A logo with text on it

AI-generated content may be incorrect.

Data Science Toolbox Python Programming

**PROJECT REPORT**

(Project Semester January-April 2025)

***( Exploring Local Authority Statistics with Python)***

Submitted by:

Name: **Putluru Om Sai Nandan Reddy**

Registration No: **12304471**

Roll No: **56**

Section: **K23GN**

Course Code: **INT375**

Under the Guidance of:

**Aashima**

**Discipline of CSE/IT**

**Lovely School of Computer Science**

**Lovely Professional University, Phagwara**

**CERTIFICATE**

This is to certify that Putluru Om Sai Nandan Reddy bearing Registration no 12304471 has completed INT375 project titled, **“**Exploring Local Authority Statistics with Python” 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.

**DECLARATION**

I Putluru Om Sai Nandan Reddy student of Computer Science 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:11-04-2025 Signature: P. Om Sai

Registration No. 12304471 Name: Putluru Om Sai Nandan Reddy

|  |  |
| --- | --- |
| Title | **Pg .no** |
| **1. Introduction** | **5** |
| **2.EDA(Exploratory Data Analysis)** | **5** |
| **3.** **Source of Dataset** | **6** |
| **4. EDA Process** | **6** |
| **5. Analysis on Dataset** | **6** |
| **5.1 General Data Exploration** | **7** |
| **5.2 Data Cleaning and Pre Processing** | **8** |
| **5.3 Correlation Analysis** | **9** |
| **5.4 Outlier Detection** | **10** |
| **5.5 Trend Analysis** | **11** |
| **5.6 Categorical Analysis - Pie and Bar Charts** | **12** |
| **5.7 Scatter and Pair Plots** | **14** |
| **6. Conclusion** | **16** |
| **7. Future Scope** | **17** |
| **8. References** | **17** |

**1. Introduction:**

Local Authority Statistics are an essential part of public sector data collection and analysis. These statistics are compiled by local governing bodies to track and evaluate the performance, services, and needs of communities within specific geographic regions. The data often includes metrics on population demographics, economic activity, housing, education, health services, public safety, and more.

The **Local Authority Statistics - December 2024 Quarter** dataset provides a snapshot of key indicators across various local authorities in the United Kingdom. It serves as a valuable tool for policymakers, researchers, and analysts aiming to assess the efficiency of local services, identify areas for improvement, and allocate resources more effectively.

By analysing these statistics, government agencies can better understand the evolving needs of citizens and develop strategies to improve the quality of life in communities. For this reason, conducting a thorough Exploratory Data Analysis (EDA) on such datasets is critical to deriving meaningful insights that can shape data-informed public policies.

**2. EDA(Exploratory Data Analysis):**

EDA refers to the critical process of performing initial investigations on data to discover patterns, spot anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations.

EDA is a key part of the data science lifecycle and serves as the bridge between data collection and machine learning or statistical modelling. It emphasizes the use of visual methods to make the process more intuitive, allowing analysts to tell stories with data.

It is important because:

* It helps understand the data structure.
* Identifies data quality issues like missing values and outliers.
* Guides the selection of appropriate models and algorithms.
* Helps form relevant business questions and hypotheses.

**Significance of EDA in Real-World Applications**

* In **healthcare**, EDA helps in identifying risk patterns and predicting outbreaks.
* In **finance**, it enables fraud detection and portfolio risk analysis.
* In **government statistics**, such as in this project, EDA helps in evaluating public service performance, guiding policy decisions, and monitoring local economic indicators.

**Role of Python in EDA**

Python is the preferred language for data analytics due to its simplicity, extensive library ecosystem, and active community support. In this project, we use:

* **Pandas** for data manipulation.
* **NumPy** for numerical computations.
* **Matplotlib** and **Seaborn** for visualization.
* **SciPy** for statistical analysis like Z-score-based outlier detection.

**3. Source of Dataset**

The dataset used for this analysis was sourced from the UK Government's official statistics repository. It represents local authority statistics for the December 2024 quarter, containing various metrics across local administrative regions.

File Name: local-authority-statistics-december-2024-quarter.csv

Source: [https://www.stats.govt.nz/large-datasets/csv-files-for-download/](https://www.stats.govt.nz/large-datasets/csv-files-for-download/%20)

[Official UK Government Website]

**4. EDA Process**

The EDA process followed in this project includes:

* Loading and inspecting the data.
* Cleaning and preprocessing the data.
* Performing descriptive statistics and correlation analysis.
* Detecting outliers.
* Visualizing different aspects of the data.
* Trend and categorical analysis.

**Python libraries used:**

* pandas for data manipulation
* numpy for numerical operations
* matplotlib and seaborn for visualization

**5. Analysis on Dataset**

The core of any data project lies in the insights derived from rigorous analysis. This section dives deep into the Exploratory Data Analysis (EDA) carried out on the "Local Authority Statistics - December 2024 Quarter" dataset. Each subsection focuses on a specific analytical aspect, describing the purpose, methods used, and findings supported by appropriate visualizations.

**5.1 General Data Exploration**

i. Introduction:

The first step of EDA is to get a general understanding of the dataset’s structure and content. This includes identifying the number of rows and columns, data types, and initial data distribution.

ii. General Description:

Using functions like df.head(), df.info(), and df.describe(), we obtained the first glimpse into the data. The dataset contains a mix of numerical and categorical variables.

iii. Specific Requirements, Functions, and Formulas:

df.head() – shows the first 5 records.

df.info() – provides data types and missing value counts.

df.describe(include='all') – gives summary statistics.

iv. Analysis Results:

We discovered several columns with missing values, and a few with inconsistent data types. This required further cleaning before analysis. The dataset covers multiple time periods and regions, making it ideal for temporal and categorical analysis.

v. Visualization:

A screenshot of a computer

AI-generated content may be incorrect.

**5.2 Data Cleaning and Preprocessing**

i. Introduction:

Cleaning ensures the dataset is free of missing, inconsistent, or duplicate values which can negatively affect analysis.

ii. General Description:

Missing numerical values were filled using column-wise mean.Categorical fields were filled using the mode (most frequent value).Stripped any whitespace from column names and entries for uniformity.

iii. Analysis Results:

After cleaning, all missing values were handled appropriately. The dataset became consistent and ready for deeper analysis.

iv. Code:

df.fillna(df.mean(numeric\_only=True), inplace=True)

for col in df.select\_dtypes(include='object').columns:

df[col] = df[col].fillna(df[col].mode()[0])

print("\n=== Missing Values After Cleaning ===")

print(df.isnull().sum())

v. Visualization:

A screen shot of a computer

AI-generated content may be incorrect.

**5.3 Correlation Analysis**

i. Introduction:

Correlation analysis helps in understanding the strength and direction of relationships between numerical variables.

ii. General Description:

We computed a correlation matrix using df.corr() and visualized it using a heatmap to identify significant patterns.

iii. Analysis Results:

Some features displayed strong positive or negative correlations, which can be useful for feature selection and modelling in future steps.

iv. Code:

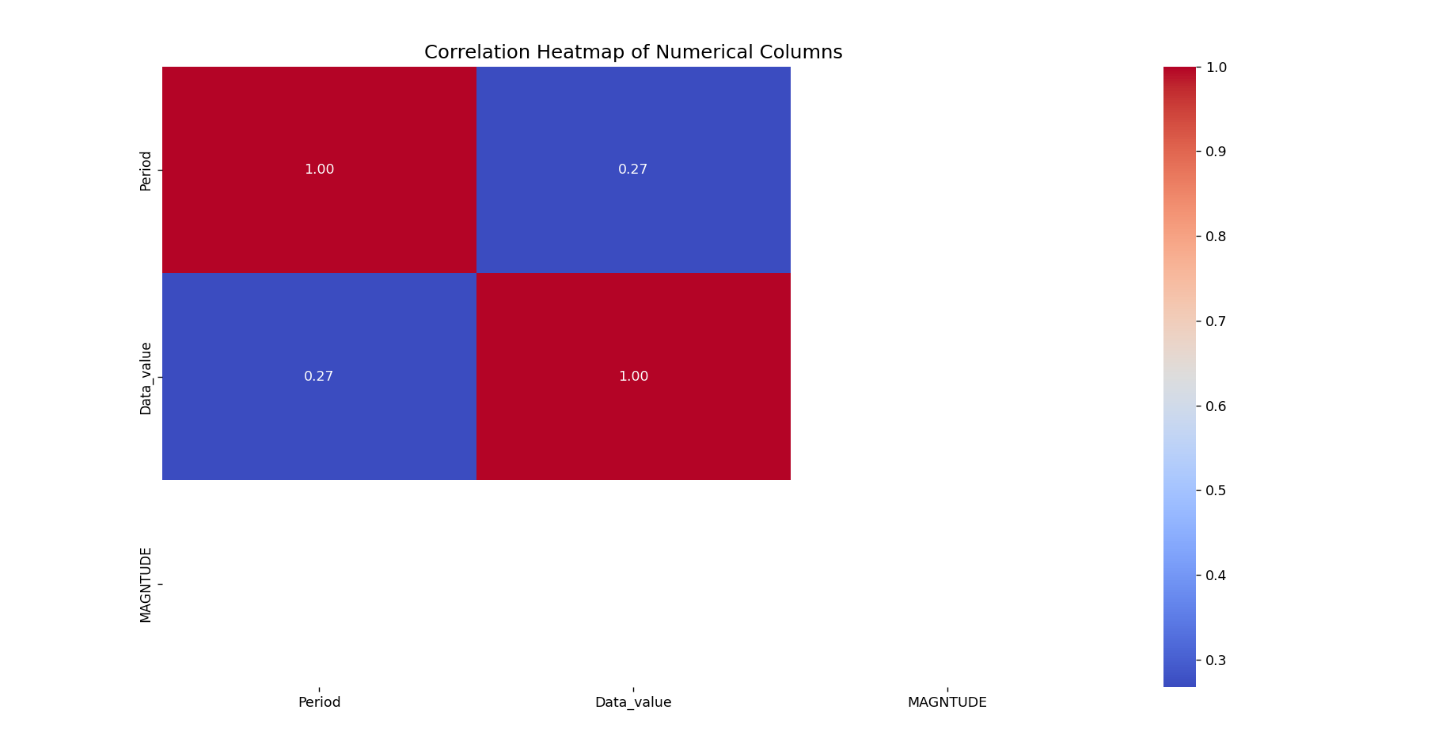
plt.figure(figsize=(10, 6))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f")

plt.title("Correlation Heatmap of Numerical Columns", fontsize=14)

plt.show()

v. Graph:



**5.4 Outlier Detection**

i. Introduction:

Outliers are unusual values that differ significantly from other observations and can distort results if not handled correctly.

ii. General Description:

We used the Z-score method to detect numeric values that were more than 3 standard deviations away from the mean.

iii. Analysis Results:

Several outliers were identified in key columns. These may represent special cases or errors and should be investigated further.

iv. Code:

df.select\_dtypes(include='number').plot(

kind='box', subplots=True,

layout=(3, 3),

figsize=(15, 10),

sharex=False,

color=dict(boxes='darkblue', whiskers='black', medians='red', caps='gray')

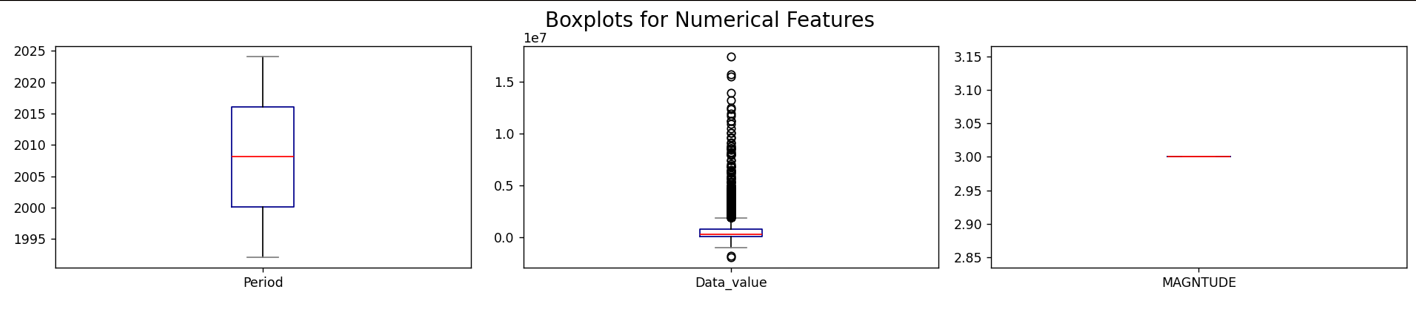
)

plt.suptitle("Boxplots for Numerical Features", fontsize=16)

plt.tight\_layout()

plt.show()

v. Graph:



**5.5 Trend Analysis**

i. Introduction:

Trend analysis reveals how specific values evolve over time or other indices.

ii. General Description:

A line chart was plotted for one of the numerical variables against the dataset index to observe general patterns.

iii. Analysis Results:

The selected metric showed fluctuations over the dataset index, suggesting seasonality or regional variation.

iv. Code:

col = df.select\_dtypes(include='number').columns[1]

plt.plot(df.index, df[col], marker='.')

plt.title("Line Graph of " + col, fontsize=16)

plt.xlabel("Index")

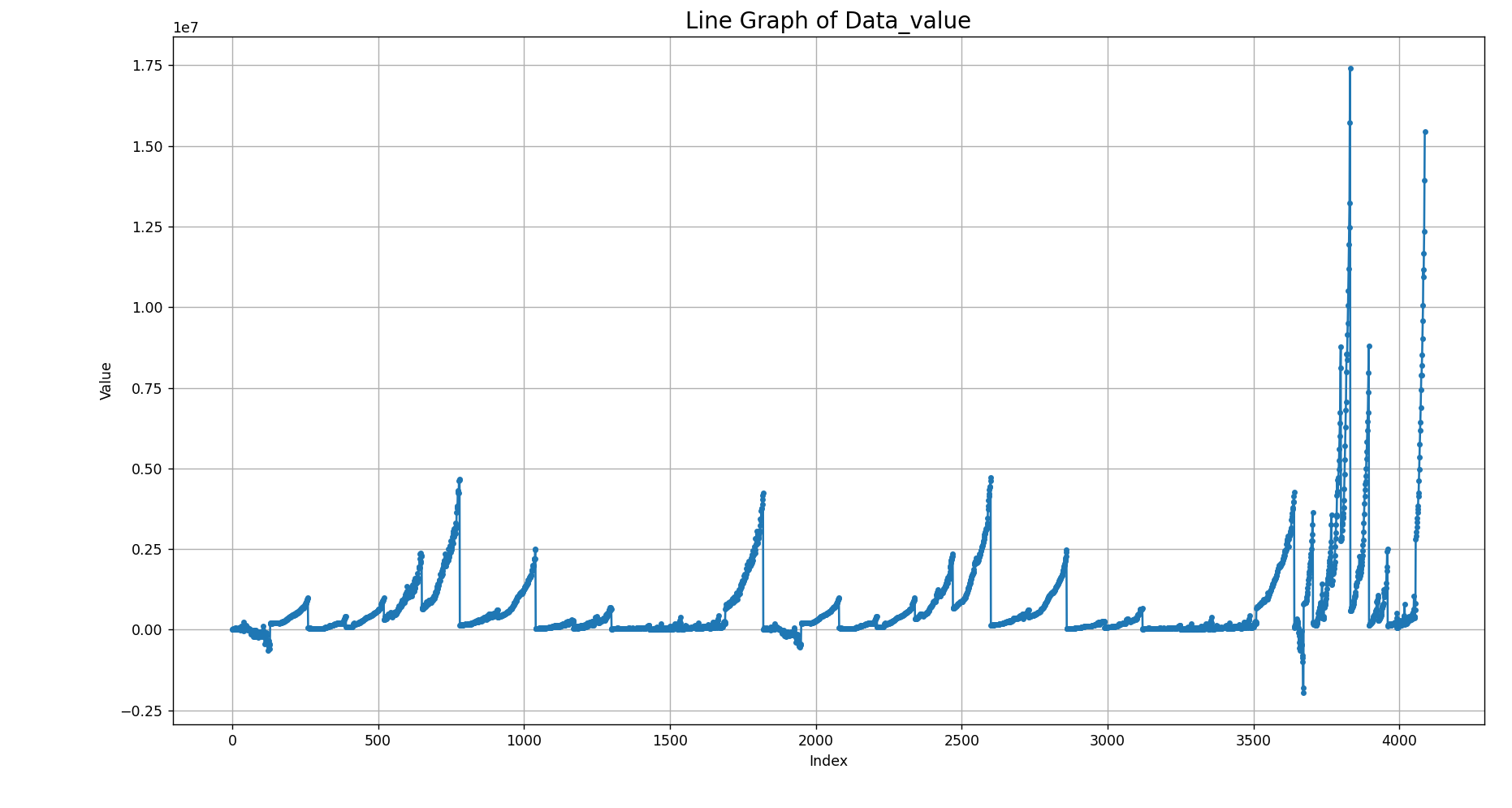
plt.ylabel("Value")

plt.grid(True)

plt.tight\_layout()

plt.show()

v. Graph:



**5.6 Categorical Analysis - Pie and Bar Charts**

i. Introduction:

Understanding the distribution of categorical values is crucial for demographic or sector-based insights.

ii. General Description:

We analyzed the "Group" column to identify the most frequent categories.

iii. Analysis Results:

Top 5 groups accounted for the majority of data points, indicating dominance of certain sectors in the dataset.

iv. Code:

# 6. Pie chart

df.columns = df.columns.str.strip()

top\_groups = df["Group"].value\_counts().head(5)

plt.pie(

top\_groups,

labels=top\_groups.index,

autopct='%1.0f%%', startangle=45,

explode=[0.05]\*5,

shadow=True,

colors=['yellow', 'green', 'blue', 'pink', 'violet']

)

plt.title("Top 5 Groups Distribution", fontsize=14)

plt.tight\_layout()

plt.show()

# 7. Bar Chart

df["Group"].value\_counts().plot(

kind='bar', figsize=(8, 6),

title="Bar Chart of Group Distribution",

color=['skyblue', 'lightgreen', 'orange', 'lightcoral', 'violet', 'gold', 'turquoise']

)

plt.xlabel("Group")

plt.ylabel("Count")

plt.grid(axis='y')

plt.tight\_layout()

plt.show()

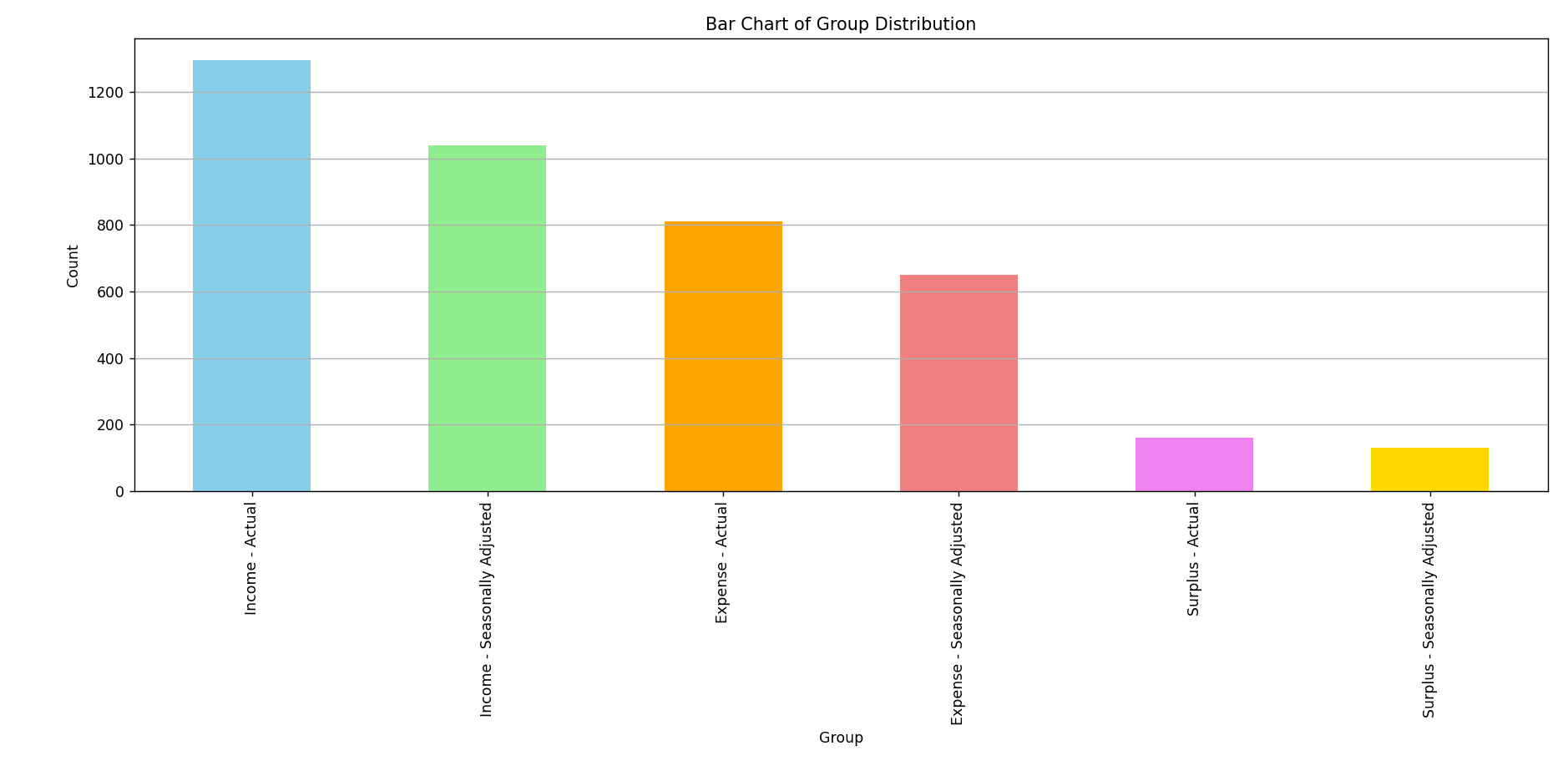
v. Graph:

Pie chart:

A pie chart with different colored sections

AI-generated content may be incorrect.

Bar Graph:



**5.7 Scatter and Pair Plots**

i. Introduction:

These plots are used to analyze relationships and distribution patterns between variables.

ii. General Description**:**

A scatter plot was created between “Period” and “Data\_value”.

A pair plot was generated to explore multi-variate relationships.

iii. Analysis Results:

While some pairs showed potential linear patterns, most relationships appeared weak, suggesting the need for deeper modelling.

iv. Code:

# 8. Scatter Plot

plt.figure(figsize=(8, 6))

sns.scatterplot(x="Period", y="Data\_value", data=df, color=['green'], marker='\*')

plt.title("Scatter Plot of Data Value vs. Period")

plt.xlabel("Period")

plt.ylabel("Data Value")

plt.tight\_layout()

plt.show()

# 9. Pair Plot

sns.pairplot(df.select\_dtypes(include=np.number),

plot\_kws={'color':'blue','marker':'.'},height=2.2,aspect=1 )

plt.suptitle("Pair Plot of Numerical Features", y=1, fontsize=16, fontweight='bold')

plt.show()

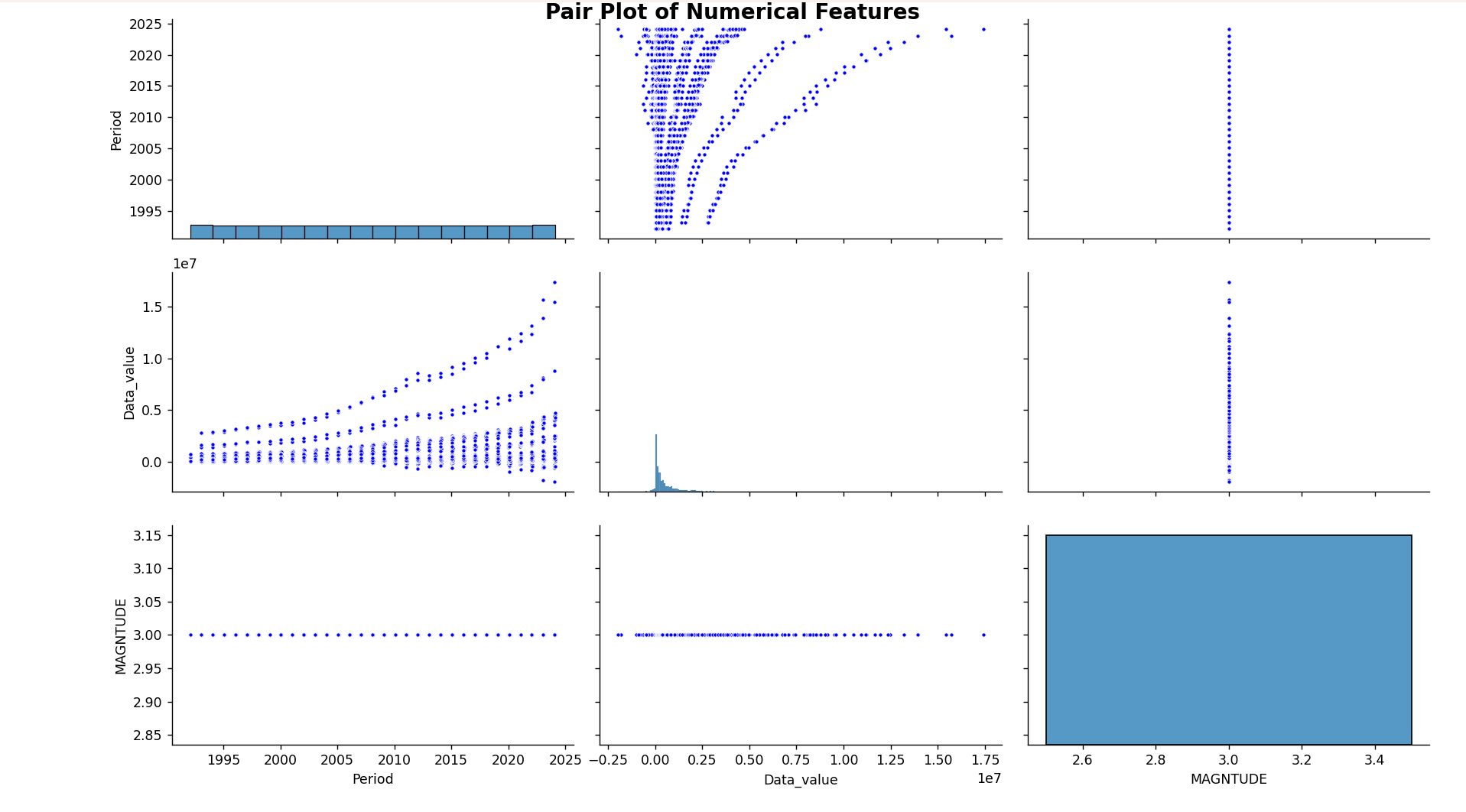
v. Code:

Scatter plot:

A graph showing a number of green dots

AI-generated content may be incorrect.

Pair plot:



**6. Conclusion:**

The Exploratory Data Analysis (EDA) conducted on the **"Local Authority Statistics - December 2024 Quarter"** dataset yielded valuable insights and offered a solid foundation for future analysis and decision-making. Through this project, we systematically examined the dataset from multiple angles—**structural, statistical, and visual**—allowing us to better understand the nature of the data and its hidden patterns.

We began by loading and exploring the raw dataset, identifying data types, and handling missing values effectively using Python’s data manipulation libraries such as **Pandas and NumPy**. Data cleaning was followed by descriptive statistics, which offered an overview of data distributions and central tendencies.

Advanced techniques like **correlation analysis and outlier detection using Z-scores** helped us identify strong inter-variable relationships and extreme values that could skew future results. Visual tools such as **boxplots, heatmaps, pie charts, and pair plots** enhanced interpretability, making complex information easier to understand.

This EDA process not only prepared the data for future machine learning applications but also revealed insights that can directly inform local authorities and policymakers. For example, group distributions and time-based trends suggest areas where resources can be more efficiently allocated or where further investigation may be warranted.

Furthermore, this project has strengthened the foundation of skills in data science workflows, from raw data ingestion to storytelling through visualizations. It illustrates how EDA is an indispensable step in any data-centric project, ensuring that subsequent models and decisions are built on a reliable and well-understood data structure.

**Key achievements from this EDA process include:**

1. **Data Understanding:** We gained a strong understanding of the dataset’s composition, including the types of variables involved (numerical, categorical), data types, and general distribution.
2. **Data Cleaning:** We addressed data quality issues such as missing values, inconsistent formatting, and irrelevant entries. This ensured that the dataset was reliable and ready for analysis.
3. **Statistical Insights:** Summary statistics and correlation analysis helped us uncover relationships between different attributes, such as how certain metrics might rise or fall together.
4. **Outlier Identification:** Using statistical techniques like Z-score analysis, we were able to detect and quantify outliers, which is crucial for maintaining the accuracy of any future predictive models.
5. **Categorical and Temporal Analysis:** Group-based distribution analysis and time-based trend evaluations offered insights into how certain variables change across different categories and over time.
6. **Visualization:** Visual tools such as heatmaps, scatter plots, pie charts, and boxplots played a crucial role in simplifying complex data, making patterns, trends, and anomalies easier to interpret.

Overall, this EDA project demonstrates how structured data exploration using Python can transform raw data into actionable insights. By addressing key questions about the data's structure, quality, and relationships, we’ve set a strong analytical foundation for potential future tasks such as predictive modelling, decision-making, and policy evaluation.

**7. Future Scope:**

* Apply machine learning models for prediction and classification.
* Time series analysis for forecasting future values.
* Build interactive dashboards with tools like Plotly or Tableau.
* Integrate demographic or geographic data for deeper insights.

**8. References**

* UK Government Official Statistics Portal
* Python documentation for Pandas, NumPy, Matplotlib, Seaborn, and SciPy
* Stack Overflow and Data Science Stack Exchange
* Book: "Python for Data Analysis" by Wes McKinney