**DATASCIENCE TOOLBOX PYTHON PROGRAMMING**

**PROJECT REPORT**

(Project Semester January-April 2025**)**

**EDA ON GST**

Submitted by: Rabia Kaur Bhatia

Registration No: 12324015

Programme and Section: CSE & K23KM

Course Code : INT375

Under the Guidance of: **Anchal Kaundal sir**

**Discipline of CSE/IT**

**Lovely School of Computer Science and Engineering**

**Lovely Professional University, Phagwara**

**CERTIFICATE**

This is to certify that Rabia bearing Registration no. 12324015 has completed INT375 project titled, “EDA ON GST” under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

**Signature and Name of the Supervisor**

**Designation of the Supervisor**

**School of Computer Science and Engineering**

**Lovely Professional University**

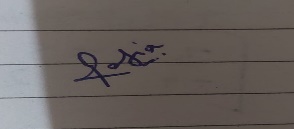
**Phagwara, Punjab.**

**Date: 10/4/2025**

**DECLARATION**

I, Rabia , student of computer science and engineering under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 10/4/2025                                                        Registration No.12324015

Name of the student :Rabia Kaur Bhatia Signature: 

**Acknowledgement**

I sincerely thank Anchal sir, my mentor, for their invaluable guidance and support throughout this project. I extend my gratitude to the faculty of the Department of Computer Science and my friends and family for their constant encouragement.

**TABLE OF CONTENTS**

| **S. No.** | **Content** | | **Page No.** |
| --- | --- | --- | --- |
| 1 | Introduction | | [Introduction](#Introduction) |
| 2 | Source of Dataset | | [Source](#Source) |
| 3 | Exploratory Data Analysis (EDA) Process | | [(EDA)](#Exploratory) |
| 4 | Analysis on Dataset | | [Analysis](#AnalysisonDataset) |
|  | 1.Summary statistics |
|  | 2.Correlation heatmaps |
|  | 3.Revenue distribution plots |
|  | 4. State-wise Total GST Revenue |
|  | 5. Monthly Trend Analysis  6.Pie chart of tax type shares  7.Outliers |
| 5 | Conclusion | | [Conclusion](#Conclusion) |
| 6 | Future Scope | | [Future](#FutureScope) |
| 7 | References | | [References](#References) |

**1. Introduction**

This dataset represents **state-wise and month-wise GST revenue data** collected across India. It includes four key types of GST components:

* **CGST** (Central Goods and Services Tax)
* **SGST** (State Goods and Services Tax)
* **IGST** (Integrated Goods and Services Tax)
* **CESS** (Compensation Cess
* Each record in the dataset corresponds to one state's GST revenue report for a specific month. The dataset is valuable for analyzing how GST collections vary by **state**, **month**, and **type of tax**.

**2. Overview**

**Total Rows (Entries):** 2,199  
Each row is one data point — a specific state's revenue report for a given month.

* **Total Columns:** 11  
  These include details about the state, month, year, and GST component-wise revenue.
* **Time Range:**  
  The srcYear, srcMonth, and Month columns indicate that the dataset spans across **multiple financial years** (like 2018-19, etc.), and **calendar months** (like April 2018).
* **Data Granularity:**
  + **Geographical granularity**: By state/UT
  + **Time granularity**: By month
* **Missing Data:**  
  A few entries in the revenue columns (especially CESS Tax Revenue) are missing — which could be due to zero collection or unreported data.

**3. Purpose and Relevance**

**Purpose**

The dataset is intended to track the **collection performance of GST** components at the **state level**. It's structured to:

* Enable year-over-year and month-over-month analysis of GST collections.
* Compare GST revenue across states.
* Understand the distribution between **central, state, and inter-state taxes**.
* Monitor trends and forecast future revenues.

**Use Cases**

* **Government & Policymakers**: To analyze which states are performing better in GST collection, identify regional disparities, and make decisions on tax allocation or reforms.
* **Economists & Analysts**:  
  For economic forecasting, understanding consumption patterns, and evaluating GST effectiveness.
* **Researchers & Academics**:  
  To study the impact of GST implementation and its evolution over time.
* **Public Finance Students**:  
  A practical example of tax structure, federal distribution, and fiscal analysis.

**4. Columns and Their Significance**

| **Column Name** | **Description & Significance** |
| --- | --- |
| srcStateName | The name of the state or union territory. This helps break down the revenue by region, which is crucial for state-level comparison and analysis. |
| srcYear | Represents the **financial year** in YYYY-YY format (e.g., 2018-19). It shows which fiscal year the data belongs to. Useful for aggregating data annually. |
| srcMonth | Shows the **month and year** in short form (e.g., Apr-18). Good for tracking monthly trends or time series analysis. |
| Central Goods and Services Tax ( CGST ) Revenue | The amount collected under CGST. This portion goes to the **central government** and is levied on intra-state supplies. |
| State Goods and Services Tax ( SGST )Revenue | The amount collected under SGST, which is the **state’s share** of tax on intra-state sales. Critical for analyzing each state’s revenue. |
| Integrated Goods and Services Tax ( IGST )Revenue | This is levied on **inter-state supplies** and imports. The revenue is shared between the center and states. Useful to understand inter-state commerce trends. |
| CESS Tax Revenue | CESS is an **extra tax** levied on luxury and sin goods. The revenue from CESS is used to compensate states for any loss due to GST. Important for fiscal compensation studies. |
| YearCode | A numerical representation of the year (e.g., 2018). Helpful for sorting or numeric operations. |
| Year | A **descriptive year format** (e.g., "Calendar Year (Jan - Dec), 2018"). Adds clarity and redundancy for better understanding. |
| MonthCode | A numeric code combining year and month (e.g., 201804 for April 2018). Useful for precise sorting and building time series plots. |
| Month | A descriptive month name (e.g., “April, 2018”). Improves readability and can be used in visualizations or reports. |

**2. Source of dataset**

<https://ndap.niti.gov.in/dataset/7114>

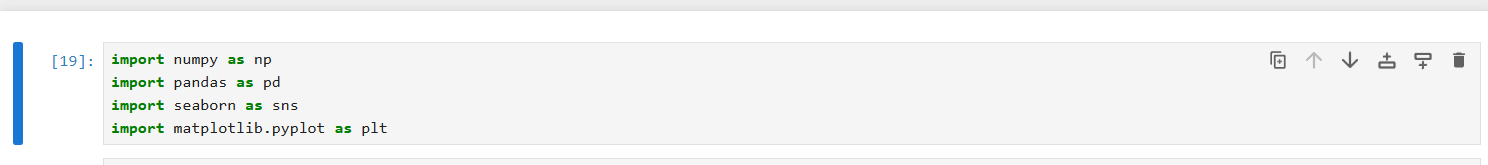
Showing datasetA screenshot of a computer

AI-generated content may be incorrect.

**3.** **Exploratory Data Analysis (EDA) Process**

Exploratory Data Analysis (EDA) is a vital part of any data science or analytical project. It helps in developing a deep understanding of the dataset, identifying any inconsistencies, and preparing the data for further analysis.Each step is described below with screenshots:

Step1 : First step is to import the libraries



Step2: Viewing basic information of dataset

This command provides essential details such as the total number of entries (rows), the number of columns (features), the type of data stored in each column (e.g., integer, float, string), and the number of non-null (non-missing) values in each column.

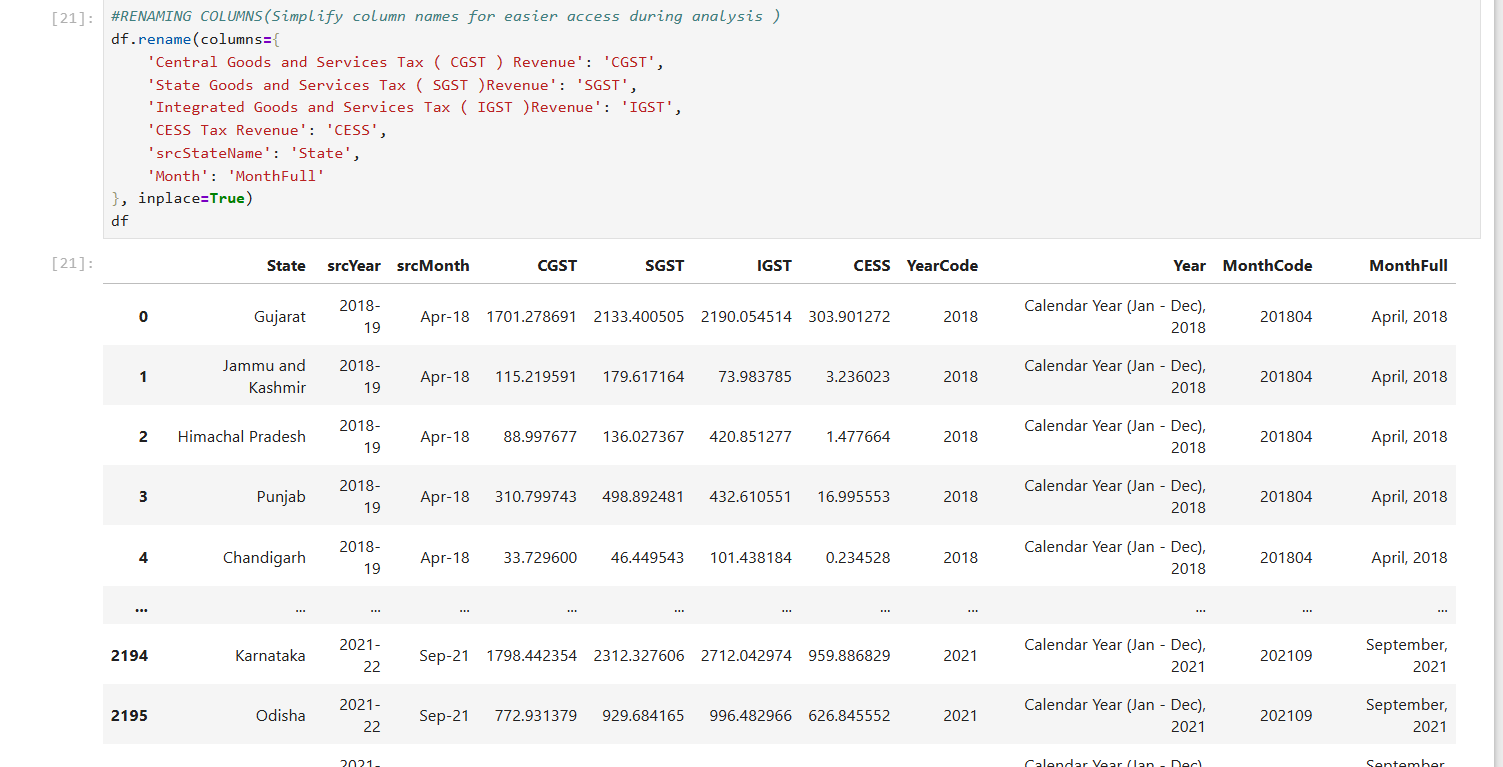
This overview helps us understand the completeness and consistency of the data, and gives a general idea of how to proceed with further cleaning or transformation.

A screenshot of a computer

AI-generated content may be incorrect.

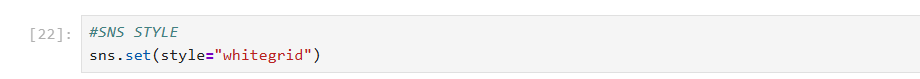
Step 3: Renaming columns

Renamed columns reduce potential mistakes and misinterpretations, ensuring that further operations (like filtering, plotting, or aggregating data) are straightforward and less prone to errors. Shorter names (like CGST, SGST, etc.) make it easier to reference columns throughout the analysis code. Streamlined column names improve code readability and help maintain consistency when sharing the analysis with others.



Step 4: Setting seaborn style

Using a preset style ensures that all subsequent plots share a consistent look and feel. The "whitegrid" setting adds a subtle background grid to plots, which aids in reading the values off the graph and makes patterns or trends easier to spot.A refined aesthetic makes the overall report more appealing and understandable to both technical and non-technical audiences.



**4. Exploratory Data Analysis and Visualization**

**Step 1 :** Summary Statistics

Provides a statistical snapshot (mean, median, min, max, standard deviation) for the revenue columns. This helps in:

* Understanding value ranges
* Spotting unusually high/low values
* Identifying data skewness

A white screen with black text

AI-generated content may be incorrect.

**Step 2:** Correlation heatmap

Shows how tax types (CGST, SGST, IGST, CESS) are correlated with each other.

This helps to:

* Understand if increases in one tax type influence others
* Detect multicollinearity (important for modeling)

A screenshot of a computer

AI-generated content may be incorrect.

Step 3: Revenue Distribution Plots

Visualizes how frequently different revenue values occur for each tax type.

Key insights:

* **KDE curve** (smooth line) shows distribution shape
* Can detect outliers, skewness, and normality

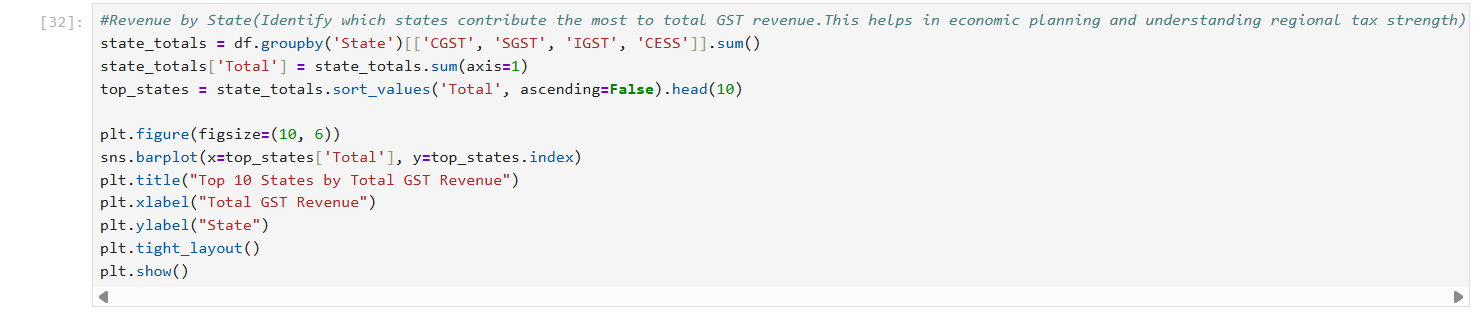
A screenshot of a computer screen

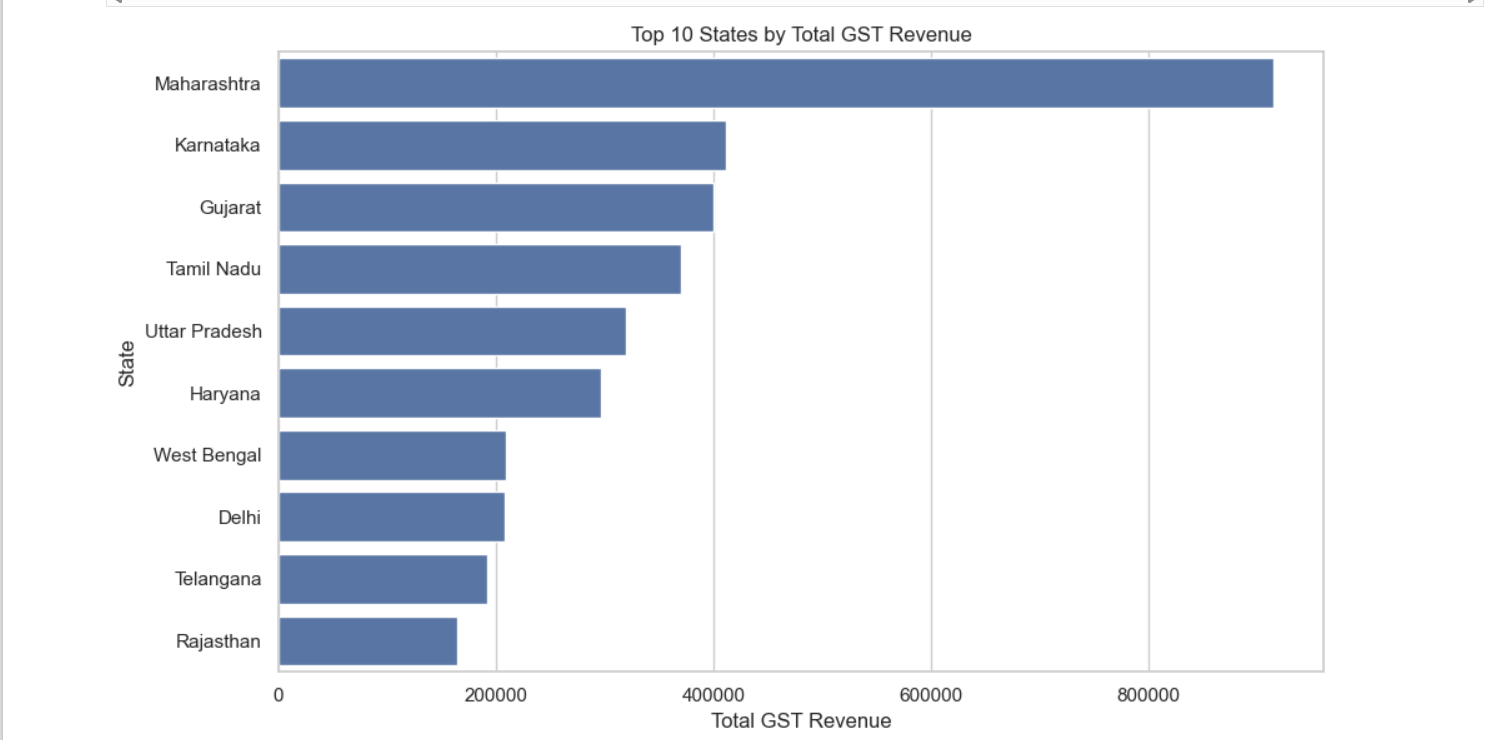
AI-generated content may be incorrect.

Step 4: State-wise Total GST Revenue

To rank **states by their total GST revenue** — identifies:

* High-performing states (like Maharashtra, Karnataka)
* Regional revenue contributors Useful for budget planning and tax allocation strategies.

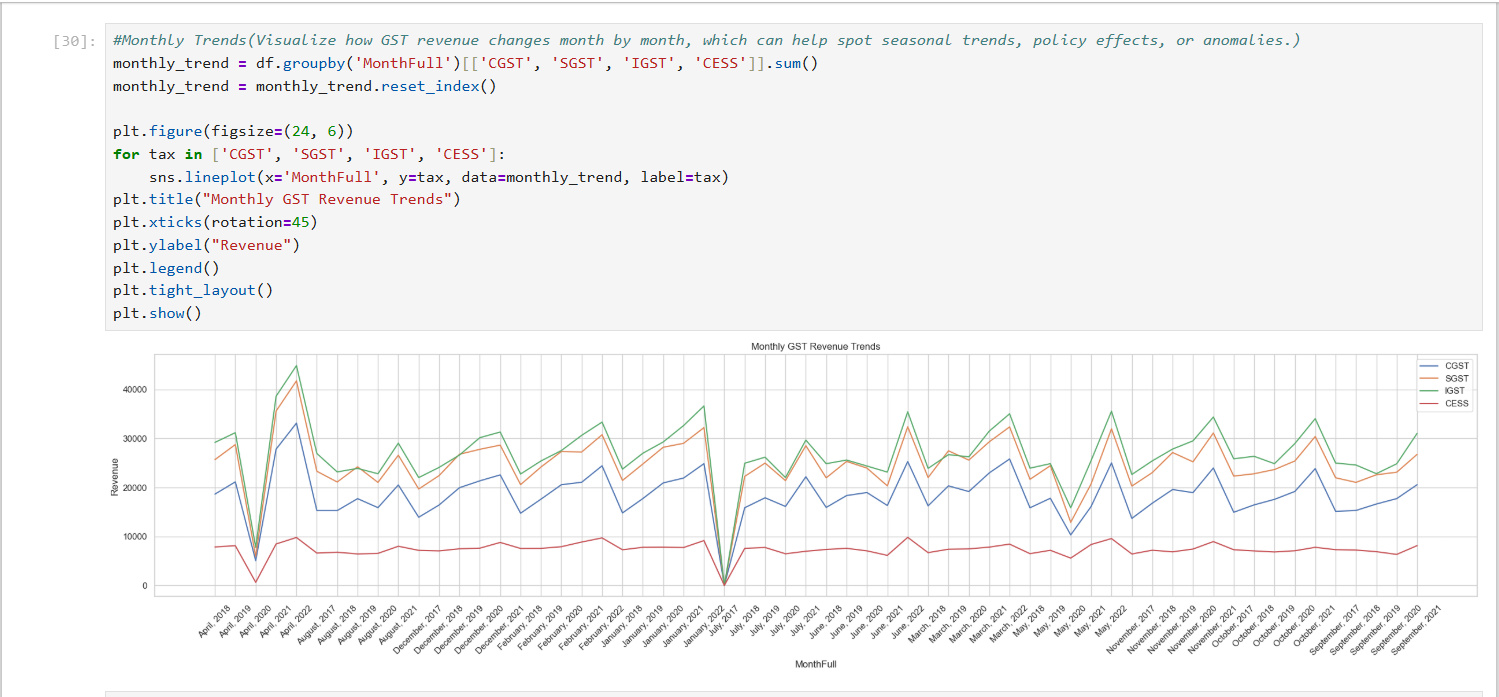




Step 5: Monthly Trend Analysis

Tracks how revenue changes over time across months. Helps in:

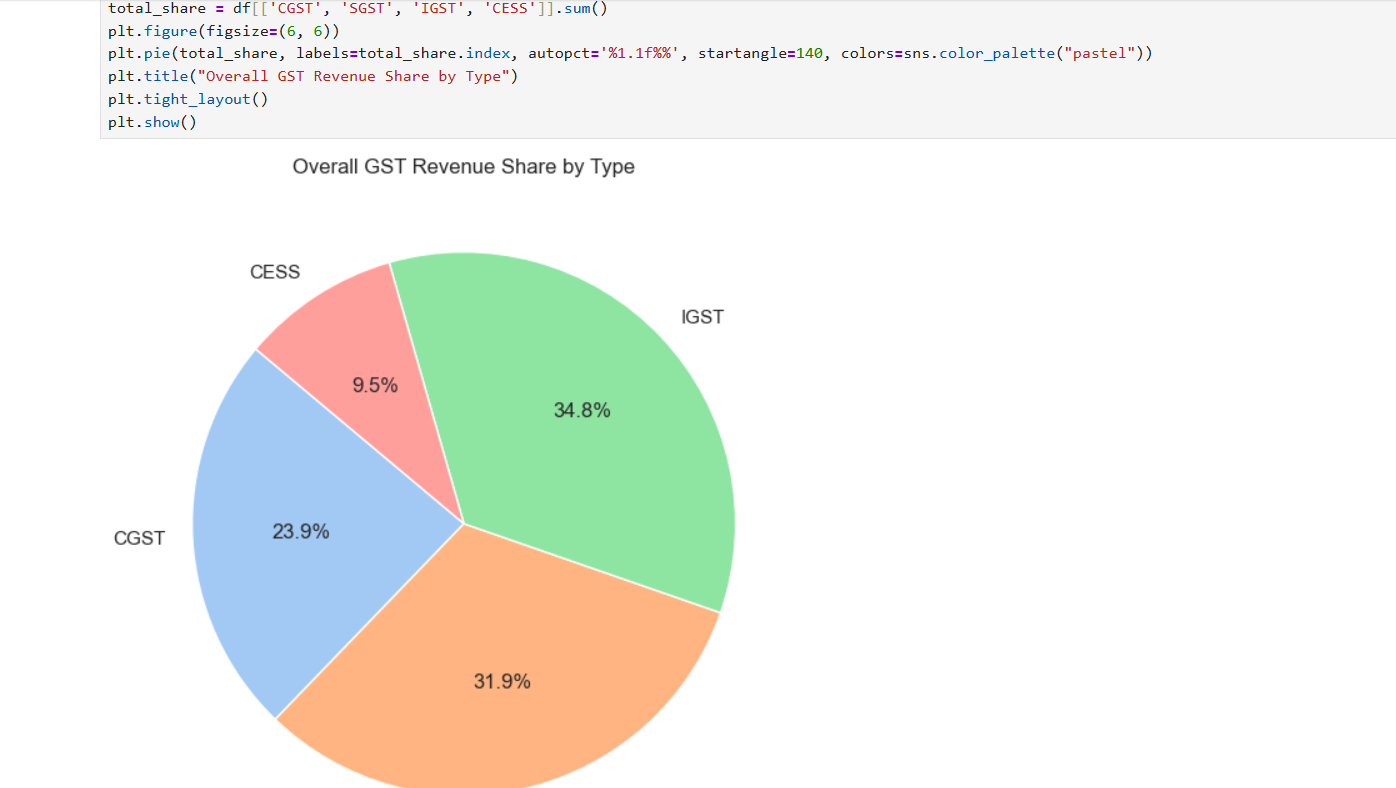
* Detecting **seasonal patterns** (e.g., festive seasons)
* Spotting **dips or surges** due to economic or policy changes



Step 6: Pie Chart of Tax Type Shares

Shows the **proportional share** of each GST type in overall collections. Useful for:

* Understanding tax structure
* Highlighting which component contributes most to the treasury



Step 7: Outlier Detection (CGST)

Identifies **extreme values** in CGST using **IQR (Interquartile Range)** method.

This step is essential to:

* Detect possible data errors or reporting spikes
* Decide if outliers should be removed or explained

A screenshot of a computer

AI-generated content may be incorrect.

**5. Summary**

1. Data Cleaning and Preprocessing

* Renamed lengthy columns for ease of use
* Checked for missing values and data types
* Summarized data to understand scale, skewness, and range

2. Univariate and Multivariate Analysis

* Visualized individual tax distributions using histograms
* Used .describe() and .corr() to get statistical and correlation insights
* Generated heatmaps to see relationships among tax types

3. Temporal and Geographical Trends

* Analyzed total tax revenue state-wise and identified top contributors
* Studied monthly revenue patterns to understand seasonality or cyclical trends
* Visualized tax type contribution using pie charts

4. Outlier Detection

* Applied Interquartile Range (IQR) method to detect unusually high/low CGST values

**6. Future scopes**

The project focuses on analyzing India's Goods and Services Tax (GST) data across states, months, and tax types. This type of fiscal dataset holds huge potential beyond EDA. Here's how you can extend, improve, and apply this work:

1. Predictive Analytics and Forecasting

What to Do:

* Build machine learning models (e.g., ARIMA, LSTM, or XGBoost) to predict future GST revenue by state or tax type.
* Use time-series data to forecast monthly revenue trends.

Why It Matters:

* Helps government departments forecast collections and set budgets.
* Enables early warnings for underperformance or unexpected revenue drops.
* Improves financial planning and policy-making.

2. Interactive Dashboards for Real-Time Insights

What to Do:

* Use tools like Power BI, Tableau, or Plotly Dash/Streamlit to build dashboards.
* Add filters for state, tax type, or date range.

Why It Matters:

* Makes the data easily accessible to non-technical users (government officials, stakeholders).
* Enables on-demand insights without needing to re-run code.
* Encourages data-driven decision making at all levels.

3. Automated Data Cleaning & Ingestion Pipelines

What to Do:

* Set up ETL pipelines to:
  + Automatically fetch new monthly data
  + Clean, validate, and format the data
  + Append it to your existing dataset

Why It Matters:

* Keeps your system up-to-date without manual intervention.
* Ensures that fresh, reliable data is always available for analysis and decision-making.

4. Advanced Statistical Analysis

What to Do:

* Perform regression analysis, hypothesis testing, or clustering:
  + Is GST growth tied to state GDP?
  + Do industrial states have different patterns than agrarian ones?

Why It Matters:

* Helps uncover underlying economic relationships.
* Allows you to compare states beyond raw tax numbers (e.g., tax per capita, efficiency ratios).

5. Integration with Economic Indicators

What to Do:

* Merge GST data with:
  + Population
  + GDP
  + Inflation
  + Employment/unemployment

Why It Matters:

* Enables multidimensional insights:
  + Are richer states collecting more tax proportionally?
  + Is GST growth aligned with economic growth?
* Helps policymakers fine-tune taxation systems based on broader trends.

6. Text Analytics on Policy Announcements

What to Do:

* Scrape or use past GST Council meeting summaries, budget speeches, and government press releases.
* Apply NLP techniques to extract major themes or correlate with revenue shifts.

Why It Matters:

* Adds context to revenue changes (e.g., tax rate changes, exemptions).
* Combines quantitative and qualitative analysis for deeper insight.

7. Policy Simulation and Scenario Modeling

What to Do:

* Create models that simulate:
  + Effects of changing tax rates
  + Introducing new GST slabs
  + Removing exemptions

Why It Matters:

* Provides “what-if” analysis tools for policy experts.
* Allows policymakers to test outcomes before implementing real changes.

8. Anomaly & Fraud Detection

What to Do:

* Use unsupervised learning (e.g., Isolation Forests) to spot:
  + Fake filings
  + Mismatched tax patterns
  + Revenue manipulation

Why It Matters:

* Supports anti-evasion efforts.
* Helps flag suspicious entities automatically for auditing.

Long-Term Impact

| Domain | Impact |
| --- | --- |
| Government | Smarter budgeting, better state comparisons, improved tax policy |
| Economy | Enhanced transparency, resource allocation based on data |
| Tech | Integration of AI/ML, automation pipelines, dashboarding |
| Academia | Use as a case study for public finance, data science, or policy |

**7. References**

1. https://www.gst.gov.in
2. https://data.gov.in
3. Seaborn and Pandas official documentation
4. Python Data Analysis Cookbook