to ensure their reliability. Validation involves testing the model on independent datasets to assess its accuracy and generalizability. Cross-validation techniques, such as k-fold cross-validation, can be used to assess the model's performance and identify potential sources of error.

In conclusion, this soil analysis and soil moisture prediction project can provide valuable insights into soil properties and help optimize crop production. By leveraging data from various sources and using appropriate data analysis techniques, data analysts can develop accurate and reliable soil moisture prediction models that can inform irrigation scheduling and other management practices. However, it is important to validate these models to ensure their accuracy and usefulness in real-world scenarios. With proper validation and refinement, soil analysis and soil moisture prediction models have the potential to revolutionize agricultural practices and improve crop yields while minimizing environmental impact.

Future Scope:

The future scope for this soil analysis and soil moisture prediction projects is vast and has the potential to revolutionize agriculture. Here are some of the potential areas for future development:

1. Integration of additional data sources: The integration of additional data sources, such as drone imagery and soil spectral data, can enhance the accuracy of soil moisture prediction models. These data sources can provide high-resolution information on soil properties, which can be used to improve the accuracy and reliability of the models.

2. Use of advanced machine learning techniques: Advanced machine learning techniques, such as deep learning and reinforcement learning, can be used to develop more accurate and reliable soil moisture prediction models. These techniques can capture complex patterns in the data that may not be apparent using traditional machine learning algorithms.
3. Development of real-time monitoring systems: Real-time monitoring systems can be developed using IoT (Internet of Things) technologies and can provide continuous data on soil moisture content, temperature, and other parameters. These systems can help farmers optimize irrigation scheduling and other management practices in real-time, leading to more efficient and sustainable crop production.
4. Application of data analytics in precision agriculture: Precision agriculture involves the use of data analytics to optimize crop production, reduce waste, and improve environmental sustainability. The application of data analytics in precision agriculture can be expanded to include soil analysis and soil moisture prediction, leading to more efficient and sustainable farming practices.
5. Development of mobile applications: Mobile applications can be developed to provide farmers with real-time information on soil moisture content and other important parameters. These applications can help farmers make informed decisions about irrigation scheduling and other management practices while in the field.

In conclusion, the future scope for soil analysis and soil moisture prediction data analytics projects is promising, with potential areas for development in the integration of additional data sources, advanced machine learning techniques, real-time monitoring systems, precision agriculture, and mobile applications. With the development of these technologies and techniques, farmers can optimize crop production, reduce waste, and improve environmental sustainability, ultimately leading to a more food-secure future.

REFERENCES

1. https://arxiv.org/ftp/arxiv/papers/1503/1503.00900.pdf#:~:text=Normalization%20is%20used%20to%20eliminate,in%20the%20differences%5B3%5D.

2. http://www.fao.org/fileadmin/templates/rap/files/meetings/2016/160524_AMIS-CM_3.2.3_Crop_forecasting_Its_importance__current_approaches__ongoing_evolution_and.pdf

3. http://www.ijitee.org/wp-content/uploads/papers/v9i3/C8683019320.pdf

4. https://www.researchgate.net/post/Does_normalization_of_data_always_improve_the_clustering_results

5. https://stats.stackexchange.com/questions/21222/are-mean-normalization-and-feature-scaling-needed-for-k-means-clustering

6. https://journals.sagepub.com/doi/pdf/10.1177/1847979018808673

7. https://campus.datacamp.com/courses/cluster-analysis-in-r/k-means-clustering?ex=9

8. https://www.analyticssteps.com/blogs/what-k-means-clustering-machine-learning

9. https://ieeexplore.ieee.org/abstract/document/8290395

10. https://ieeexplore.ieee.org/abstract/document/9154036

11. https://statsandr.com/blog/clustering-analysis-k-means-and-hierarchical-clustering-by-hand-and-in-r/

12. https://www.researchgate.net/publication/318745067_Analysis_of_Soil_Samples_for_its_Physico-Chemical_Parameters_from_Kadi_City

13. "Soil Analysis Handbook of Reference Methods for Soil Analysis" by J. M. Bigham and A. M. Mielke. This book provides a comprehensive overview of soil analysis techniques and procedures, including soil moisture measurement and analysis.

14. "Soil Moisture Measurement: A Practical Handbook" by R. Campbell. This book provides a practical guide to soil moisture measurement, including different types of sensors and techniques for data analysis.

15. "Soil Moisture Measurement and Control for Agriculture" by W. P. Miller and S. R. Evett. This book provides an overview of the importance of soil moisture management in agriculture and describes different soil moisture measurement techniques and their applications.

16. "Machine Learning for Agriculture" by S. Prasad, V. Kumar, and M. K. Rakshit. This book provides an overview of machine learning techniques applied to agriculture, including soil moisture prediction models and their implementation.

17. "Data Analytics for Agriculture: Sensing, Methods and Applications" edited by K. D. Devi, S. Kumar, and S. S. Chaudhari. This book provides an overview of data analytics techniques applied to agriculture, including soil moisture prediction models and their implementation.

APPENDIX

R code:

```
soildata = read.csv("D:\\VIT\\sem6\\DV\\jcomp\\soilanalysis.csv")
soildata
View(soildata)
plot(pH~Aval.N,soildata,col="red",lwd=2)
with(soildata,text(pH~Aval.N,soildata,labels=name,pos=4,cex=1))


soildata1 = soildata[-1] #removes the name column for further
calculations
soildata1

#normalise the dataset.
m <- apply(soildata1,2,mean)
s <- apply(soildata1,2,sd)
soildata1 <- scale(soildata1,m,s)
soildata1
print(soildata1,digits = 3)


#elbow curve
wssplot <- function(data,nc=15,seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for(i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data,centers=i)$withinss)}
  plot(1:nc,wss,type="b",xlab="Number of clusters",
       ylab="Within Groups Sum of Squares",col="blue",lwd=2)}
wssplot(soildata1,nc=19,seed=1234)


#kmeans
testing_kmeans <- kmeans(soildata1,3)
testing_kmeans
```

```
library("scatterplot3d")
scatterplot3d(soildata1[,2:4],color = testing_kmeans$cluster,lwd=3)
scatterplot3d(soildata1[,2:4],color = "red",lwd=3)


distance <- dist(soildata1)#similarity matrix
print(distance, digits = 3)

#cluster dendogram
#complete
hc.c <- hclust(distance)
plot(hc.c,labels=soildata$name,lwd=2,col="blue")

hc.a <- hclust(distance,method = "average")#center-linkage
plot(hc.a,labels=soildata$name,lwd=2)

#cluster membership
member.c <- cutree(hc.c,3)#cutting the dendrogram tree into several
groups by specifying the
#desired number of cluster k(s)
member.a <- cutree(hc.a,3)
#plot(member.c,member.a)
table(member.c, member.a)


#cluster means
aggregate(soildata1,list(member.c),mean)

library(cluster)
#silhoute plot
plot(silhouette(cutree(hc.c,3),distance,color="blue"))



##TIME SERIES

library(ggfortify)  #for plotting time series data
library(tseries)
library(forecast)
data=read.csv("D:\\VIT\\sem6\\DV\\jcomp\\district_rainfall.csv")
View(data)
data$annual<-rowSums(data[-1])


g <- ggplot(data, aes(annual,STATE))
g + geom_bar(stat="identity", width = 0.5, fill="tomato2") +
  labs(title="Bar Chart",
       subtitle="RAINFALL STATS FOR EACH STATE",
       caption="Source: Frequency of Manufacturers from 'mpg' dataset") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6))

data_1 = data[2:13] #removes the name column for further calculations
data_1
data_1$id = rownames(data_1)
data_1
library(reshape2)
mm = melt(data_1, id='id')
library(ggplot2)
ggplot(mm)+geom_line(aes(x=variable, y=value, group=id, color=id))
```