Assignment 1 - Readme

Table of Contents

Overview

1. This project involves the loading, preprocessing, and visualizing of the Covid-19 data from a JSON type file.
2. The name of the JSON file is : `states_daily.json`
3. The code was written in python language (Python 3.X).
4. The IDE used was Jupyter Notebook.
5. The JSON file contains the following data -
    - The daily counts of covid cases for every state in India.
    - The date in the format - `"dd-Month-yy"`
    - Another date in the format - `"yyyy-mm-dd"`
    - Also contains three status of people -
        - **Confirmed**
        - **Recovered**
        - **Deceased**.
    - Each state is represented by a two-letter code.
    - The total number of cases is represented by `'tt'`

Methodology

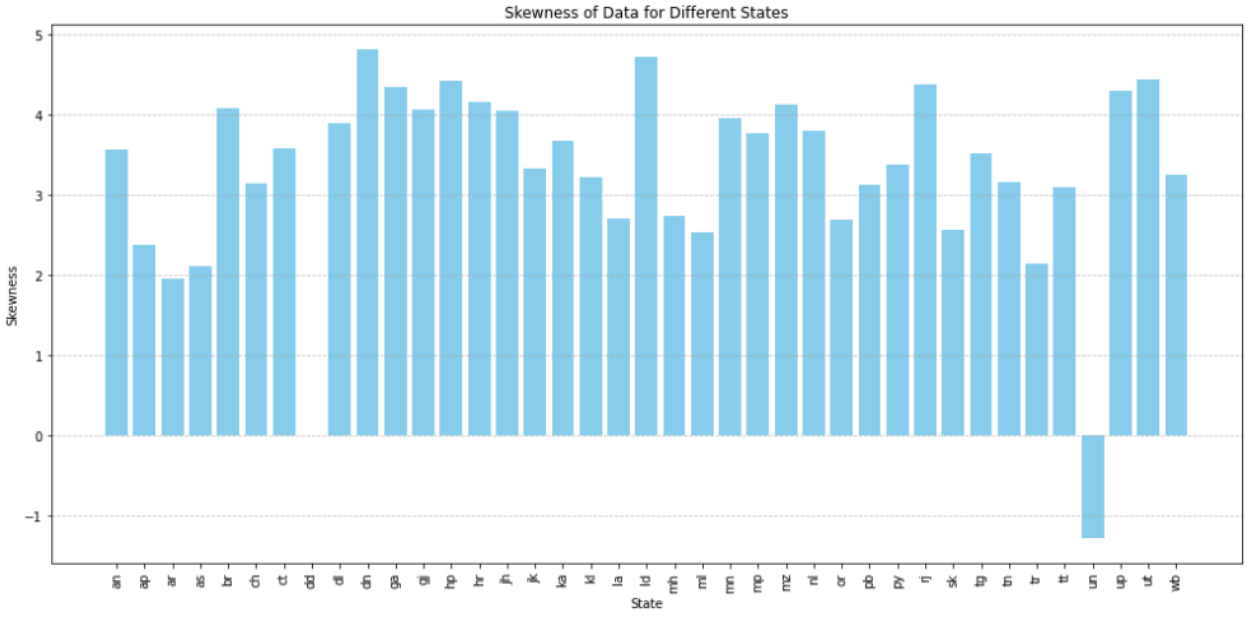
1. **Loading the data**:
    - Used json library to load the data from the JSON file.
    - Used pandas library to convert the data into a data-frame using the json_normalize method.
2. **Data Exploration**:
    - Inspected the data present in the data-frame using functions like describe(), info(), shape, etc.
3. **Data Cleaning**:
    - Converted the date fields to their proper format using the datetime function.
    - Checked and handled any missing values if present.
4. **Data Analysis**:
    - Allows analyzer to input specific dates to get the information of the data in that time period.
    - Performed analysis based on the questions asked in the ***Data Manipulation*** section.
5. **Data Visualization**:
    - Plotted the trends asked and understood the trend types.
6. **Linear Regression**:
    - Implemented a linear regression function to understand the data and predict future data.

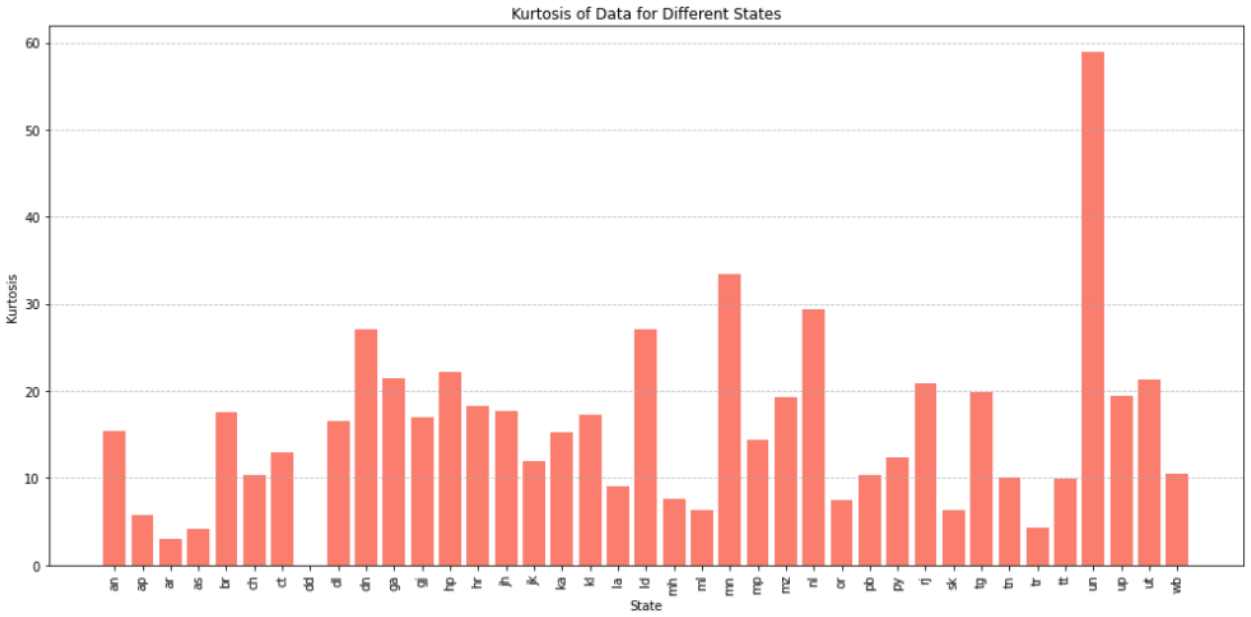
Numeric Analysis:

- **Mean**:
    - The *mean* represents the average value of the dataset. It provides a central point around which the values are distributed.
- **Median**:
    - The *median* is the middle value of the dataset when ordered. It divides the dataset into two halves, providing a measure of central tendency that is not affected by extreme values.
- **Standard Deviation**:
    - The *standard deviation* measures the average amount of variability or dispersion from the mean.
    - A higher standard deviation indicates that data points are more spread out from the mean, while a lower standard deviation indicates they are closer to the mean.
- **Variance**:
    - *Variance* measures the average squared deviations from the mean, providing an indication of the spread of data points. It is the square of the standard deviation.
- **Minimum**:

  - The *minimum* value is the lowest value in the dataset. It provides the lower boundary of the data range.
- **Maximum**:
  - The *maximum* value is the highest value in the dataset. It provides the upper boundary of the data range.
- **Range**:
  - The *range* is the difference between the maximum and minimum values. It measures the spread or extent of the data.
- **Skewness**:
  - *Skewness* measures the asymmetry of the distribution of data around its mean. It indicates whether the data is skewed to the left or right.
  - Positive values indicate a rightward skew.
  - Negative values indicate a leftward skew.
  - Values close to 0 indicate a symmetric distribution.
- **Kurtosis**:
  - *Kurtosis* measures the "tailedness" of the distribution. It indicates whether the data has heavy or light tails compared to a normal distribution.
  - Values greater than 0 (excess kurtosis) indicate heavy tails.
  - Values less than 0 indicate light tails.
  - A value close to 0 indicates normal distribution tails.

Graph Analysis:





Data Manipulation

Question 1:

- Count the total number of “Confirmed”, “Recovered” and “Deceased” from 14-Mar-2020 to 30- Sept-2020 and report the numbers.
- Steps taken:
  - Option given to users if they wish to enter a date,
    - if, yes then user input is considered
    - else, default date will be used.
  - Date is converted to the correct format using the datetime.date() function and to_datetime() function.
  - Created a new data-frame with the values from the time period given.
  - Generated the sum for each of the following status -
    - Confirmed
    - Recovered
    - Deceased
      - and then called the `tt` column and got their sum.
  - Printed the sums by creating another dataframe.

Question 2:

- Count the total number of “Confirmed”, “Recovered” and “Deceased” from 14-Mar-2020 to 05- Sept-2020 for the state of Delhi (dl).
- Steps taken:
  - Option given to users if they wish to enter a date,

        - if, yes then user input is considered
        - else, default date will be used.
    - Date is converted to the correct format using the datetime.date() function and to_datetime() function.
    - Created a new data-frame with the values from the time period given.
    - Generated the sum for each of the following status -
      - Confirmed
      - Recovered
      - Deceased
        - and then called the `dl` column and got its sum.
    - Printed the sums by creating another dataframe.

Question 3:

- Report total count of “Confirmed”, “Recovered” and “Deceased” count from states Delhi + Karnataka (ka) (Sum of both states count) from 14-Mar-2020 to 05-Sept-2020.
- Steps taken:
    - Option given to users if they wish to enter a date,
        - if, yes then user input is considered
        - else, default date will be used.
    - Date is converted to the correct format using the datetime.date() function and to_datetime() function.
    - Created a new data-frame with the values from the time period given.
    - Generated the sum for each of the following status -
      - Confirmed
      - Recovered
      - Deceased
        - first called the `dl` column and got its sum.
        - then called the `ka` column and got its sum.
    - Added the sums of both columns together.
    - Printed the sums by creating another dataframe.

Question 4:

- Report the highest affected state in terms of “Confirmed”, “Recovered” and “Deceased” with the count till 05-Sept-2020 from 14-Mar-2020.
- Steps taken:
    - *Set Dates:*
        - Convert the start date (14-Mar-2020) and end date (05-Sept-2020) into a format suitable for filtering the dataset.

- *Filter Data*:
  - Extract data from the main dataset for the specified date range.
- *Sum Cases:*
  - Calculate the total number of cases (Confirmed, Recovered, and Deceased) for each state by summing the daily numbers within the date range.
- *Find Highest Values:*
  - Use functions to find the state with the highest total cases for each status:
    - Confirmed:
      - Find the state with the maximum number of confirmed cases.
    - Recovered:
      - Find the state with the maximum number of recovered cases.
    - Deceased:
      - Find the state with the maximum number of deceased cases.
- *Show Results:*
  - Create a table showing the states with the highest number of cases for each status and print it out.

Question 5:

- Report the lowest affected state in terms of “Confirmed”, “Recovered” and “Deceased” with the count till 05-Sept-2020 from 14-Mar-2020.
- Steps taken:
  - *Set Dates*:
    - Convert the start date (14-Mar-2020) and end date (05-Sept-2020) into a format suitable for filtering.
  - *Filter Data:*
    - Extract data from the main dataset for the specified date range.
  - *Sum Cases:*
    - Calculate the total number of cases (Confirmed, Recovered, and Deceased) for each state by summing the daily numbers within the date range.
  - *Identify Lowest Values:*
    - Use functions to find the state with the lowest total cases for each status:
      - Confirmed:
        - Find the state with the minimum number of confirmed cases.
      - Recovered:
        - Find the state with the minimum number of recovered cases.
      - Deceased:
        - Find the state with the minimum number of deceased cases.
  - *Display Results:*

        - Create a table showing the states with the lowest number of cases for each status and print it out.

Question 6:

- Find the day and count with the highest spike in a day in the number of cases for the state Delhi for “Confirmed”, “Recovered” and “Deceased” between dates 14-Mar-2020 and 05-Sept-2020.
- Steps taken:
    - *Set Dates:*
        - Convert the start date (14-Mar-2020) and end date (05-Sept-2020) into a format suitable for filtering.
    - *Filter Data:*
        - Extract data specific to the state of Delhi for the specified date range.
    - *Organize Data:*
        - Group the data by date and status, then pivot the table to get separate columns for Confirmed, Recovered, and Deceased cases.
    - *Calculate Daily Changes:*
        - Determine the daily changes in cases by computing the difference between consecutive days.
    - *Identify Highest Spikes:*
        - Find the day with the largest increase for each status (Confirmed, Recovered, Deceased) and get the corresponding count for that day.
    - *Display Results:*
        - Print the day and the number of cases where the highest spikes occurred for each status.

Question 7:

- Report active cases (Assume active = Confirmed - (Recovered + Deceased)) state wise for all states separately on date 05-Sept-2020 (This date only) starting from 14-March-2020.
- Steps taken:
    - *Set Date:*
        - Convert the specific date (05-Sept-2020) into a format suitable for filtering.
    - *Filter Data:*
        - Extract data for the specific date (05-Sept-2020) from the main dataset.
    - *Separate by Status:*
        - Split the data into categories based on status (Confirmed, Recovered, Deceased).

- *Calculate Active Cases*:
  - For each state, calculate the number of active cases using the formula:
    - Active = Confirmed - (Recovered + Deceased).
  - Ensure that if the calculation results in a negative number, it is set to zero.
- *Create DataFrame*:
  - Organize the results into a DataFrame showing active cases for each state.
- *Display Results*:
  - Print the DataFrame to show the active cases for each state on the specified date.

Plotting

Question 1:

- Plot the area trend line for total “Confirmed”, “Recovered” and “Deceased” cases from 14-Mar- 2020 to 05-Sept-2010.
- Steps taken:
  - *Set Dates*:
    - Convert the start date (14-Mar-2020) and end date (05-Sept-2020) into a format suitable for filtering.
  - *Filter Data*:
    - Extract the data for all states within the specified date range.
  - *Sum Cases*:
    - Aggregate the data to get the total number of Confirmed, Recovered, and Deceased cases for each date.
  - *Plot Data*:
    - Create a plot with area trend lines to visualize the total cases for Confirmed, Recovered, and Deceased over time.
  - *Show Trends*:
    - The plot will display how the totals for each type of case change from March to September 2020.

Question 2:

- Plot the area trend line for total “Confirmed”, “Recovered” and “Deceased” cases for the state Delhi (dl) from 14-Mar-2020 to 05-Sept-2020.
- Steps taken:
  - *Set Dates*:

        - Convert the start date (14-Mar-2020) and end date (05-Sept-2020) into a format suitable for filtering.
    - *Filter Data*:
        - Extract the data specifically for the state of Delhi within the given date range.
    - *Sum Cases*:
        - Aggregate the data to get the total number of Confirmed, Recovered, and Deceased cases for Delhi on each date.
    - *Plot Data*:
        - Create a plot with area trend lines to visualize the total cases for Confirmed, Recovered, and Deceased in Delhi over time.
    - *Show Trends*:
        - The plot will show how these case totals evolve for Delhi from March to September 2020.

Question 3:

- Plot the area trend line for active cases. Assume active = Confirmed - (Recovered + Deceased) from 14-Mar-2020 to 05-Sept-2020.
- Steps taken:
    - *Set Dates*:
        - Convert the start date (14-Mar-2020) and end date (05-Sept-2020) into a format suitable for filtering.
    - *Filter Data*:
        - Extract the data for all states within the specified date range.
    - *Calculate Active Cases*:
        - Compute the number of active cases as `Confirmed - (Recovered + Deceased)` for each date.
    - *Plot Data*:
        - Create a plot with an area trend line to visualize the number of active cases over time.
    - *Show Trends*:
        - The plot will illustrate how the number of active cases changes from March to September 2020.

Linear Regression

- Steps taken:
    - *Set Dates*:

        - Convert the start date (14-Mar-2020) and end date (05-Sept-2020) into a format suitable for filtering.
    - *Filter Data*:
        - Extract the data for the state of Delhi within the specified date range
    - *Organize Data*:
        - Group the data by date and status, then pivot the table to create separate columns for Confirmed, Recovered, and Deceased cases.
    - *Prepare Data for Regression*:
        - Convert the date into a numerical format (e.g., number of days from the start date) to use as the independent variable in the regression model.
    - *Perform Linear Regression*:
        - Confirmed Cases:
            - Apply linear regression to the Confirmed cases data to find the intercept and slope of the trend line.
        - Recovered Cases:
            - Apply linear regression to the Recovered cases data to find the intercept and slope.
        - Deceased Cases:
            - Apply linear regression to the Deceased cases data to find the intercept and slope.
    - *Report Coefficients:*
        - Print the intercept and slope coefficients for each of the three cases (Confirmed, Recovered, Deceased) to show the trend over time.

Assumptions

1. The JSON data is well-formed and doesn't have any records which are corrupted.
2. The data fields are in their proper formats
3. External knowledge regarding the dataset is required for someone to be able to understand and grasp the analysis conducted.

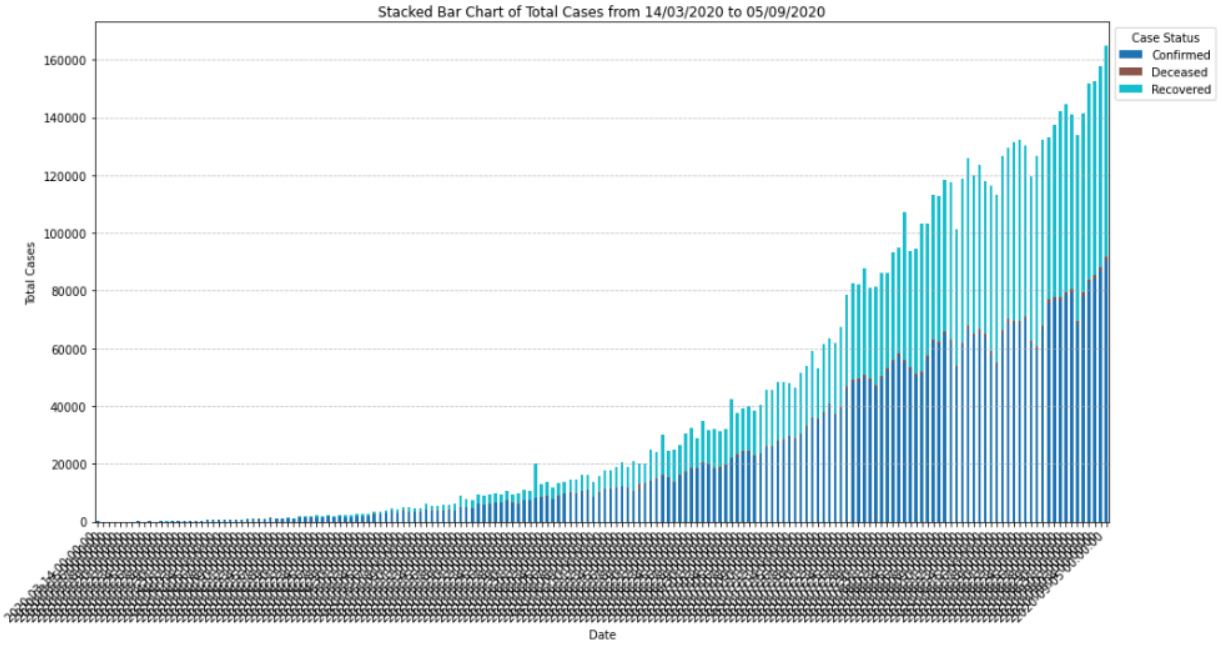
Results

1. Successfully loaded and cleaned the data.
2. Performed data analysis on the JSON file.
3. Visualized the trends of Covid-19 cases based on the questions asked.
4. Implemented a linear regression function and noted the intercept and slope coefficients

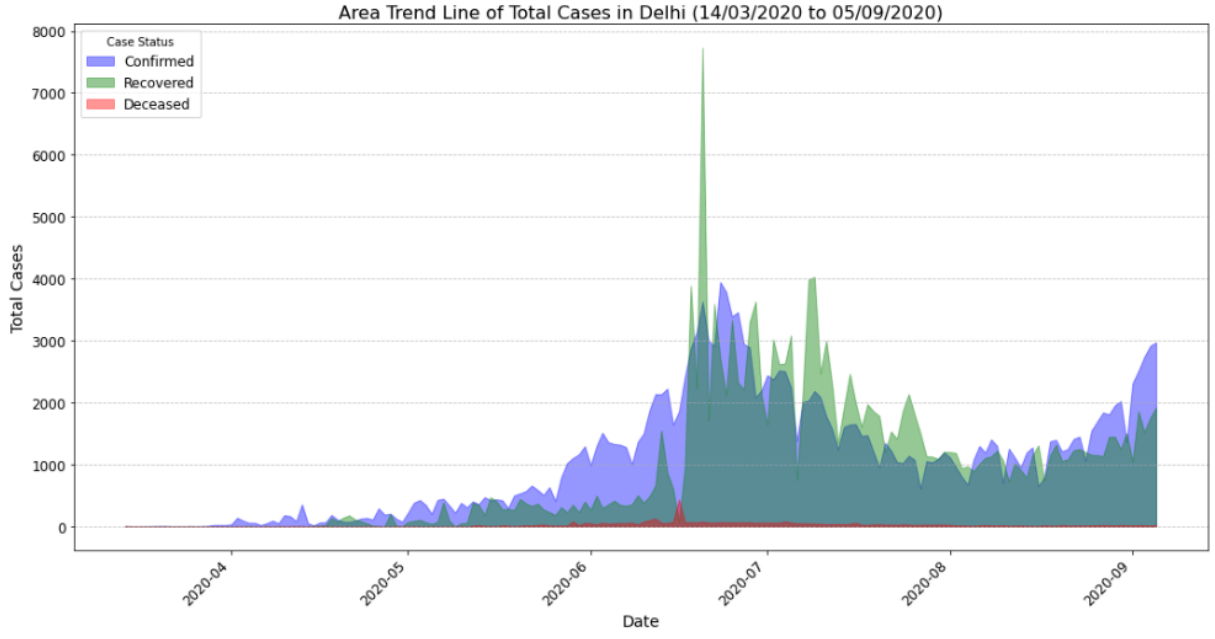
for the different cases mentioned.

Graphs

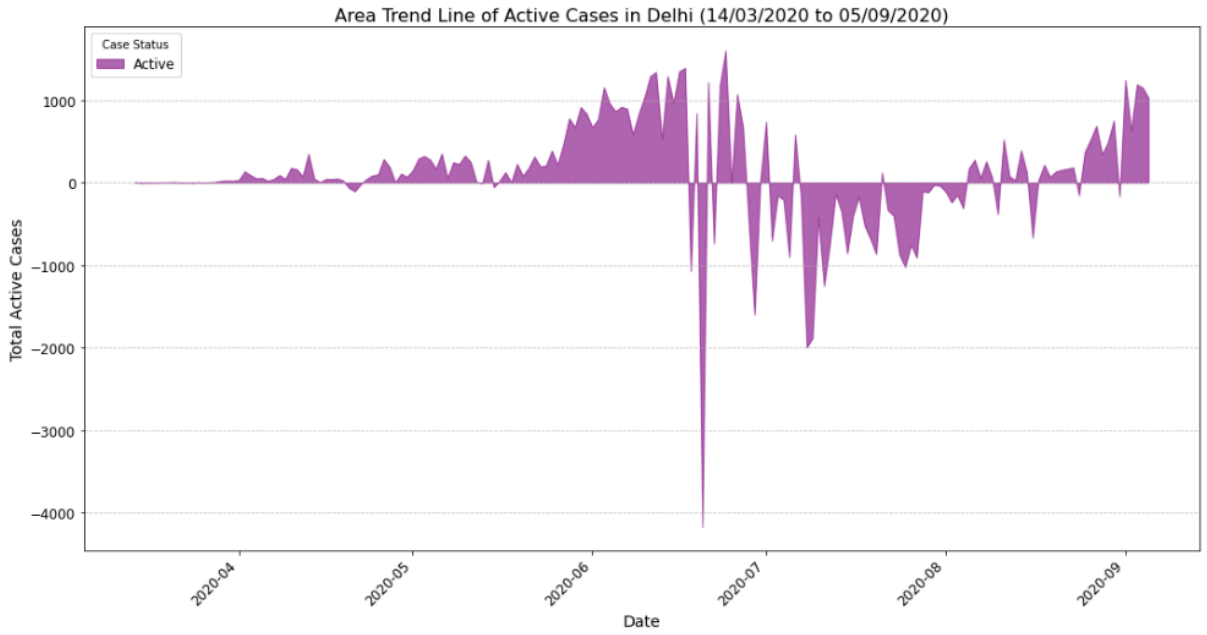
Plotting Question - 1



Plotting Question - 2

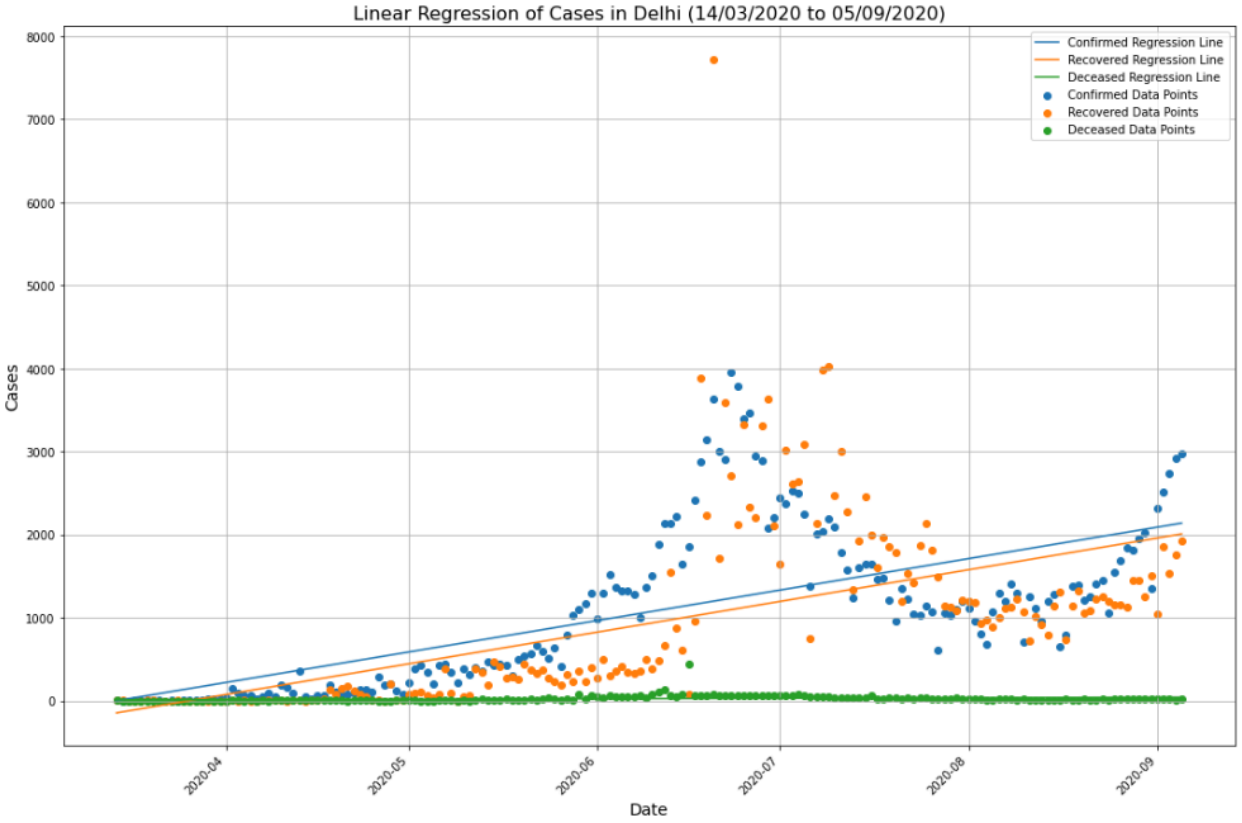


Plotting Question - 3



Linear Regression Outcomes

- **Confirmed** -
    1. *Intercept*: 0.53
    2. *Slope*: 12.21
- **Recovered** -
    1. *Intercept*: -146.14
    2. *Slope*: 12.31
- **Deceased** -
    1. *Intercept*: 9.14
    2. *Slope*: 0.19


Linear Regression of Cases in Delhi (14/03/2020 to 05/09/2020)
Confirmed Regression Line
Recovered Regression Line
Deceased Regression Line
Confirmed Data Points
Recovered Data Points
Deceased Data Points
Cases
Date
8000
7000
6000
5000
4000
3000
2000
1000
0
2020-04
2020-05
2020-06
2020-07
2020-08
2020-09