NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

Nitte (DU) established under Section 3 of UGC Act 1956 | Accredited with 'A+' Grade by NAAC

Department of Computer Science & Engineering

Report on Mini Project

Covid-19 analysis: Understanding the pandemic

Course Code: CS1602-1
Course Name: Data Analysis using R Programmig

Semester: IIISEM Section: A

Submitted To:

Ms. Keerthana B C
Assistant Professor [GD-III]
Department of Computer Science and Engineering

Submitted By:

Varshith Pawar H R (NNM22CS200)
V Shreevasa Navada (NNM22CS194)
Vaishnavi Kamath (NNM22CS197)

Date of submission:
16/11/2023

Signature of Course Instructor

ABSTRACT

The COVID-19 pandemic, an unprecedented global health crisis, has necessitated comprehensive data analysis to better understand its impact on public health, society, and the world at large. This project delves into a thorough examination of COVID-19 data, encompassing a spectrum of critical aspects such as total cases, state-wise statistics, data exploration, visualization, and more. By employing advanced data analysis techniques and visualization tools, this endeavor aims to provide valuable insights into the pandemic's dynamics, trends, and implications.

The project initiates with the collection of up-to-date COVID-19 data from diverse sources, ensuring the reliability and relevance of the dataset. Rigorous data preprocessing procedures are undertaken to clean and prepare the data, enabling a robust foundation for in-depth analysis.

Exploratory data analysis is at the core of this endeavor, unraveling intricate patterns and relationships within the dataset. It unveils epidemiological trends, identifying regions of concern and potential hotspots. Demographic insights shed light on the virus's varying impact, while geospatial analyses contribute to localized decision-making.

As the project unfolds, it is vital to acknowledge that the COVID-19 situation remains fluid, and data continues to evolve. This analysis, while a snapshot in time, stands as a testament to the power of data in understanding and responding to a global crisis.

TABLE OF CONTENTS

INTRODUCTION

In the wake of the COVID-19 pandemic, understanding the virus's dynamics, spread, and impact has become a paramount global concern. The project "COVID-19 Data Analysis: Understanding the Pandemic" embarks on a mission to shed light on this complex crisis by harnessing the analytical power of the R programming language. This project serves as a vital tool in unraveling the intricacies of COVID-19, offering valuable insights that can inform public health decisions, guide policies, and contribute to the collective effort in combating this global health crisis.

At its core, this endeavor focuses on three key pillars: data collection, data preprocessing, and exploratory data analysis. These pillars, when executed meticulously, create a robust foundation for the in-depth analysis of COVID-19 data.

The project begins with the meticulous gathering of COVID-19 data from authoritative sources, ensuring the accuracy and relevance of the information at hand. Our reliance on credible repositories, government health agencies, and research institutions guarantees the integrity of the data we analyze.

Data preprocessing is the next vital step. It involves the meticulous cleaning and transformation of the collected data, ensuring that any anomalies, missing values, or inconsistencies are addressed. This stage is imperative in providing reliable data for further analysis.

Finally, we embark on an exploratory data analysis journey, where we delve into the depths of the COVID-19 dataset. Through data visualization and statistical analysis, we aim to uncover patterns, correlations, and trends that offer a preliminary understanding of the pandemic's nuances.

This project invites you to join us in our quest to decipher the COVID-19 pandemic through the lens of data. Together, we will leverage the powerful capabilities of the R programming language to provide clarity, knowledge, and actionable insights, ultimately contributing to our collective understanding of this global health challenge

PROBLEM STATEMENT

The COVID-19 pandemic, caused by the novel coronavirus, has emerged as one of the most significant global challenges of our time. Its rapid spread, profound impact on public health, economies, and societies worldwide necessitates a comprehensive and data-driven approach to understand, mitigate, and respond effectively. Despite the abundance of COVID-19 data, challenges persist in data analysis and insights extraction to inform evidence-based decision-making and strategies for combating the pandemic.

Create a project seeks to bridge these gaps by leveraging the capabilities of the R programming language to conduct thorough data analysis, visualization, and exploration. By doing so, aim to provide meaningful and actionable knowledge to address the global challenges posed by the pandemic.

OBJECTIVES

1.Data Import and Preparation:

Read the COVID-19 data from a CSV file, ensuring data integrity and completeness.

2.State-Wise Analysis:

Determine the total cases and identify states with the maximum single-day cases, providing insights into regional variations.

3.Statistical Visualization:

Plot graphs showcasing the relationship between states and the mean of confirmed cases, offering a visual representation of the pandemic's intensity across regions.

4.Data Quality Check:

Identify and address missing values (NAs) within the dataset, ensuring the accuracy and reliability of the data for subsequent analysis.

5.Numeric Data Extraction:

Extract only numeric columns from the dataset, streamlining the data for focused analysis on relevant variables.

6.Pie Chart Representation:

Create a pie chart to visualize the distribution of confirmed, cured, and death cases in India, providing a concise overview of the pandemic's impact.

7.Exploration of Numeric Data:

Obtain value counts and unique values of numeric columns, gaining a deeper understanding of the data distribution and variability.

8.State-Specific Insights:

Analyze total cases and identify states with maximum cases, offering insights into regional trends and potential hotspots.

Temporal Analysis with Line Graphs:

9.Create line graphs depicting the temporal evolution of confirmed, cured, and death cases, enabling a dynamic visualization of the pandemic's progression over time.

By achieving these objectives, the project aims to contribute meaningful insights into the COVID-19 pandemic. The utilization of R programming for data analysis and visualization ensures a comprehensive exploration of the dataset, providing a foundation for informed decision-making and a deeper understanding of the pandemic's multifaceted aspects.

METHODOLOGY

1. Data Collection:

-Gather COVID-19 data from credible sources, such as government health agencies, research institutions, or global databases, ensuring the dataset is reliable, up-to-date, and comprehensive. Here we have used the dataset from Kaggle.

2. Data Import and Initial Exploration:

-Read the CSV file containing COVID-19 data into R. Using the function read.csv.
Perform initial exploratory data analysis (EDA) to understand the structure and content of the dataset that is summary, column names and classes

3. State-Wise Analysis:

-Calculate total cases and identify states with maximum single-day cases, providing an overview of the regional impact of COVID-19. We have used max and summary functions.

4. Statistical Visualization:

-Plot graphs of state versus the mean of confirmed cases, utilizing R's visualization capabilities to identify patterns and trends across regions. We have used ggplot2 library and bar graph to represent this

5. Data Quality Check:

-Identify columns with missing values (NAs) and implement appropriate strategies to handle them, ensuring data consistency and reliability.

6. Numeric Data Extraction:

-Extract only numeric columns from the dataset, focusing the analysis on relevant quantitative variables.

7. Pie Chart Representation:

-Create a pie chart to visually represent the distribution of confirmed, cured, and death cases in India, offering a concise summary of the pandemic's impact.

8. Exploration of Numeric Data:

-Obtain value counts and unique values of numeric columns, providing insights into the distribution and variability of key variables.

9. State-Specific Insights:

-Analyze total cases and identify states with maximum cases, offering insights into regional trends and potential hotspots.

10. Temporal Analysis with Line Graphs:

- Create line graphs for confirmed, cured, and death cases, allowing for a temporal analysis to visualize the progression of the pandemic over time.

11. Documentation and Reporting:

- Document all code, steps, and findings for reproducibility and transparency.

- Generate a comprehensive report summarizing key insights, visualizations, and potential implications based on the data analysis. The complete documentation can be found in git hub repository :(**https://github.com/VarshithPawarHR/COVID-19-Insights-R**).

12. Iterative Refinement:

- Review and refine the analysis iteratively, incorporating feedback and adjusting methods as needed for a robust and insightful analysis.

By following this methodology, the project aims to extract valuable insights from the COVID-19 data using the R programming language, contributing to a better understanding of the pandemic's various dimensions.

IMPLEMENTATION

1. Data Loading and Initial Exploration:

```
# Load the required libraries

library(dplyr)
library("ggplot2")
library(lubridate)
library("ggpubr")
library(“scales”)

# Read the CSV file

df_India <- read.csv(“file name”)

# Check class, structure, column names, and summary statistics

class(df_India)
str(df_India)
colnames(df_India)
summary(df_India)
```

2. State-Wise Analysis:

```
# Calculate total cases and identify states with maximum single-day cases

df_India %>%
  group_by(State.unionTerritory) %>%
  summarise(cases_sum = sum(Confirmed), cases_max = max(Confirmed)) %>%
  arrange(desc(cases_sum))
```

3. Statistical Visualization:

```
# Plot graph of state versus mean of confirmed cases

df_India %>%
  group_by(State.unionTerritory) %>%
  summarise(mean_conf = mean(Confirmed)) %>%
  ggplot(aes(x = State.unionTerritory, y = mean_conf, fill = State.unionTerritory)) +
  geom_bar(stat = "identity") +
  scale_x_discrete(guide = guide_axis(n.dodge = 13)) +
  theme_classic() +
  labs(
    x = "State/UnionTerritory",
    y = "Mean of Confirmed Cases",
    title = "Grouped by State/UnionTerritory with Summarise()"
  )
```

4. Identifying Columns with Missing Values:

```
# Identify columns with missing values

colnames(df_India)[apply(df_India, 2, anyNA)]
unique(df_India$State.unionTerritory)
```

5. Extracting Only Numeric Columns:

```
# Select numeric columns

num <- df_India %>%
  select_if(is.numeric)
# Print the numeric columns
print(num)
```

6. Creating a Pie Chart:

```
# Convert Date to Date format

df_India$Date <- as.Date(df_India$Date, format = "%d/%m/%Y")

# Summarize the total cases
confirm <- sum(df_India$Confirmed)
cured <- sum(df_India$Cured)
deaths <- sum(df_India$Deaths)

# Create a data frame for the pie chart
df_value <- c(confirm, cured, deaths)
```

```
df_key <- c("Confirmed", "Cured", "Deaths")

# Create a pie chart

pie_chart <- ggplot(data = data.frame(df_key, df_value), aes(x = "", y = df_value, fill =
df_key)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  scale_fill_manual(values = c("Confirmed" = "lightblue", "Cured" = "orange", "Deaths"
= "red")) +
  labs(title = "COVID-19 Cases Distribution",
       fill = "Category",
       x = NULL,
       y = NULL) +
  theme_void() +
  theme(legend.position = "right")

print(pie_chart)
```

7. Getting Value Counts and Unique Values of Numeric Columns:

```
# Select numeric columns

num_cols <- df_India %>%
  select_if(is.numeric) %>%
  colnames()

# Iterate through numeric columns

for (col in num_cols) {
  cat("Column: ", col, "\n")
  cat("Value counts of", col, ": ", sum(!is.na(df_India[[col]]), "\n"))
  cat("Number of Unique Values in", col, ": ", n_distinct(df_India[[col]]), "\n\n")
}
```

8. Total Cases and Maximum Cases in Particular States:

```
df_India[df_India$State.unionTerritory == "Bihar****", ]
df_India[df_India$State.unionTerritory == "Karnataka", ]
df_India[df_India$State.unionTerritory == "Maharastra", ]
```

```
df_India[df_India$State.unionTerritory == "Kerala", ]
```

9. Creating Line Graphs:

```
# Create a data frame

df <- data.frame(
  x = c('Confirmed', 'Cured', 'Deaths'),
  y = c(confirm, cured, deaths)
)

# Create a line graph

lineplot <- ggplot(data = df, aes(x = x, y = y)) +
  geom_line(aes(group = 1), color = "steelblue") +
  scale_y_continuous(labels = comma) +
  labs(x = "", y = "") +
  theme_minimal()

print(lineplot).
```

INFO ON LIBRARIES USED:

1.dplyr:

- dplyr is a popular data manipulation package that provides a set of functions for data manipulation and
transformation. It simplifies data manipulation tasks like filtering, summarizing, mutating (creating new variables), and arranging data.
- *Common Functions:* filter(), select(), mutate(), summarize(), arrange(), and more.
- *Example Use:* You can use dplyr to filter rows based on conditions, calculate summary statistics, create new columns, and
reorder data.

2.ggplot2:

ggplot2 is a powerful and flexible package for creating data visualizations and plots. It follows the Grammar of Graphics framework, allowing you to build

complex and customized visualizations.
- *Common Functions:* ggplot(), geom_point(), geom_line(), geom_bar(), facet_wrap(), and more.
- *Example Use:* You can use ggplot2 to create a wide range of plots, including scatterplots,linecharts, bar charts, and more, with a high degree of customization.

3.lubridate:

lubridate is a package designed to make working with dates and times in R easier. It provides functions for parsing, formatting, and performing operations on date and time objects.

- *Common Functions:* ymd(), mdy(), hms(), year(), month(), day(), and more.
- *Example Use:* lubridate simplifies tasks like converting date strings to date objects, extracting components of dates and times, and performing arithmetic with dates and times.

4.ggpubr:

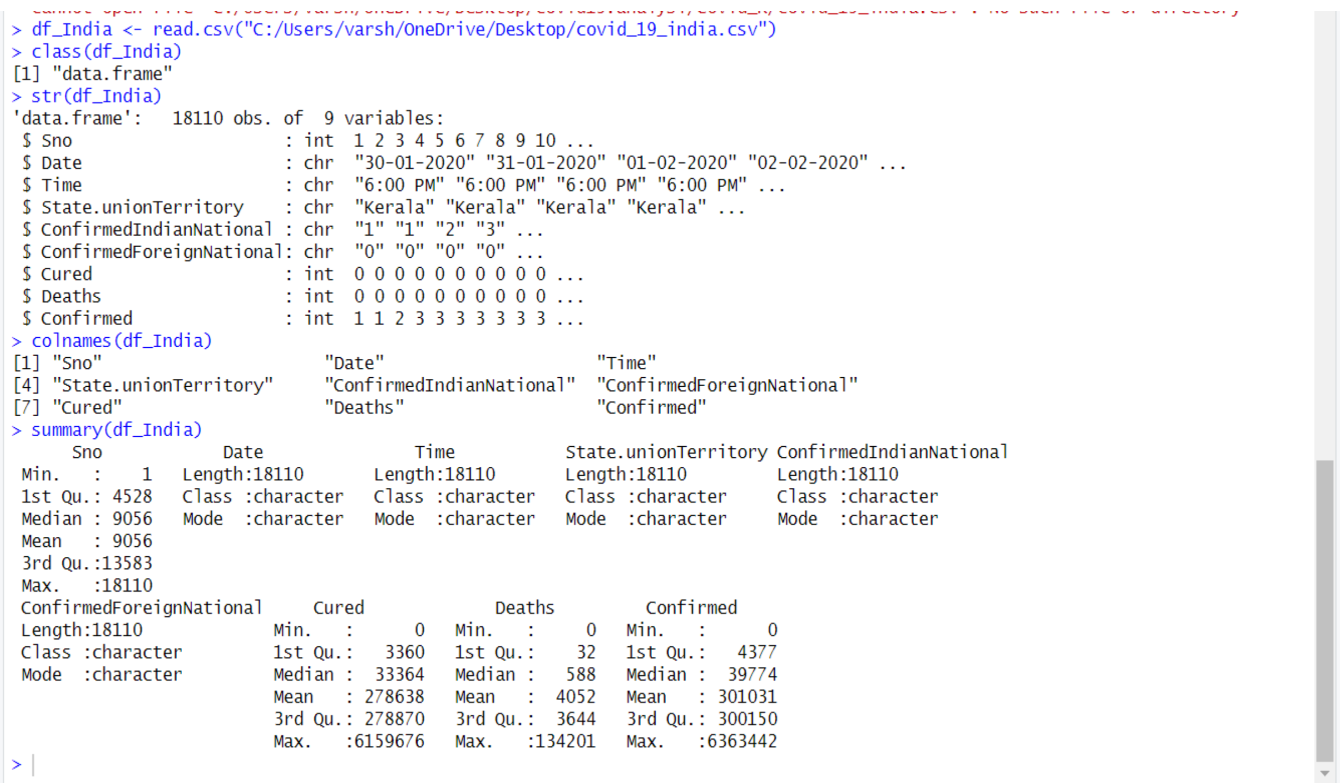
ggpubr is an extension package for ggplot2 that provides additional functions for creating publication-ready plots and enhancing the appearance of ggplot2 plots.
- *Common Functions:* stat_compare_means(), theme_pubr(), annotate_figure() etc.
- *Example Use:* ggpubr offers functions for adding statistical comparisons to your plots, adjusting plot themes, and creating complex layouts for multiple plots.

5.scales:

The scales package provides functions for customizing and formatting scales in ggplot2 plots. It allows you to control the appearance of axes, legends.
- *Common Functions:* comma(), percent_format(), scale_x_continuous(), scale_color_manual(), and more.
- *Example Use:* scales is often used in conjunction with ggplot2 to format axis labels, apply custom number formats, and control the colour scales in plots.

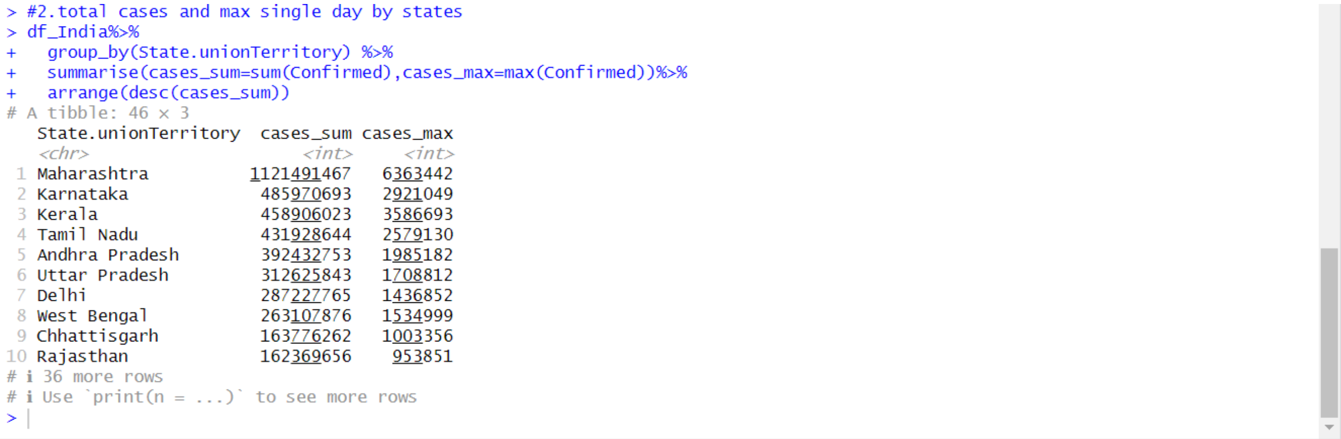
RESULTS AND DISCUSSIONS

1.Data import and basic analysis:



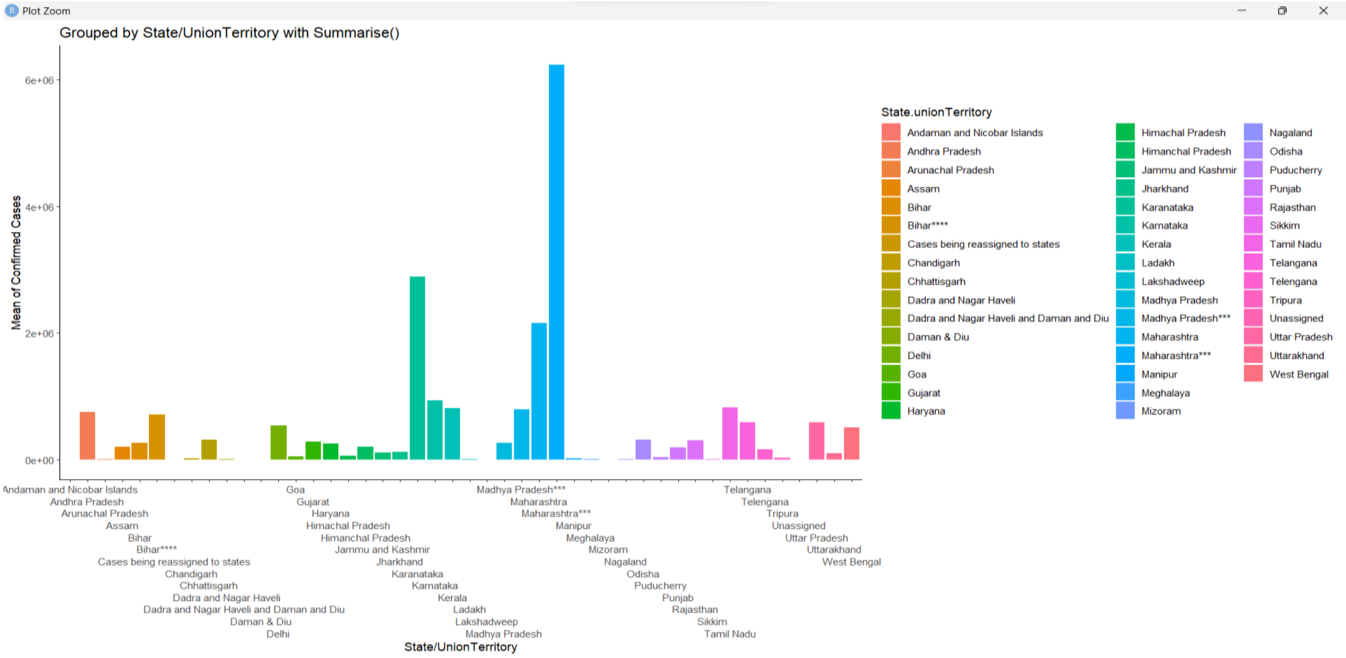
```
  cannot open file 'C:/Users/varsh/OneDrive/Desktop/covidAnalysis/covid_R/covid_19_india.csv': No such file or directory
> df_India <- read.csv("C:/Users/varsh/OneDrive/Desktop/covid_19_india.csv")
> class(df_India)
[1] "data.frame"
> str(df_India)
'data.frame':	18110 obs. of  9 variables:
 $ Sno                     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Date                    : chr  "30-01-2020" "31-01-2020" "01-02-2020" "02-02-2020" ...
 $ Time                    : chr  "6:00 PM" "6:00 PM" "6:00 PM" "6:00 PM" ...
 $ State.unionTerritory    : chr  "Kerala" "Kerala" "Kerala" "Kerala" ...
 $ ConfirmedIndianNational : chr  "1" "1" "2" "3" ...
 $ ConfirmedForeignNational: chr  "0" "0" "0" "0" ...
 $ Cured                   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Deaths                  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Confirmed               : int  1 1 2 3 3 3 3 3 3 3 ...
> colnames(df_India)
[1] "Sno"                      "Date"                     "Time"                    
[4] "State.unionTerritory"     "ConfirmedIndianNational"  "ConfirmedForeignNational"
[7] "Cured"                    "Deaths"                   "Confirmed"               
> summary(df_India)
      Sno            Date               Time           State.unionTerritory ConfirmedIndianNational
 Min.   :    1   Length:18110       Length:18110       Length:18110         Length:18110           
 1st Qu.: 4528   Class :character   Class :character   Class :character     Class :character       
 Median : 9056   Mode  :character   Mode  :character   Mode  :character     Mode  :character       
 Mean   : 9056                                                                                     
 3rd Qu.:13583                                                                                     
 Max.   :18110                                                                                     
 ConfirmedForeignNational     Cured             Deaths          Confirmed      
 Length:18110             Min.   :      0   Min.   :     0   Min.   :      0  
 Class :character         1st Qu.:   3360   1st Qu.:    32   1st Qu.:   4377  
 Mode  :character         Median :  33364   Median :   588   Median :  39774  
                          Mean   : 278638   Mean   :  4052   Mean   : 301031  
                          3rd Qu.: 278870   3rd Qu.:  3644   3rd Qu.: 300150  
                          Max.   :6159676   Max.   :134201   Max.   :6363442  
> |
```

2.Total cases and max single day by states:

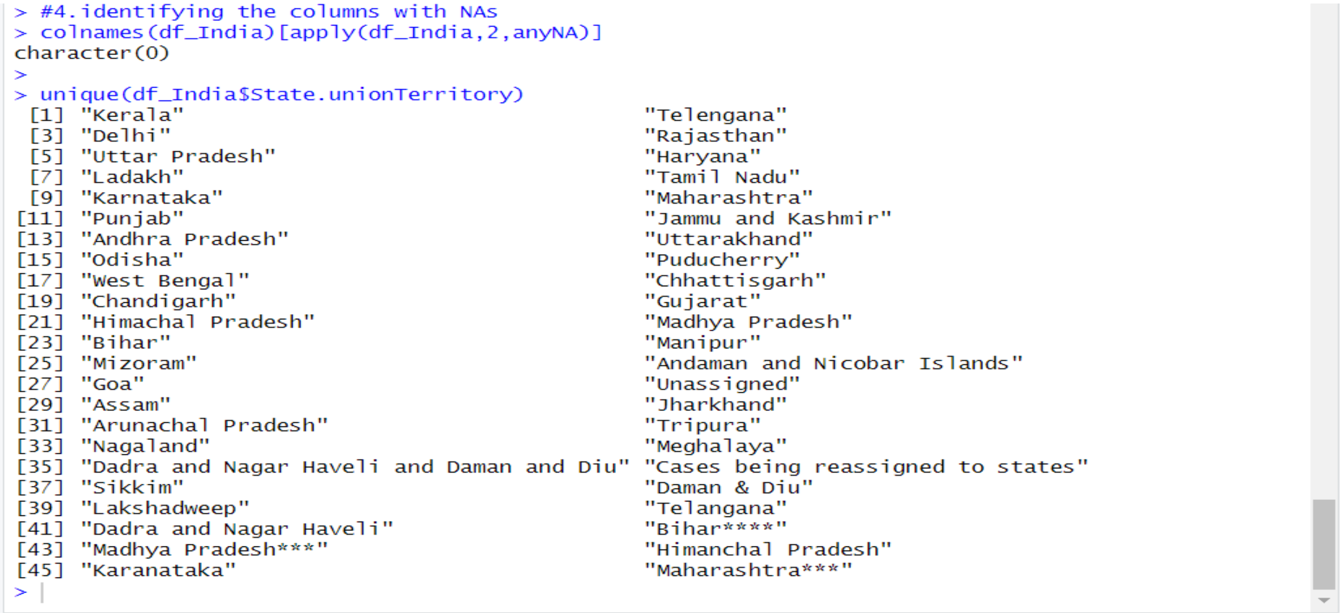


```
> #2.total cases and max single day by states
> df_India%>%
+   group_by(State.unionTerritory) %>%
+   summarise(cases_sum=sum(Confirmed),cases_max=max(Confirmed))%>%
+   arrange(desc(cases_sum))
# A tibble: 46 × 3
   State.unionTerritory  cases_sum cases_max
   <chr>                     <int>     <int>
 1 Maharashtra          1121491467   6363442
 2 Karnataka             485970693   2921049
 3 Kerala                458906023   3586693
 4 Tamil Nadu            431928644   2579130
 5 Andhra Pradesh        392432753   1985182
 6 Uttar Pradesh         312625843   1708812
 7 Delhi                 287227765   1436852
 8 West Bengal           263107876   1534999
 9 Chhattisgarh          163776262   1003356
10 Rajasthan             162369656    953851
# ℹ 36 more rows
# ℹ Use `print(n = ...)` to see more rows
> |
```

3. Graph of state vs mean of confirmed cases:

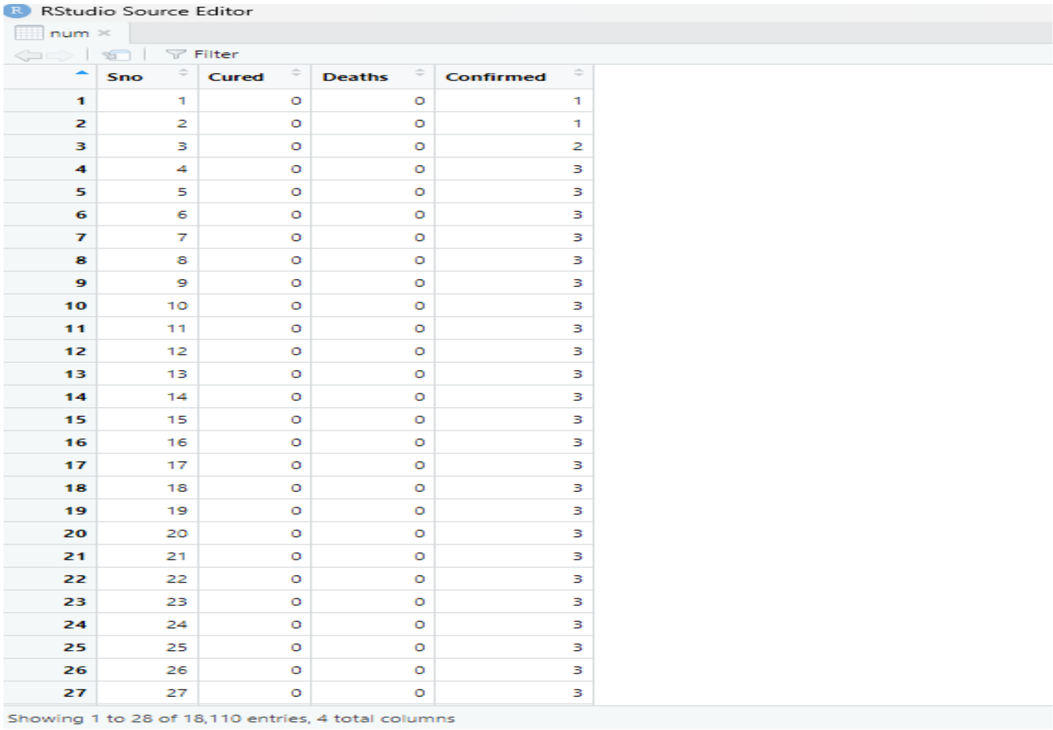


4.Identifying columns with NAs



```
> #4.identifying the columns with NAs
> colnames(df_India)[apply(df_India,2,anyNA)]
character(0)
>
> unique(df_India$State.unionTerritory)
 [1] "Kerala"                                    "Telengana"
 [3] "Delhi"                                     "Rajasthan"
 [5] "Uttar Pradesh"                             "Haryana"
 [7] "Ladakh"                                    "Tamil Nadu"
 [9] "Karnataka"                                 "Maharashtra"
[11] "Punjab"                                    "Jammu and Kashmir"
[13] "Andhra Pradesh"                            "Uttarakhand"
[15] "Odisha"                                    "Puducherry"
[17] "West Bengal"                               "Chhattisgarh"
[19] "Chandigarh"                                "Gujarat"
[21] "Himachal Pradesh"                          "Madhya Pradesh"
[23] "Bihar"                                     "Manipur"
[25] "Mizoram"                                   "Andaman and Nicobar Islands"
[27] "Goa"                                       "Unassigned"
[29] "Assam"                                     "Jharkhand"
[31] "Arunachal Pradesh"                         "Tripura"
[33] "Nagaland"                                  "Meghalaya"
[35] "Dadra and Nagar Haveli and Daman and Diu"  "Cases being reassigned to states"
[37] "Sikkim"                                    "Daman & Diu"
[39] "Lakshadweep"                               "Telangana"
[41] "Dadra and Nagar Haveli"                    "Bihar****"
[43] "Madhya Pradesh***"                         "Himanchal Pradesh"
[45] "Karanataka"                                "Maharashtra***"
>
```

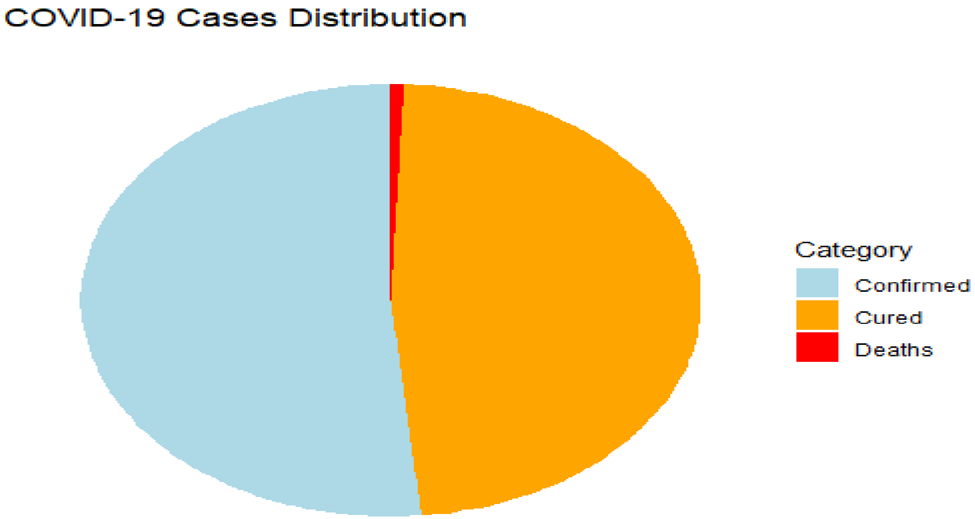
5.Getting only numeric columns from the dataset:



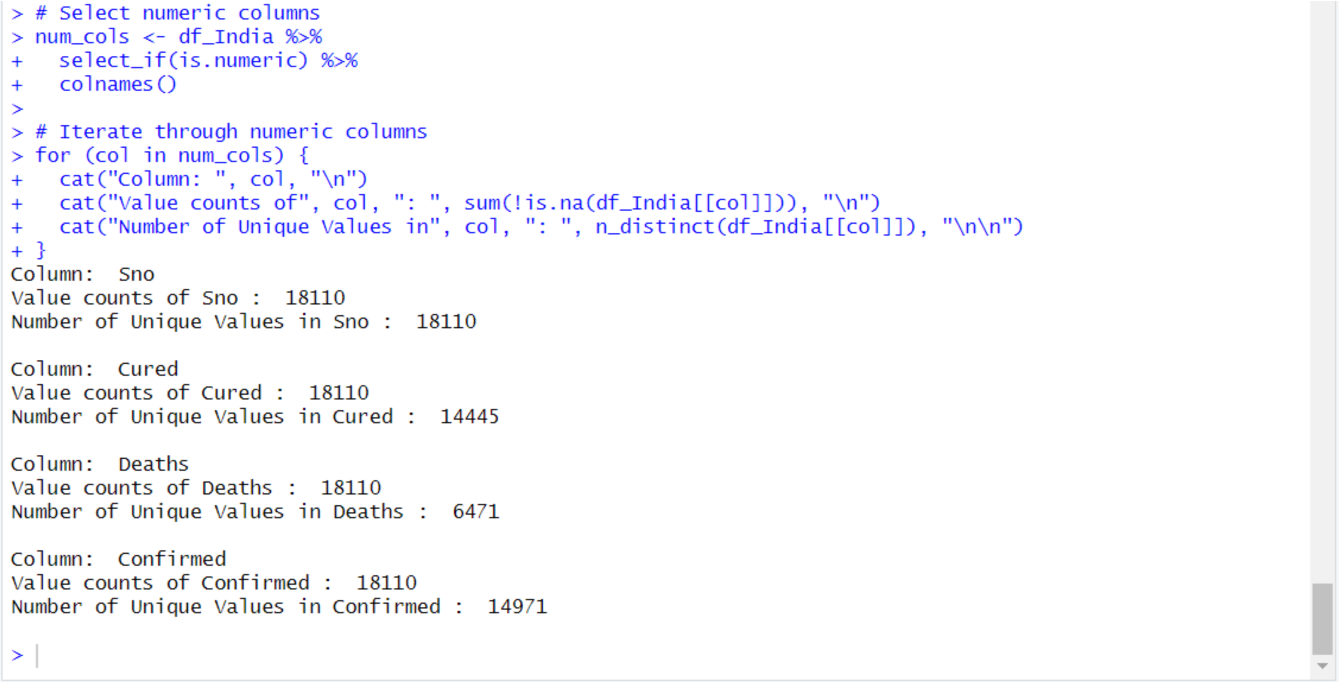
	Sno	Cured	Deaths	Confirmed
1	1	0	0	1
2	2	0	0	1
3	3	0	0	2
4	4	0	0	3
5	5	0	0	3
6	6	0	0	3
7	7	0	0	3
8	8	0	0	3
9	9	0	0	3
10	10	0	0	3
11	11	0	0	3
12	12	0	0	3
13	13	0	0	3
14	14	0	0	3
15	15	0	0	3
16	16	0	0	3
17	17	0	0	3
18	18	0	0	3
19	19	0	0	3
20	20	0	0	3
21	21	0	0	3
22	22	0	0	3
23	23	0	0	3
24	24	0	0	3
25	25	0	0	3
26	26	0	0	3
27	27	0	0	3

Showing 1 to 28 of 18,110 entries, 4 total columns

6.pie chart for the confirmed, cured and deaths:

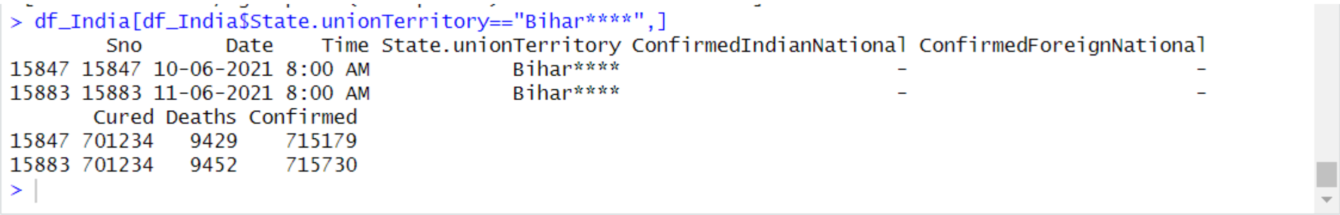


7.To get the value of numeric columns and the unique value in the columns:

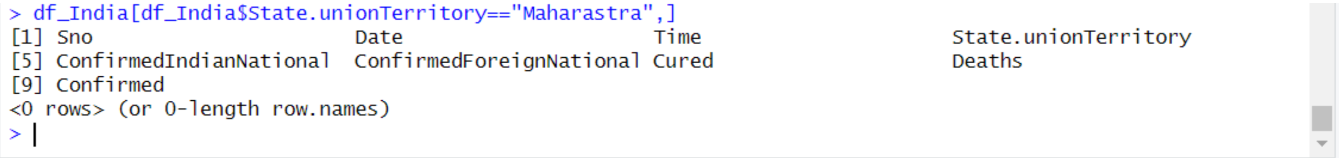


```
> # Select numeric columns
> num_cols <- df_India %>%
+   select_if(is.numeric) %>%
+   colnames()
>
> # Iterate through numeric columns
> for (col in num_cols) {
+   cat("Column: ", col, "\n")
+   cat("Value counts of", col, ": ", sum(!is.na(df_India[[col]])), "\n")
+   cat("Number of Unique Values in", col, ": ", n_distinct(df_India[[col]]), "\n\n")
+ }
Column:  Sno
Value counts of Sno :  18110
Number of Unique Values in Sno :  18110

Column:  Cured
Value counts of Cured :  18110
Number of Unique Values in Cured :  14445

Column:  Deaths
Value counts of Deaths :  18110
Number of Unique Values in Deaths :  6471

Column:  Confirmed
Value counts of Confirmed :  18110
Number of Unique Values in Confirmed :  14971

> 
```

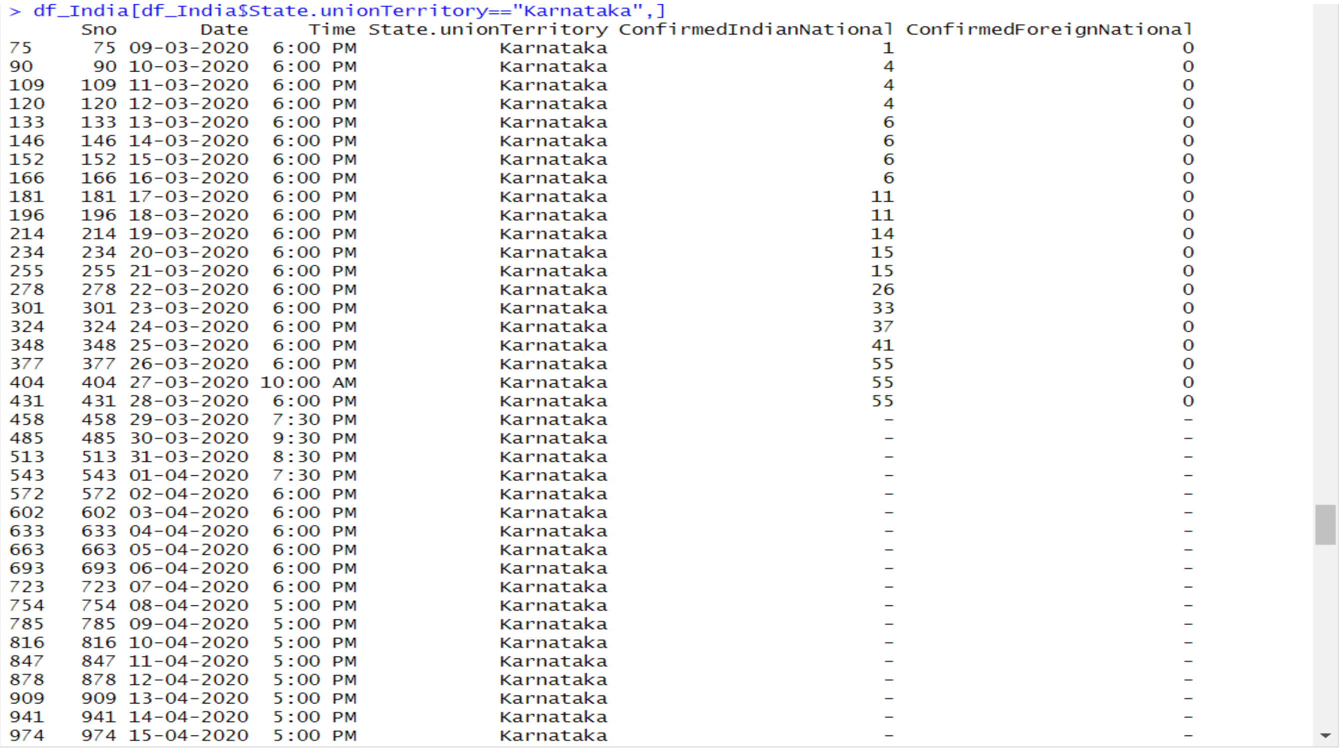
8.Total cases and max cases in particular states:



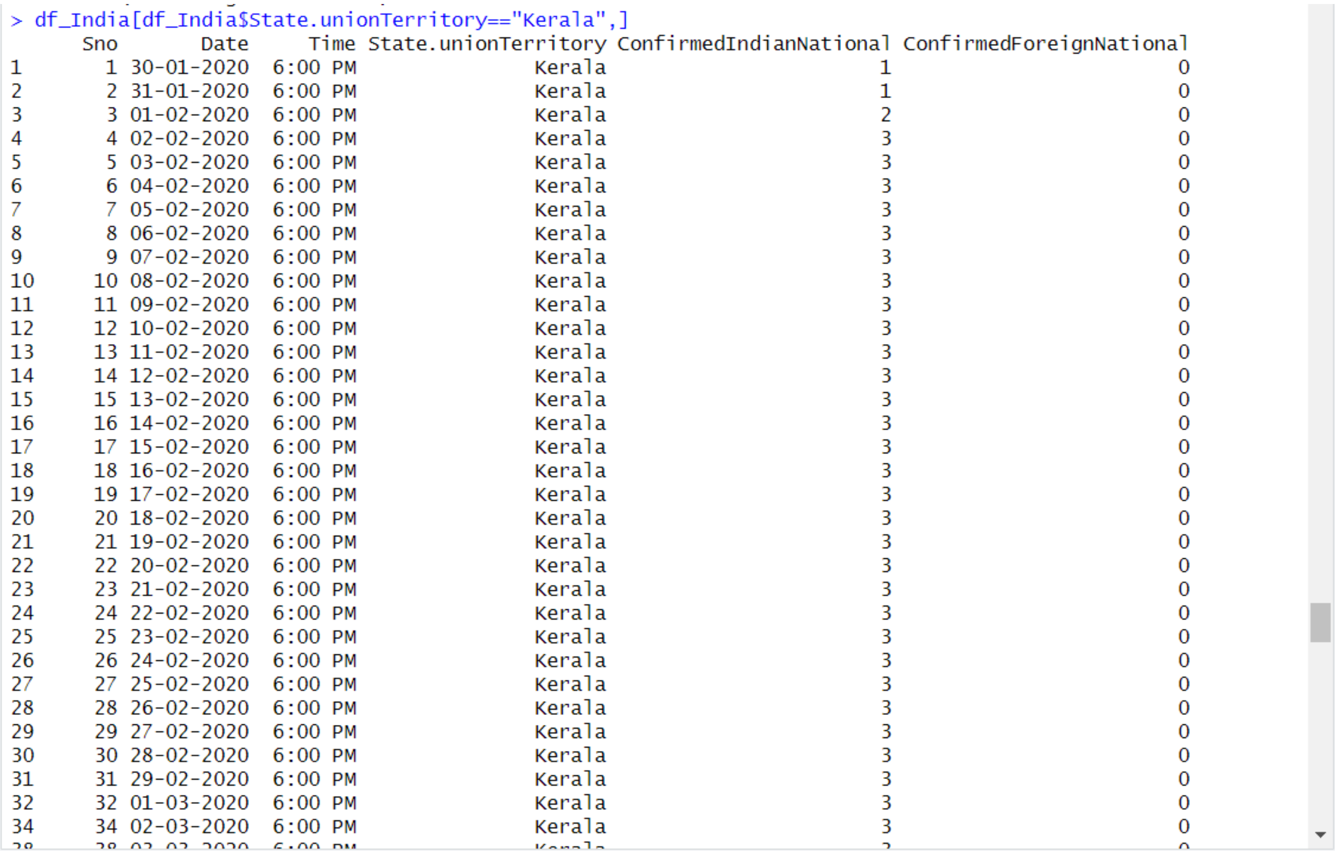
```
> df_India[df_India$State.unionTerritory=="Bihar****",]
        Sno       Date    Time State.unionTerritory ConfirmedIndianNational ConfirmedForeignNational
15847 15847 10-06-2021 8:00 AM            Bihar****                       -                        -
15883 15883 11-06-2021 8:00 AM            Bihar****                       -                        -
       Cured Deaths Confirmed
15847 701234   9429    715179
15883 701234   9452    715730
> 
```



```
> df_India[df_India$State.unionTerritory=="Maharastra",]
[1] Sno                     Date                     Time                     State.unionTerritory
[5] ConfirmedIndianNational ConfirmedForeignNational Cured                    Deaths
[9] Confirmed
<0 rows> (or 0-length row.names)
> 
```

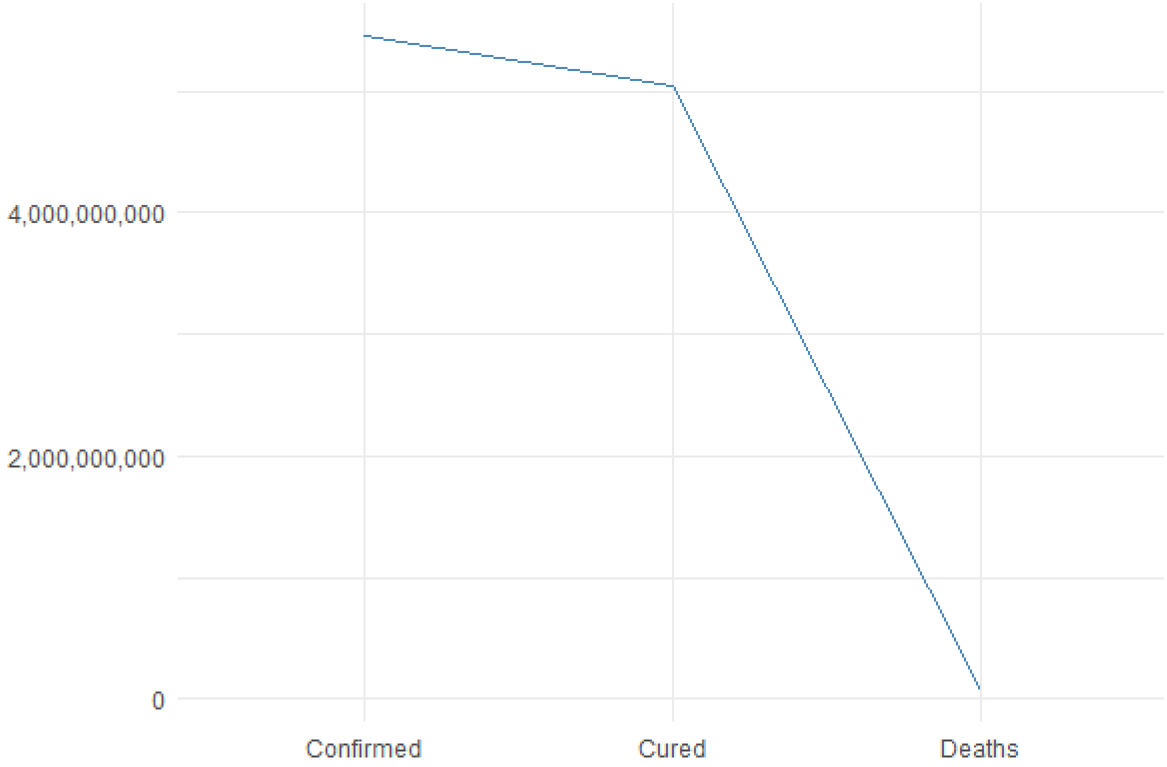


```
> df_India[df_India$State.unionTerritory=="Karnataka",]
    Sno       Date     Time State.unionTerritory ConfirmedIndianNational ConfirmedForeignNational
75   75 09-03-2020  6:00 PM            Karnataka                       1                        0
90   90 10-03-2020  6:00 PM            Karnataka                       4                        0
109 109 11-03-2020  6:00 PM            Karnataka                       4                        0
120 120 12-03-2020  6:00 PM            Karnataka                       4                        0
133 133 13-03-2020  6:00 PM            Karnataka                       6                        0
146 146 14-03-2020  6:00 PM            Karnataka                       6                        0
152 152 15-03-2020  6:00 PM            Karnataka                       6                        0
166 166 16-03-2020  6:00 PM            Karnataka                       6                        0
181 181 17-03-2020  6:00 PM            Karnataka                      11                        0
196 196 18-03-2020  6:00 PM            Karnataka                      11                        0
214 214 19-03-2020  6:00 PM            Karnataka                      14                        0
234 234 20-03-2020  6:00 PM            Karnataka                      15                        0
255 255 21-03-2020  6:00 PM            Karnataka                      15                        0
278 278 22-03-2020  6:00 PM            Karnataka                      26                        0
301 301 23-03-2020  6:00 PM            Karnataka                      33                        0
324 324 24-03-2020  6:00 PM            Karnataka                      37                        0
348 348 25-03-2020  6:00 PM            Karnataka                      41                        0
377 377 26-03-2020  6:00 PM            Karnataka                      55                        0
404 404 27-03-2020 10:00 AM            Karnataka                      55                        0
431 431 28-03-2020  6:00 PM            Karnataka                      55                        0
458 458 29-03-2020  7:30 PM            Karnataka                       -                        -
485 485 30-03-2020  9:30 PM            Karnataka                       -                        -
513 513 31-03-2020  8:30 PM            Karnataka                       -                        -
543 543 01-04-2020  7:30 PM            Karnataka                       -                        -
572 572 02-04-2020  6:00 PM            Karnataka                       -                        -
602 602 03-04-2020  6:00 PM            Karnataka                       -                        -
633 633 04-04-2020  6:00 PM            Karnataka                       -                        -
663 663 05-04-2020  6:00 PM            Karnataka                       -                        -
693 693 06-04-2020  6:00 PM            Karnataka                       -                        -
723 723 07-04-2020  6:00 PM            Karnataka                       -                        -
754 754 08-04-2020  5:00 PM            Karnataka                       -                        -
785 785 09-04-2020  5:00 PM            Karnataka                       -                        -
816 816 10-04-2020  5:00 PM            Karnataka                       -                        -
847 847 11-04-2020  5:00 PM            Karnataka                       -                        -
878 878 12-04-2020  5:00 PM            Karnataka                       -                        -
909 909 13-04-2020  5:00 PM            Karnataka                       -                        -
941 941 14-04-2020  5:00 PM            Karnataka                       -                        -
974 974 15-04-2020  5:00 PM            Karnataka                       -                        -
```



```
> df_India[df_India$State.unionTerritory=="Kerala",]
    Sno       Date     Time State.unionTerritory ConfirmedIndianNational ConfirmedForeignNational
1     1 30-01-2020  6:00 PM               Kerala                       1                        0
2     2 31-01-2020  6:00 PM               Kerala                       1                        0
3     3 01-02-2020  6:00 PM               Kerala                       2                        0
4     4 02-02-2020  6:00 PM               Kerala                       3                        0
5     5 03-02-2020  6:00 PM               Kerala                       3                        0
6     6 04-02-2020  6:00 PM               Kerala                       3                        0
7     7 05-02-2020  6:00 PM               Kerala                       3                        0
8     8 06-02-2020  6:00 PM               Kerala                       3                        0
9     9 07-02-2020  6:00 PM               Kerala                       3                        0
10   10 08-02-2020  6:00 PM               Kerala                       3                        0
11   11 09-02-2020  6:00 PM               Kerala                       3                        0
12   12 10-02-2020  6:00 PM               Kerala                       3                        0
13   13 11-02-2020  6:00 PM               Kerala                       3                        0
14   14 12-02-2020  6:00 PM               Kerala                       3                        0
15   15 13-02-2020  6:00 PM               Kerala                       3                        0
16   16 14-02-2020  6:00 PM               Kerala                       3                        0
17   17 15-02-2020  6:00 PM               Kerala                       3                        0
18   18 16-02-2020  6:00 PM               Kerala                       3                        0
19   19 17-02-2020  6:00 PM               Kerala                       3                        0
20   20 18-02-2020  6:00 PM               Kerala                       3                        0
21   21 19-02-2020  6:00 PM               Kerala                       3                        0
22   22 20-02-2020  6:00 PM               Kerala                       3                        0
23   23 21-02-2020  6:00 PM               Kerala                       3                        0
24   24 22-02-2020  6:00 PM               Kerala                       3                        0
25   25 23-02-2020  6:00 PM               Kerala                       3                        0
26   26 24-02-2020  6:00 PM               Kerala                       3                        0
27   27 25-02-2020  6:00 PM               Kerala                       3                        0
28   28 26-02-2020  6:00 PM               Kerala                       3                        0
29   29 27-02-2020  6:00 PM               Kerala                       3                        0
30   30 28-02-2020  6:00 PM               Kerala                       3                        0
31   31 29-02-2020  6:00 PM               Kerala                       3                        0
32   32 01-03-2020  6:00 PM               Kerala                       3                        0
34   34 02-03-2020  6:00 PM               Kerala                       3                        0
38   38 03-03-2020  6:00 PM               Kerala                       3                        0
```

9.Line graph for confirmed for confirmed, cured and deaths:



The analysis of COVID-19 data has unearthed critical insights into regional variations and trends. High caseloads in states like Maharashtra underscore the uneven impact of the pandemic. The mean confirmed cases visualization highlights the importance of understanding the varying intensity across different states. Addressing missing values ensures the robustness of our findings. Specific insights into states like Bihar and Karnataka allow for targeted responses. The temporal analysis reveals the dynamic nature of the pandemic, emphasizing the need for real-time monitoring. These findings not only contribute to our understanding of the current situation but also provide actionable information for effective public health strategies.

CONCLUSION AND FUTURE SCOPE

In conclusion, the analysis of COVID-19 data using R programming has provided valuable insights into the multifaceted dynamics of the pandemic. The state-wise analysis uncovered varying levels of impact, with certain regions emerging as significant hotspots. Visualization of mean confirmed cases highlighted the importance of regional nuances, guiding the formulation of targeted strategies. Addressing data quality issues, such as missing values, enhances the reliability of our findings.

The exploration of numeric data offered a deeper understanding of the dataset's characteristics, contributing to a more nuanced interpretation. State-specific insights for Bihar, Karnataka, Maharashtra, and Kerala provide a granular view, aiding in regional response planning. Temporal analysis through line graphs depicted the evolving nature of the pandemic, emphasizing the need for adaptive and real-time decision-making.

Future Scope:

The analysis sets the stage for future investigations and enhancements:

1.Predictive Modeling:

Implementing predictive models can forecast potential outbreaks, enabling proactive measures and resource allocation.

2.Demographic Analysis:

Integrating demographic data can provide insights into population-specific vulnerabilities, aiding in targeted healthcare interventions.

3.Spatial Analysis:

Geographic Information System (GIS) mapping can offer a spatial perspective, helping identify geographical clusters and optimize resource distribution.

4.Impact Assessment:

Evaluating the socio-economic impact of the pandemic on communities can inform long-term recovery plans.

5.Collaboration and Data Sharing:

Collaborative efforts with other research initiatives and data-sharing platforms can enrich the dataset, leading to more comprehensive analyses.

6.Public Health Interventions:

Integrating insights into public health policies and interventions, ensuring an evidence-based and adaptive approach to managing the ongoing and future health crises.

This analysis serves as a foundation for a continuous and evolving understanding of the COVID-19 pandemic, emphasizing the importance of data-driven decision-making and collaborative efforts in mitigating its impact.

REFERENCES

[1] **https://www.kaggle.com/code/atharvakadam333/eda-information-of-covid-19-india**

[2] **https://www.mygov.in/covid-19/**

[3] **https://covid19.india.gov.in/**

[4] **https://en.wikipedia.org/wiki/COVID-19_pandemic_in_India**

[5] **https://www.geeksforgeeks.org/graph-plotting-in-r-programming/**

[6] **https://www.mohfw.gov.in/**