**PUBLIC TRANSPORT ANALYSIS**

**OBJECTIVE**

To analyze and improve the efficiency, reliability, and quality of public transport services in a given region, with the goal of enhancing the overall passenger experience and promoting sustainable transportation solutions.

This objective captures the core purpose of a public transport analysis, which typically involves assessing the performance of public transportation, identifying areas for improvement, and working towards a more efficient and customer-centric transport system. The specific goals and scope of the analysis can be further refined to meet the needs of the project.

**1. Data Collection:**

Gather relevant data sources, which may include trip records, route information, passenger feedback, and any other data related to public transport. Ensure that the data is well-documented and organized.

**2. Data Import:**

Import the collected data into your data analysis environment, such as Jupyter Notebook, using data import functions or libraries. Ensure that you have a clear understanding of the data format and structure.

**3. Initial Data Exploration:**

Take an initial look at the dataset to understand its characteristics:

Review the first few rows to check for any obvious issues or anomalies.

Examine basic data statistics to understand the data's central tendencies and distributions.

Check for missing values and outliers.

**4. Data Cleaning:**

Perform data cleaning tasks to ensure data quality:

Handle missing values by imputing them or removing rows with missing data.

Correct any inconsistent or erroneous data values.

Remove duplicate records, if any.

**5. Data Transformation:**

Transform the data as needed to prepare it for analysis:

Convert data types (e.g., date columns) to the appropriate format.

Normalize or standardize numerical features if required.

Create new derived features or variables that may be useful for your analysis.

**6. Data Filtering:**

Remove outliers or irrelevant data points to focus on the most relevant information. You can filter data based on specific criteria or conditions**.**

**7. Saving Preprocessed Data:**

Save the cleaned and preprocessed data to a new file to ensure that you retain the original data for reference and to avoid repeating the preprocessing steps.

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

Conduct EDA to visualize and understand the cleaned data better. Generate summary statistics, visualizations, and insights that can guide your subsequent analysis.

**SOURCE CODE**

import pandas as pd

import matplotlib.pyplot as plt

# Read the data from a CSV or Excel file (assuming it's in a file named "public\_transportation\_data.csv")

data = pd.read\_csv("/content/data.xlsx - 20140711.csv")

# Convert the 'WeekBeginning' column to a datetime object

data['WeekBeginning'] = pd.to\_datetime(data['WeekBeginning'])

# Group the data by 'WeekBeginning' and sum the 'NumberOfBoardings' for each week

time\_data = data.groupby('WeekBeginning')['NumberOfBoardings'].sum().reset\_index()

# Create a line chart

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

plt.plot(time\_data['WeekBeginning'], time\_data['NumberOfBoardings'], marker='o', linestyle='-', color='g')

plt.xlabel('Date')

plt.ylabel('Number of Boardings')

plt.title('Trend in Boardings Over Time')

plt.grid(True)

# Display the line chart

plt.tight\_layout()

plt.show()

import pandas as pd

import matplotlib.pyplot as plt

# Sample data (you can replace this with your actual data)

data = {

'TripID': [23631, 23631, 23632, 23633, 23633, 23634, 23634, 23634, 23634],

'RouteID': [100, 100, 100, 100, 100, 100, 100, 100, 100],

'StopID': [14156, 14144, 14132, 12266, 14147, 13907, 14132, 13335, 13875],

'StopName': ['181 Cross Rd', '177 Cross Rd', '175 Cross Rd', 'Zone A Arndale Interchange', '178 Cross Rd', '9A Marion Rd', '175 Cross Rd', '9A Holbrooks Rd', '9 Marion Rd'],

'WeekBeginning': ['6/30/13 0:00', '6/30/13 0:00', '6/30/13 0:00', '6/30/13 0:00', '6/30/13 0:00', '6/30/13 0:00', '6/30/13 0:00', '6/30/13 0:00', '6/30/13 0:00'],

'NumberOfBoardings': [1, 1, 1, 2, 1, 1, 1, 1, 1]

}

# Create a DataFrame from the sample data

df = pd.DataFrame(data)

# Group the data by 'StopName' and calculate the total number of boardings for each stop

stop\_boardings = df.groupby('StopName')['NumberOfBoardings'].sum().reset\_index()

# Sort the stops by the total number of boardings in descending order

stop\_boardings = stop\_boardings.sort\_values(by='NumberOfBoardings', ascending=False)

# Create a bar chart

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

plt.bar(stop\_boardings['StopName'], stop\_boardings['NumberOfBoardings'], color='skyblue')

plt.xlabel('Bus Stop Name')

plt.ylabel('Total Number of Boardings')

plt.title('Number of Boardings by StopName')

plt.xticks(rotation=90)

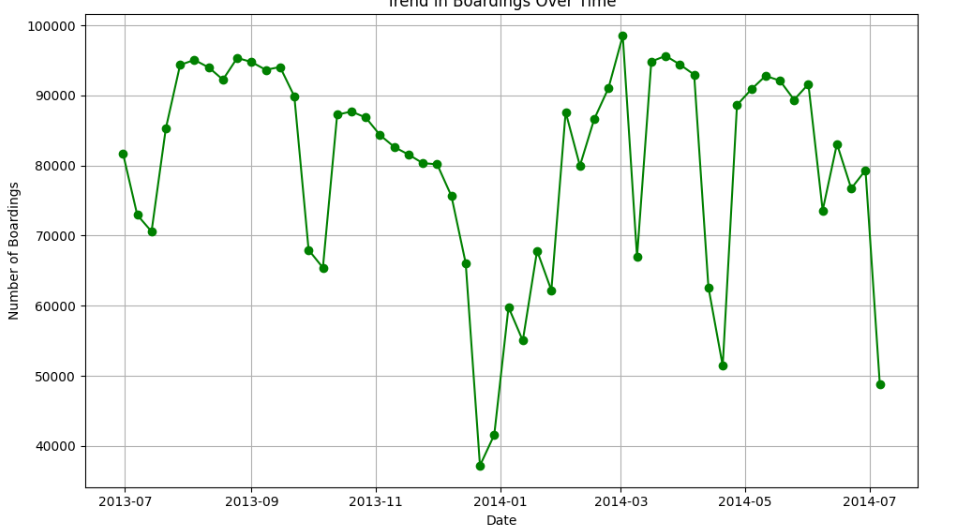
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Display the bar chart

plt.tight\_layout()

plt.show()

**OUTPUT**

****

