COVID-19 Data Analysis and Predictive Modeling

PREDICTION ANALYSIS

A Statistical Approach to Predicting Death Rates and Recovery

TABLE OF CONTENTS

1 Introduction

2 Data Overview and Methodology

3 Data Cleaning and Preprocessing

4 Exploratory Data Analysis

5 Correlation Analysis

6 Statistical Tests & Outlier Detection

7 Regression Analysis

8 Recommendations and conclusion

1

Introduction

Introduction

COVID-19 Impact in India:

- India faced one of the world's largest outbreaks.
- Over 30 million cases and 400,000 deaths recorded.

Objective:

- To explore the factors affecting COVID-19 deaths in India.
- Analyze trends, regional differences, and the Case Fatality Rate (CFR).

Approach:

- Data preprocessing and statistical analysis.
- Focus on identifying patterns and relationships.

2

Data overview and Methodology

Data Overview and Methodology

Data sources

COVID-19 Cases(22-04-2021).csv

Open with Microsoft Excel

S. No.	Date	Region	Confirmed Cases	Active Cases	Cured/Discharged	Death
1	12/03/2020	India	74	71	3	0
2	13/03/2020	India	75	71	3	1
3	14/03/2020	India	84	72	10	2
4	15/03/2020	India	107	95	10	2
5	16/03/2020	India	114	99	13	2

COVID-19 India dataset

confirmed cases, deaths, recoveries

Data Overview and Methodology

Data sources

covid_19_india.csv

Open with Microsoft Excel

Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
1	2020-01-30	6:00 PM	Kerala	1	0	0	0	1
2	2020-01-31	6:00 PM	Kerala	1	0	0	0	1
3	2020-02-01	6:00 PM	Kerala	2	0	0	0	2
4	2020-02-02	6:00 PM	Kerala	3	0	0	0	3
5	2020-02-03	6:00 PM	Kerala	3	0	0	0	3

COVID-19 Cases

confirmed cases, active cases,
recovered, deaths by region

Data Overview and Methodology

Methodology

Data Cleaning:

- Imputation (missRanger)
- Date format conversion
- Merging datasets

Feature Engineering:

- Case Fatality Rate (CFR)
- Weekly data extraction
- Outlier removal

Statistical Analysis:

- Correlation analysis
- Kruskal-Wallis test (regional comparison)

Regression Analysis:

- Week 5: Multivariate regression to predict deaths.
- Week 6: Auto-regression with Week 5 deaths.

3

Data cleaning and Preprocessing

Data Cleaning and Preprocessing

Handling Missing Data:

Imputation using missRanger to fill missing values for key variables like Deaths, Active Cases, and Cured.

```
Step 2: Handling Missing Data
> covid_india <- missRanger(covid_india, pmm.k = 5)
Missing value imputation by random forests

Nothing to impute!
> covid_cases <- missRanger(covid_cases, pmm.k = 5)
Missing value imputation by random forests

Nothing to impute!
> cat("Missing data imputation completed for both datasets.\n")
Missing data imputation completed for both datasets.
>
```

Both datasets had no missing values after imputation, meaning no further imputation was needed.

Data Cleaning and Preprocessing

Merging Datasets:

- Converted the Date column in both datasets to Date format for consistency.
- Merged the two datasets by the common Date column.

```
> # Step 3: Convert Date Column and Merge Datasets
> covid_india$Date <- as.Date(covid_india$Date, format = "%Y-%m-%d")
> covid_cases$Date <- as.Date(covid_cases$Date, format = "%d/%m/%Y")
>
> merged_data <- merge(
+   covid_india[, c("Date", "State.UnionTerritory", "Cured", "Deaths", "Confirmed")],
+   covid_cases[, c("Date", "Region", "Confirmed.Cases", "Active.Cases", "Cured.Discharged", "Death")],
+   by = "Date", all = TRUE
+ )
>
> cat("Step 3: Merged Datasets by Date\n")
Step 3: Merged Datasets by Date
> cat("Dataset structure after merge:\n")
Dataset structure after merge:
> print(str(merged_data))
'data.frame':   509288 obs. of  10 variables:
 $ Date                : Date, format: "2020-01-30" "2020-01-31" "2020-02-01" ...
 $ State.UnionTerritory: chr  "Kerala" "Kerala" "Kerala" "Kerala" ...
 $ 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           : num  1 1 2 3 3 3 3 3 3 3 ...
 $ Region              : chr  NA NA NA NA ...
 $ Confirmed.Cases     : num  NA NA NA NA NA NA NA NA NA NA ...
 $ Active.Cases        : int  NA NA NA NA NA NA NA NA NA NA ...
 $ Cured.Discharged    : int  NA NA NA NA NA NA NA NA NA NA ...
 $ Death               : int  NA NA NA NA NA NA NA NA NA NA ...
NULL
>
```

Data Cleaning and Preprocessing

Week Calculation & CFR (Case Fatality Rate) Calculation:

- Extracted the week number from the Date column using the format() function.
- CFR Formula: The ratio of deaths to confirmed cases, expressed as a percentage

```
> # Step 4: Add 'Week' Column and Calculate CFR (Case Fatality Rate)
> merged_data <- merged_data %>%
+   mutate(Week = as.numeric(format(Date, "%U"))) %>%
+   mutate(CFR = ifelse(Confirmed > 0, (Deaths / Confirmed) * 100, 0))
>
> cat("\nStep 4: Case Fatality Rate (CFR) Added\n")

Step 4: Case Fatality Rate (CFR) Added
> print(head(merged_data[, c("Date", "Deaths", "Confirmed", "CFR")]))
        Date Deaths Confirmed CFR
1 2020-01-30      0         1   0
2 2020-01-31      0         1   0
3 2020-02-01      0         2   0
4 2020-02-02      0         3   0
5 2020-02-03      0         3   0
6 2020-02-04      0         3   0
>
```

Data Cleaning and Preprocessing

Removing Zero Variance Columns:

```
> # Step 5: Check and Remove Zero Variance Columns
> covid_india_numeric <- covid_india %>% select_if(is.numeric)
> covid_cases_numeric <- covid_cases %>% select_if(is.numeric)
>
> zero_variance_columns_india <- sapply(covid_india_numeric, function(x) sd(x, na.rm = TRUE) == 0)
> zero_variance_columns_cases <- sapply(covid_cases_numeric, function(x) sd(x, na.rm = TRUE) == 0)
>
> cat("\nStep 5: Columns with Zero Variance\n")

Step 5: Columns with Zero Variance
> cat("Zero variance columns in covid_india: ", names(zero_variance_columns_india[zero_variance_columns_i
ndia]), "\n")
Zero variance columns in covid_india:
> cat("Zero variance columns in covid_cases: ", names(zero_variance_columns_cases[zero_variance_columns_c
ases]), "\n")
Zero variance columns in covid_cases:
>
> covid_india_no_zero_variance <- covid_india_numeric[, !zero_variance_columns_india]
> covid_cases_no_zero_variance <- covid_cases_numeric[, !zero_variance_columns_cases]
>
```

Data Cleaning and Preprocessing

Week-Specific Data Extraction and Alignment:

- Extracted data for specific weeks (Week 4, Week 5, and Week 6) for focused analysis.
- Aligned data across weeks to ensure consistency for comparative analysis.

```
> # Step 8: Extract Week-Specific Data (For analysis)
> extract_week_data <- function(data, week_num) {
+   week_data <- filter(data, Week == week_num) %>% filter(complete.cases(.))
+   return(week_data)
+ }
>
> week_4_data <- extract_week_data(merged_data, 4)
> week_5_data <- extract_week_data(merged_data, 5)
> week_6_data <- extract_week_data(merged_data, 6)
>
> # Align data across weeks
> min_rows <- min(nrow(week_4_data), nrow(week_5_data), nrow(week_6_data))
> week_4_data <- week_4_data[1:min_rows, ]
> week_5_data <- week_5_data[1:min_rows, ]
> week_6_data <- week_6_data[1:min_rows, ]
```

4

Exploratory Data Analysis

Exploratory Data Analysis

Key Statistical Insights:

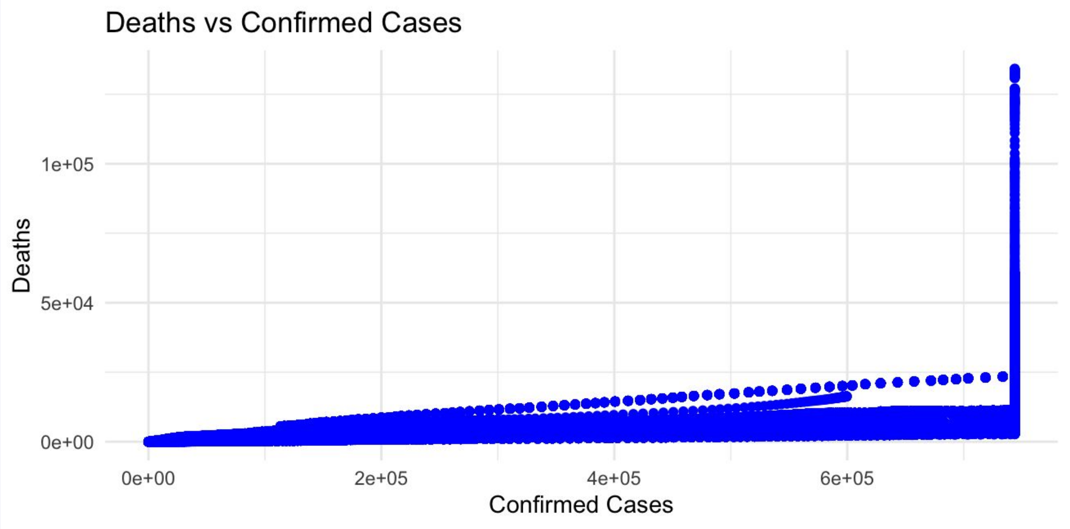
Summary statistics for Deaths and Confirmed cases to understand central tendency and range.

```
> # Perform simple descriptive analysis on key variables
> summary(merged_data$Deaths)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0      13     348    2541    1935  134201
> summary(merged_data$Confirmed)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0    2954   23902  142586  210268  743809
```

Exploratory Data Analysis

Scatter Plot: Deaths vs Confirmed Cases

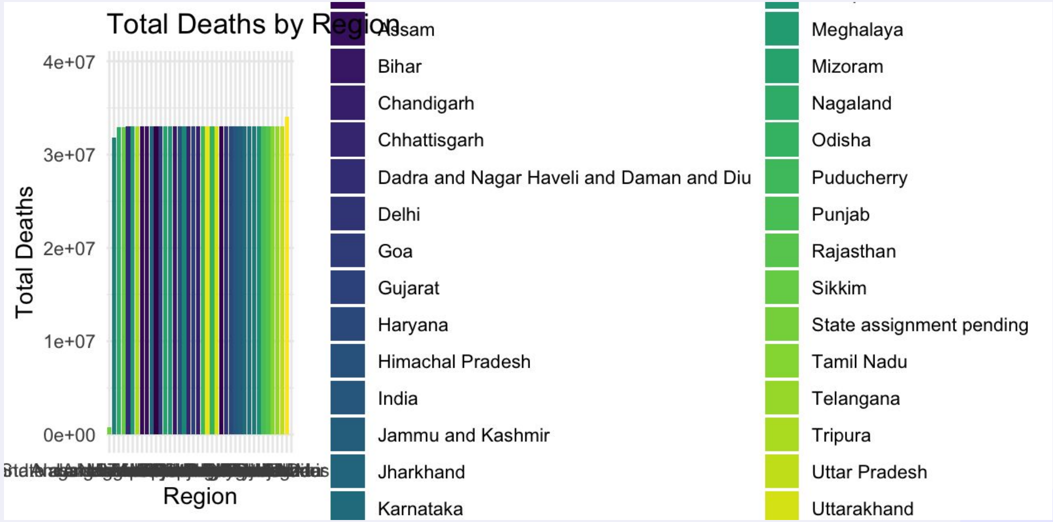
- Relationship Between Deaths and Confirmed Cases
- **Interpretation:** A positive correlation suggests that more confirmed cases generally lead to more deaths.



Exploratory Data Analysis

Bar Plot: Total Deaths by Region

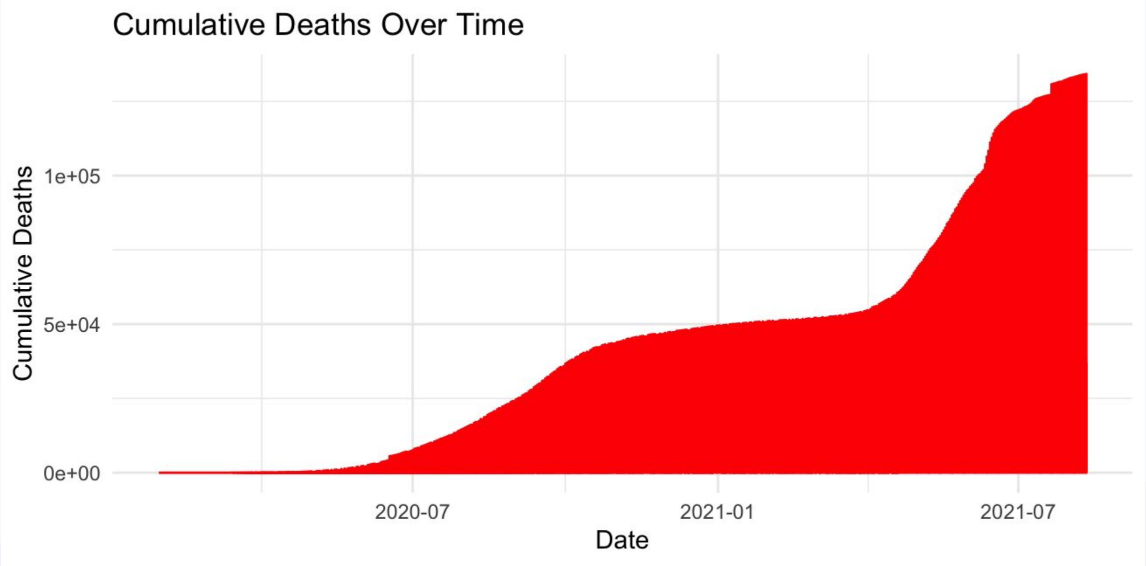
- Visualizes the total number of deaths across regions.
- **Interpretation:** Some regions may have significantly higher deaths, highlighting areas with the most severe impact.



Exploratory Data Analysis

Line Plot: Cumulative Deaths Over Time

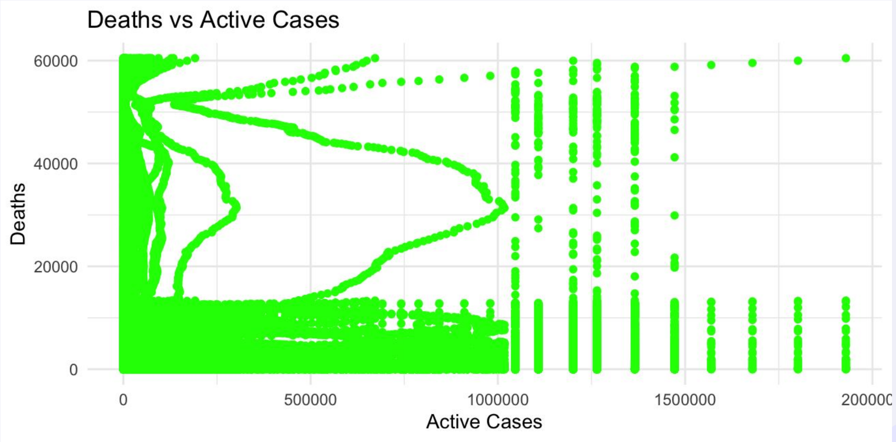
- Shows the accumulation of deaths over time.
- **Interpretation:** Observing this trend will highlight significant spikes or declines during specific periods.



Exploratory Data Analysis

Scatter Plot: Deaths vs Active Cases

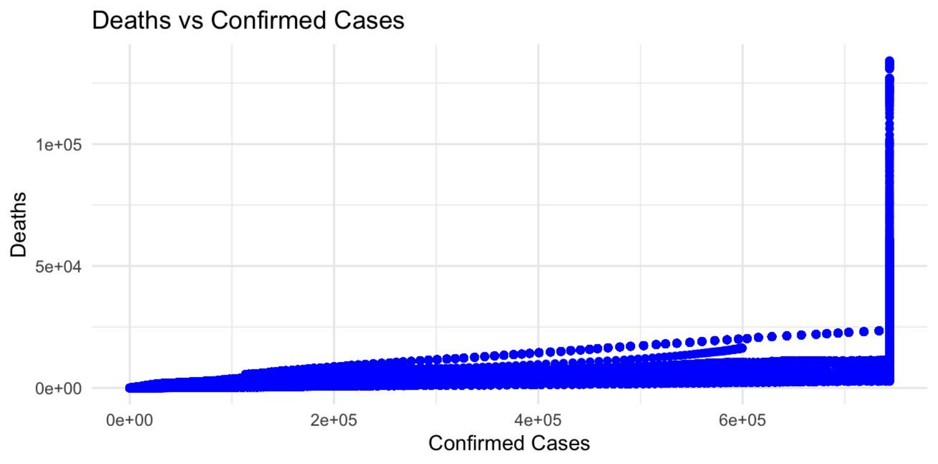
- Explore the correlation between active cases and deaths, helping understand the impact of ongoing infections.
- **Interpretation:** A positive correlation is observed, indicating that higher active cases tend to result in more deaths.



Exploratory Data Analysis

Scatter Plot: Deaths vs Active Cases

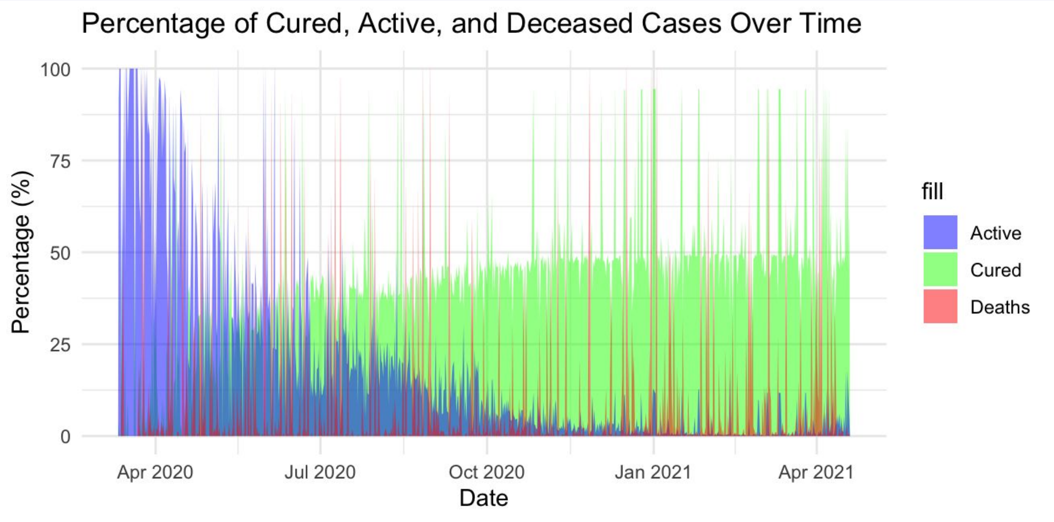
- Visualize the relationship between confirmed cases and the number of deaths.
- **Interpretation:** A positive correlation between confirmed cases and deaths. Higher confirmed cases tend to correlate with higher death rates.



Exploratory Data Analysis

Cured, Active and Deceased Cases Over Time

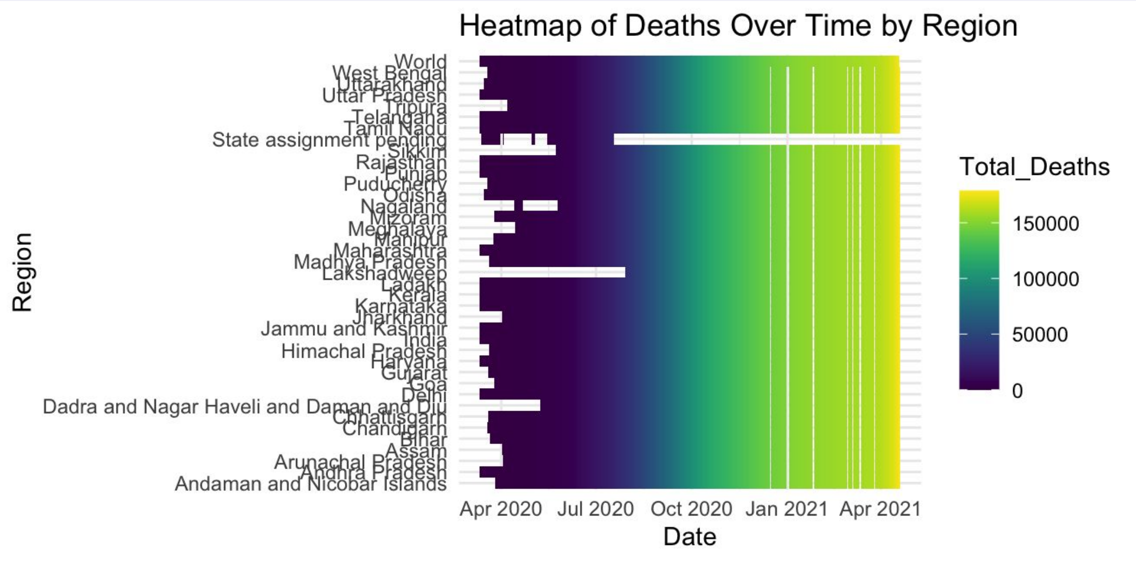
- Visualizes how the percentages of cured, active, and deceased cases evolved over time.
- **Interpretation:** A positive correlation between confirmed cases and deaths. Higher confirmed cases tend to correlate with higher death rates.



Exploratory Data Analysis

Heatmap: Deaths Over Time By Region:

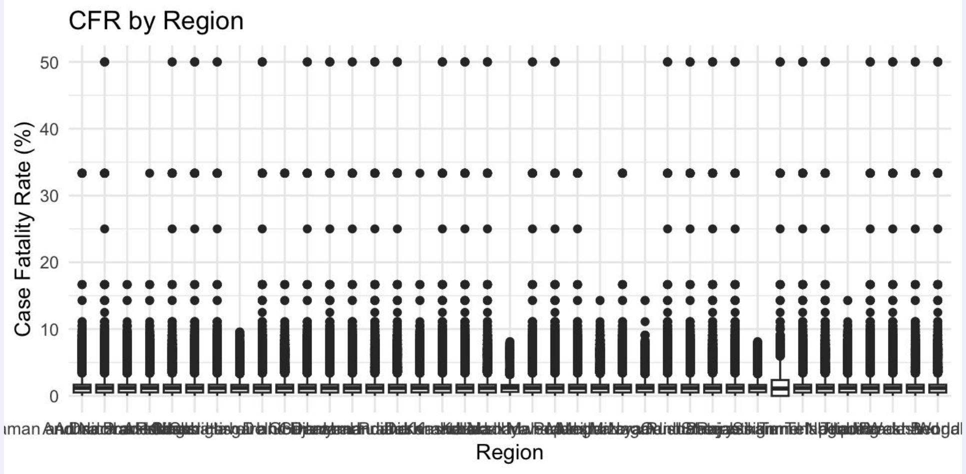
- Highlights patterns of death concentration across regions and times.



Exploratory Data Analysis

CFR By Region:

- A boxplot to analyze the variation of case fatality rates (CFR) across different regions.



5

Correlation Analysis

Correlation Analysis - Covid India

Objective:

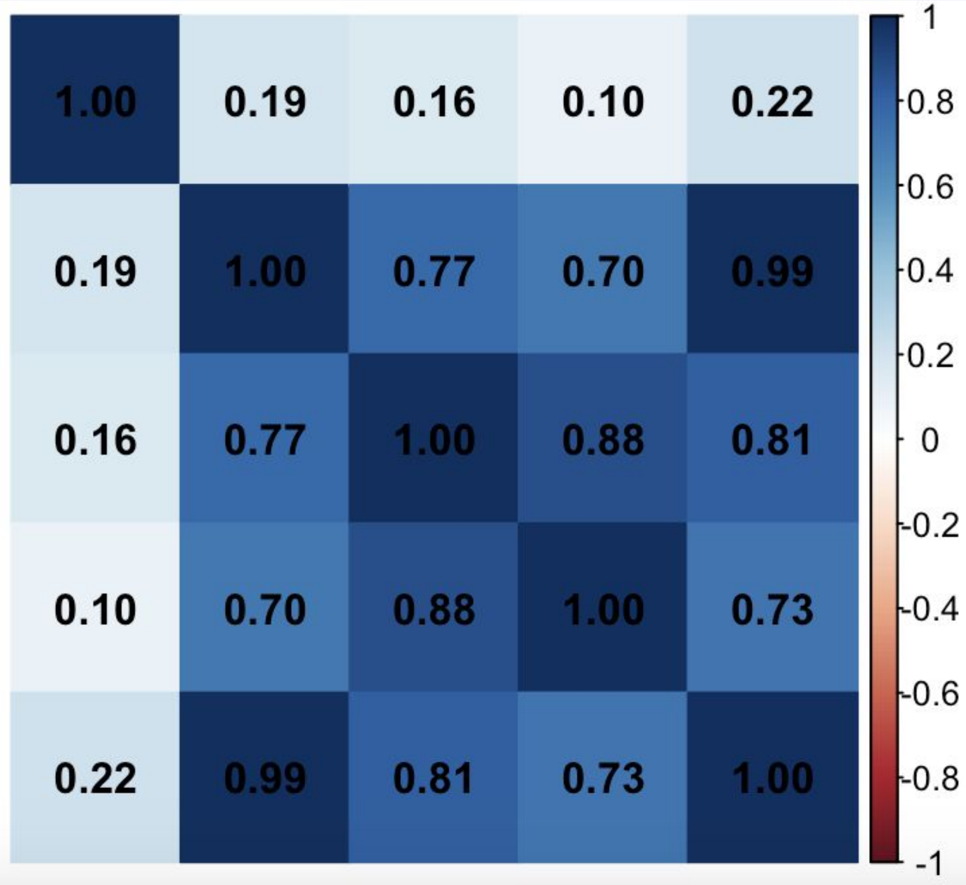
- Analyze relationships among key variables such as confirmed cases, deaths, and cured cases.

Key Findings:

- Covid India Dataset:
    - Strong positive correlation between active cases and confirmed cases, as expected.
    - Notable correlation between deaths and confirmed cases, emphasizing the proportionality between these metrics.

```
> cat("Correlation Matrix for covid_india:\n")
Correlation Matrix for covid_india:
> print(correlation_matrix_india)
                Sno     Cured    Deaths Confirmed
Sno       1.0000000 0.4084822 0.3017418 0.5200524
Cured     0.4084822 1.0000000 0.9175294 0.7349504
Deaths    0.3017418 0.9175294 1.0000000 0.5963692
Confirmed 0.5200524 0.7349504 0.5963692 1.0000000
> 
```

Correlation Analysis - Covid India



Correlation Analysis - Covid Cases

Objective:

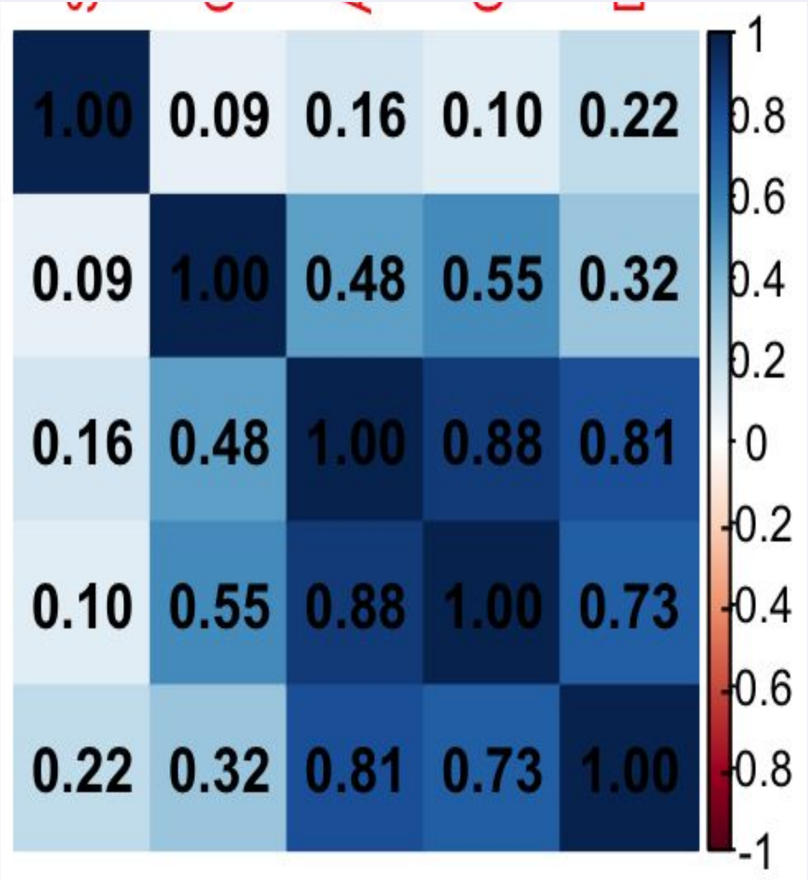
- Explore the relationship between confirmed cases, active cases, deaths, and other metrics.

Key Findings:

- Covid Cases Dataset:
  - High positive correlation between confirmed cases and deaths.
  - Moderate correlation between cured cases and confirmed cases, suggesting effective recovery management in some regions.

```
Correlation Matrix for covid_cases:
> print(correlation_matrix_cases)
                      S..No. Confirmed.Cases Active.Cases Cured.Discharged     Death
S..No.            1.00000000      0.08869163    0.1597615        0.1007460 0.2175650
Confirmed.Cases   0.08869163      1.00000000    0.4818943        0.5521995 0.3234832
Active.Cases      0.15976146      0.48189432    1.0000000        0.8803811 0.8054788
Cured.Discharged  0.10074605      0.55219951    0.8803811        1.0000000 0.7327062
Death             0.21756501      0.32348317    0.8054788        0.7327062 1.0000000
> 
```

Correlation Analysis - Covid Cases



6

Statistical Tests & Outlier Detection

Outlier Detection

Outlier Identification and Capping:

Method:

- Interquartile Range (IQR) technique used to identify and cap extreme values.
- Extreme values were adjusted within acceptable limits for key variables, such as Confirmed Cases and Deaths.

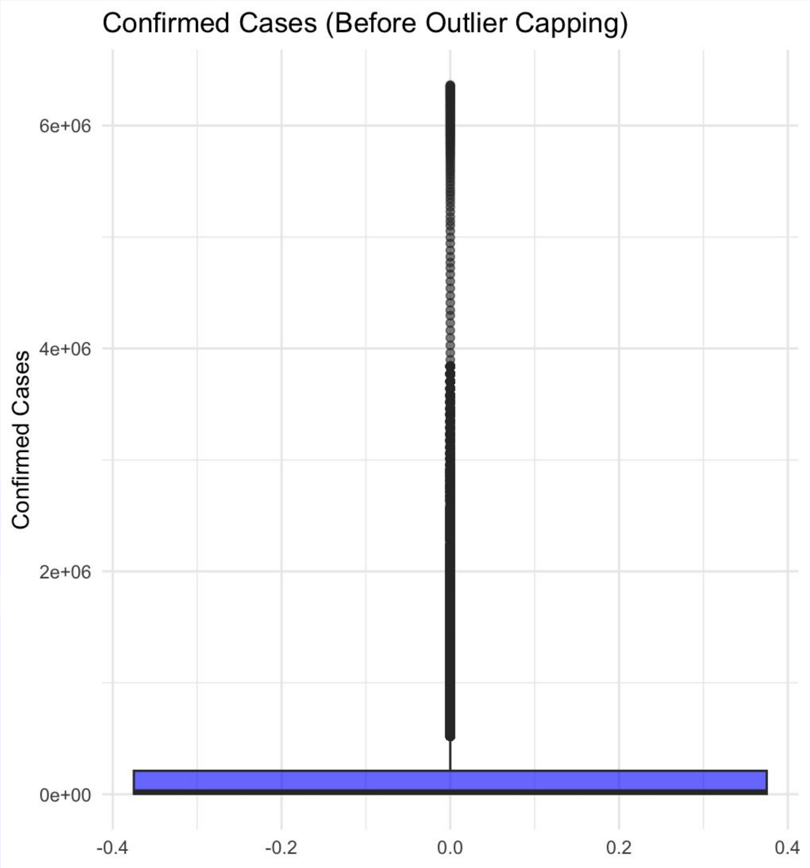
Impact:

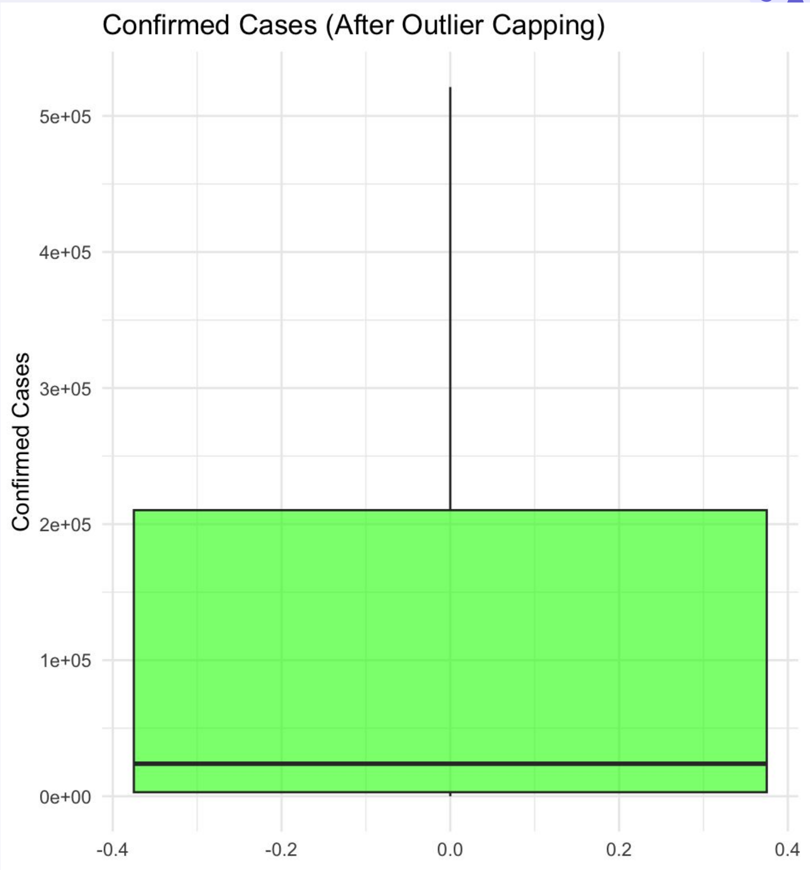
- Outlier handling improved data consistency for meaningful analysis.

```
> # Step 7: Handle Outliers
> cap_outliers <- function(x) {
+   Q1 <- quantile(x, 0.25, na.rm = TRUE)
+   Q3 <- quantile(x, 0.75, na.rm = TRUE)
+   IQR <- Q3 - Q1
+   upper_limit <- Q3 + 1.5 * IQR
+   lower_limit <- Q1 - 1.5 * IQR
+   pmin(pmax(x, lower_limit), upper_limit)
+ }
>
> covid_india$Confirmed <- cap_outliers(covid_india$Confirmed)
> covid_cases$Confirmed.Cases <- cap_outliers(covid_cases$Confirmed.Cases)
>
> cat("\nStep 7: Outliers Capped\n")

Step 7: Outliers Capped
```

Outlier Detection





Statistical Tests

Kruskal-Wallis Test (by Region):

- **Purpose:** Assess whether there are significant differences in Deaths across different regions.
- **Result:** Sugget significant differences in death counts across regions, indicating regional disparities.

```
> # Step 10: Kruskal-Wallis Test (Analyze by Region)
> cat("\nStep 10: Kruskal-Wallis Test by Region\n")

Step 10: Kruskal-Wallis Test by Region
> kruskal_test <- kruskal.test(Deaths ~ Region, data = merged_data)
> cat("Kruskal-Wallis Test Result: \n")
Kruskal-Wallis Test Result:
> print(kruskal_test)

        Kruskal-Wallis rank sum test

data:  Deaths by Region
Kruskal-Wallis chi-squared = 4630.5, df = 38, p-value < 2.2e-16
```

7

Regression Analysis

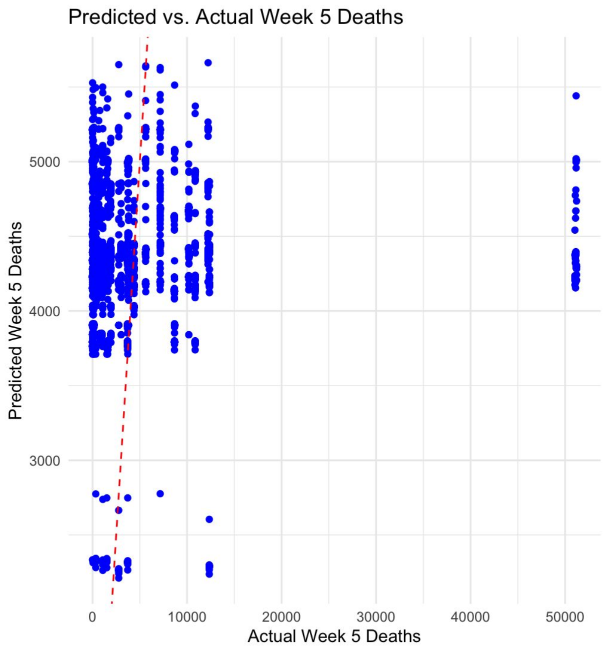
Data Overview and Methodology

Week 5: Multivariate Linear Regression

Variables Used: Confirmed Cases, Active Cases, Cured, CFR.

Key Insights:

- Predicted vs. actual Week 5 deaths comparison.
- Significant underestimation for values >30,000.
- Model needs refinement for extreme cases.



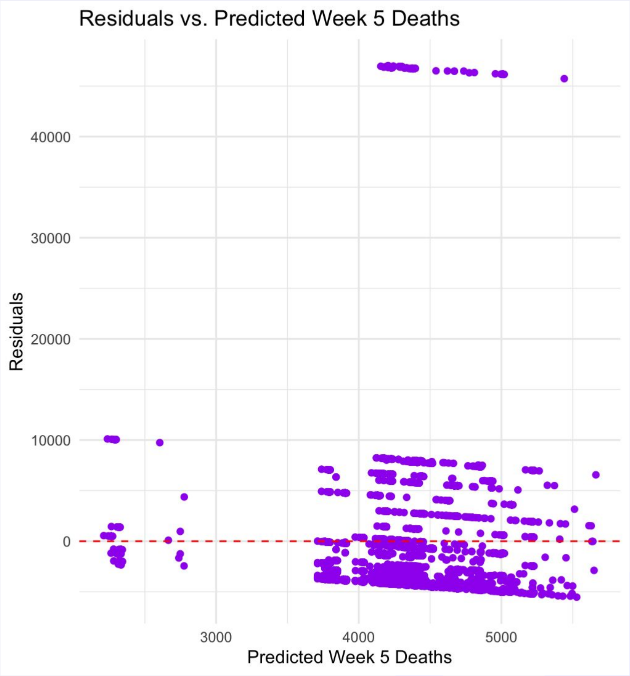
Data Overview and Methodology

Week 5: Multivariate Linear Regression

Variables Used: Confirmed Cases, Active Cases, Cured, CFR.

Key Insights:

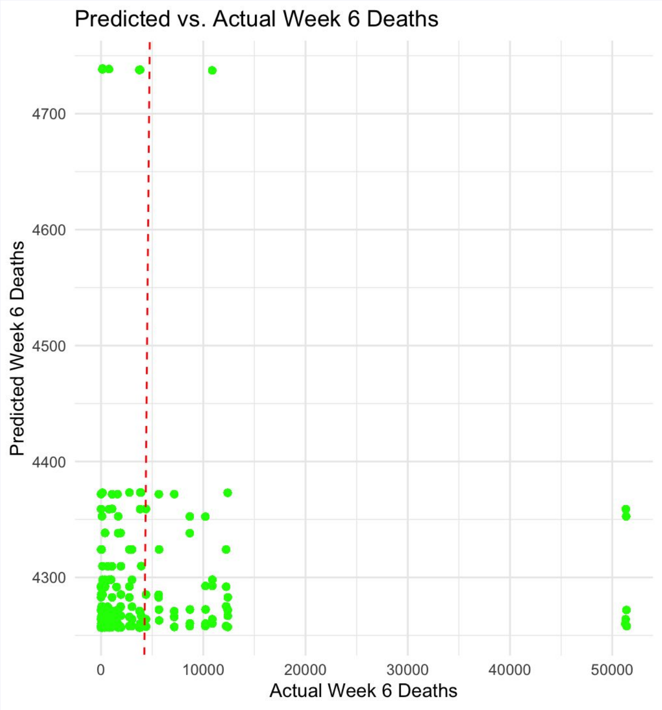
- Residuals vs. predicted Week 5 deaths.
- Most errors near 0 for predictions around 4,000–5,000.
- Large errors (>10,000) show underprediction.
- Model struggles with extreme cases



Data Overview and Methodology

Week 6: Auto-Regression

- Predicted vs. actual Week 6 deaths.
- Significant underestimation for actual values >10,000.
- Model lacks sensitivity to extremes.



8

Recommendations and conclusion

Recommendations

Model Improvement

- Add predictors like population density or healthcare infrastructure.
- Test non-linear models (e.g., GAM, polynomial regression).

Improve Data Quality:

Regularly update and verify case and death reports to enhance model accuracy. Standardize data collection and reporting methods across all regions.

Recommendations

Model Comparison:

Compare the performance of auto-regression with alternative techniques like Random Forest or XGBoost to validate the predictive reliability for Week 6.

Scenario Testing:

Simulate the impact of varying key predictors (e.g., CFR or Active Cases) to understand sensitivity and improve reliability of predictions..

Conclusion

The analysis successfully predicted COVID-19 death trends using regression models, highlighting key drivers and regional variations. While effective overall, further refinement is needed to improve accuracy for high-death regions. These insights provide a solid basis for informed, data-driven decisions.

THANKS FOR YOUR ATTENTION!