Project Title : Public Transportation Efficiency Analysis

Problem Definition:

The project involves analyzing public transportation data to assess service efficiency, on time performance, and passenger feedback. The objective is to provide insights that support transportation improvement initiatives and enhance the overall public transportation experience. This project includes defining analysis objectives, collecting transportation data, designing relevant visualizations in IBM Cognos, and using code for data analysis.

Overview:

This documentation provides a comprehensive overview of all the tasks completed during the session, beginning with Design Thinking to determine how the assigned project should be completed and what tools and techniques are needed. It also includes the incorporation of a machine learning model into the application, loading and pre-processing the provided dataset, and analysing it using a variety of visualisation techniques, such as plots, charts, and graphs, to extract insights and analyze public transport efficiency.

23647	100	13331	9A Holbro	########	3
23647	100	14083	165 Cross	########	1
23647	100	12747	23 Findon	########	1
23647	100	13499	8C Marior	########	2
23647	100	14135	175 Cross	########	1
23647	100	14152	179 Cross	########	1
23647	100	14129	174 Cross	########	1
23648	100	13331	9A Holbro	########	1
23648	100	13931	9B Marior	########	1
23648	100	13269	10 Holbro	########	1
23648	100	13817	8K Marior	########	1
23648	100	14155	180 Cross	########	1
23649	100	12939	18 Critten	########	1
23649	100	14152	179 Cross	########	1
23649	100	13874	9 Marion	########	1
23649	100	14142	176 Cross	########	1
23649	100	14131	11A Maric	########	2
23649	100	14000	10 Marior	########	1
23649	100	14047	10A Maric	########	2
23649	100	13730	8G Marior	########	1
23650	100	13499	8C Marior	########	1
23650	100	13905	9A Marior	########	1
23650	100	14047	10A Maric	########	3
23650	100	14131	11A Maric	########	1
23650	100	13931	9B Marior	########	1
23651	100	12806	21 Critten	########	1
23651	100	13269	10 Holbro	########	1

Design Thinking:

1. Project Objectives:

Objective 1: Assessing On-time Performance

- Develop a machine learning model to predict the estimated arrival time of government buses.
- Achieve a high level of prediction accuracy and reliability to improve on-time performance.

Objective 2: Customer Experience and Satisfaction Improvement

- Passenger Feedback Analysis: Collect and analyze customer feedback to identify pain points and areas for improvement in service quality.
- Technology Integration: Assess the integration of modern technology, such as mobile apps and real-time tracking, to enhance the overall passenger experience and convenience.

Objective 3 : Cost Reduction and Financial Sustainability

- Fare Structure Assessment: Analyze fare pricing and payment systems to ensure they cover operational costs while remaining affordable for passengers.
- Energy Efficiency: Evaluate the environmental impact and energy consumption of public transportation modes, aiming for cost-effective and sustainable alternatives.

Analysis Approach:

Step 1: Data Collection and Integration

Data Source Identification:

Begin by identifying and collecting data from various sources. These sources may include GPS tracking systems on buses and trains, fare collection systems, passenger surveys, traffic cameras, weather reports, and more. It's crucial to pinpoint the relevant data sources that provide information about routes, schedules, ridership, and operational metrics.

Real-Time and Historical Data:

Public transportation analysis benefits from both real-time and historical data. Real-time data provides immediate insights into the current state of the system, such as bus locations or delays. Historical data, on the other hand, helps identify trends and patterns over time, allowing for long-term planning and performance evaluation. Both types of data should be integrated to provide a comprehensive view.

Geospatial Integration:

Geographic information plays a crucial role in public transportation analysis. Geospatial data, such as GIS (Geographic Information System) data and mapping information, allows you to analyze routes, stops, and spatial relationships. Integrating geospatial data with other datasets enables the visualization of transportation networks and helps in optimizing routes and schedules.

Collecting User Feedback:

Collecting passenger feedback for public transportation analysis is essential for improving the quality of services and making data-driven decisions. Here are some steps and methods you can use to collect passenger feedback:

a. **Online Surveys:**

- We can Create online surveys using platforms like SurveyMonkey, Google Forms, or custom-built survey tools.
- Share the survey links through social media, the transportation agency's website, or email newsletters.
- Ask passengers about their overall satisfaction, specific issues, suggestions for improvements, and demographic information.

b. Mobile Apps:

- We Can Develop a dedicated mobile app for passengers to provide feedback easily.
- In that app we can Include features like rating rides, reporting problems, and leaving comments.
- Use push notifications to encourage users to provide feedback after using the service.

c. In-App Feedback:

- If the transportation system has a mobile ticketing or tracking app, integrate a feedback feature directly into the app.
- We can Allow passengers to report issues, submit complaints, or offer suggestions through the app.

d. Social Media Monitoring:

- We can Monitor social media platforms for mentions, comments, and messages related to public transportation.
- Engage with passengers and address their concerns promptly.

e. Customer Service Centers:

- We can Set up customer service centers at key transportation hubs where passengers can provide feedback in person or via forms.
- Ensure staff are trained to handle feedback professionally.

f. Onboard Surveys:

- We can Conduct onboard surveys using paper forms or electronic devices.
- We can Ask passengers to rate their experience and provide comments during their journey.

g. **Focus Groups:**

- We can Organize focus group discussions with passengers to dive deeper into specific issues and gather qualitative insights.
- We can Use these sessions to brainstorm solutions and improvements.

h. Comment Cards:

- We can Place comment cards or suggestion boxes at transportation stops, stations, and vehicles.
- Encourage passengers to drop their comments or suggestions anonymously.

i. Email and Text Surveys:

- We can Send periodic email or text surveys to passengers who have registered with the transportation agency.
- Keep the surveys short and focused on key issues.

j. 12. Data from Complaints and Incident Reports:

- We can Analyze data from formal complaints and incident reports to identify recurring issues.
- We can Use this information to address systemic problems.

k. 14. Regular Feedback Cycles:

- We can Implement a regular feedback cycle and communicate to passengers how their input has led to improvements.
- This can incentivize ongoing participation.

3. **Visualization Selection:**

a. Exploratory Data Analysis:

- In this section, focus on visualizations that help you understand the basic structure of your dataset. Use histograms and box plots to explore the distribution of numerical variables, and bar charts or pie charts for categorical variables. Scatter plots can reveal relationships between pairs of variables. This initial analysis is crucial for gaining insights into the dataset's characteristics and potential patterns.

b. Temporal Analysis:

- If your dataset contains temporal data, consider visualizations like time series plots or calendar heatmaps. Time series plots are excellent for displaying trends and seasonality, while calendar heatmaps can highlight patterns over days, months, or years. These visualizations are essential for understanding how your data evolves over time and identifying long-term trends or periodic fluctuations.

c. Spatial Analysis:

- For datasets with geographic components, choropleth maps or heatmaps can be powerful tools. Choropleth maps use color-coding to represent data values in different geographic regions, offering insights into regional variations. Heatmaps can display concentrations of data points in specific geographic areas, helping you identify hotspots or trends related to location.

d. Correlation and Relationships:

- When exploring relationships between multiple variables, scatter plots matrix, correlation matrices, or network graphs can be valuable. Scatter plots matrix displays scatter plots for pairs of variables, allowing you to quickly identify correlations. Correlation matrices provide a numerical overview of relationships, while network graphs visually represent connections and dependencies between variables or entities in your dataset.

e. Outlier Detection:

- Detecting outliers is crucial in large datasets. Box plots, scatter plots, and violin plots are effective for identifying outliers in numerical data. Heatmaps can be used to visualize anomalies in multivariate datasets. Robust visualization techniques are essential for understanding the impact of outliers on your analyses and ensuring the integrity of your results.

f. Interactive Visualizations and Dashboards:

- Consider incorporating interactive visualizations and dashboards into your project. Tools like Tableau, Power BI, or D3.js allow users to interact with the data dynamically. Interactive charts, filters, and drill-down options enable users to explore specific aspects of the dataset, making the findings more accessible and engaging. Interactive visualizations can enhance user experience and facilitate a deeper understanding of complex datasets.

g. Comparative Analysis:

- When your project involves comparing different groups or categories, stacked bar charts, grouped bar charts, or parallel coordinates plots can be helpful. These visualizations allow you to compare multiple variables across different categories simultaneously, providing insights into how different factors interact and influence one another. Comparative analysis visualizations are vital for drawing meaningful conclusions about group differences and similarities within the dataset.

4.**Code Integration**

1. Data Cleaning:

Data cleaning involves handling missing values, removing duplicates, correcting inconsistencies, and ensuring data integrity. Python offers powerful libraries like Pandas for data manipulation and cleaning.

Example: Handling Missing Values with Pandas

```
import pandas as pd
# Load your dataset into a Pandas DataFrame
data = pd.read_csv('your_dataset.csv')
# Handling missing values by filling NaN values with mean of the column
data.fillna(data.mean(), inplace=True)
# Removing duplicates
data.drop_duplicates(inplace=True)
# Correcting inconsistencies - Example: converting text to lowercase
data['column_name'] = data['column_name'].str.lower()
# Save the cleaned data back to a CSV file
data.to_csv('cleaned_data.csv', index=False)
```

2. Data Transformation:

Data transformation involves converting data into a suitable format for analysis. This could include normalizing data, encoding categorical variables, or creating new features from existing ones.

Example: Encoding Categorical Variables with Pandas

```
# Assuming 'category_column' is a categorical column in your DataFrame
encoded_data = pd.get_dummies(data, columns=['category_column'])
# Normalizing numerical data (scaling to a range)
```

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data['normalized_column'] = scaler.fit_transform(data[['numerical_column']])
```

3. Statistical Analysis:

Statistical analysis involves deriving meaningful insights from the data, such as calculating summary statistics, correlation, hypothesis testing, etc.

Example: Calculating Summary Statistics and Correlation with Pandas

```
# Summary statistics
summary_stats = data.describe()
# Correlation matrix
correlation_matrix = data.corr()
# Hypothesis testing (Example: t-test for two independent samples)
from scipy.stats import ttest_ind
group1 = data[data['group_column'] == 'group1']['value_column']
group2 = data[data['group_column'] == 'group2']['value_column']
t_stat, p_value = ttest_ind(group1, group2)
```

5.**Conclusion Questions:**

1. What is the average time taken to complete a route, and how does it vary during different times of the day?
2. What percentage of buses/trains operate on time, and how does this performance vary across different routes and days of the week?
3. What are the most common positive aspects mentioned in passenger feedback, and how can these be reinforced or expanded upon?
4. Which routes experience the highest passenger demand, and are they adequately served in terms of frequency and capacity?

5. Are there specific locations or routes with higher reported incidents, requiring increased security measures?

6. Are there trends in fuel consumption or maintenance costs that can be addressed to improve cost-efficiency?

7. How accessible is public transportation for people with disabilities, and what improvements can be made to ensure inclusivity?

8. What is the carbon footprint of public transportation, and how can initiatives like electric or hybrid vehicles be integrated to reduce environmental impact?

9. What is the impact of maintenance work or road closures on transportation schedules, and how can alternative routes be optimized during such events?

10. Based on current trends and feedback, what are the long-term recommendations for infrastructure development, route expansion, or technological integration to enhance the overall public transportation experience?

Innovation

1. Collect transportation data. Public transportation data can be collected from a variety of sources, such as automated passenger counting systems, GPS tracking devices, and passenger surveys. Once we have collected the data, we will need to clean it and prepare it for analysis. This may involve removing outliers, correcting errors, and converting the data into a format that can be easily analyzed.
2. Design relevant visualizations in IBM Cognos. IBM Cognos is a powerful data visualization tool that can be used to create interactive dashboards and reports. Visualizations can help us to identify patterns and trends in the data that would be difficult to see with the naked eye. For example, we could create a visualization that shows the average on-time performance of each bus route.
3. Use code for data analysis. In addition to IBM Cognos, we may also want to use code to perform more complex data analysis tasks. For example, we could write a script to identify the most common passenger complaints.

   Here is a brief explanation of how to implement each step:

   Define analysis objectives:

   For example, we are interested in improving the efficiency of bus routes, Which routes have the longest travel times?

- Which routes have the most stops?
- Which routes are the most crowded?
- Which routes have the highest rates of delays?

Answering these questions can help us to identify areas where the bus service can be improved.

Collect transportation data:

Once we have defined our analysis objectives, we can start to collect the data we need. Public transportation data can be collected from a variety of sources, such as:

- Automated passenger counting systems (APC): APCs are devices that automatically count the number of passengers on a bus or train.
- GPS tracking devices: GPS tracking devices can be used to track the location of buses and trains in real time.
- Passenger surveys: Passenger surveys can be used to collect feedback from passengers about their experience with public transportation.

In addition to these sources, we may also be able to obtain public transportation data from local government agencies or transportation authorities.

Design relevant visualizations in IBM Cognos:

Once we have collected and prepared our data, we can start to design visualizations in IBM Cognos. IBM Cognos offers a variety of visualization types, such as charts, tables, and maps. We can choose the most appropriate visualization type for each question we are trying to answer.

Use code for data analysis:

In addition to IBM Cognos, we may also want to use code to perform more complex data analysis tasks. For example, we could write a script to identify the most common passenger complaints. Or, we could write a script to predict the number of passengers that will use a particular bus route on a given day.

There are a variety of programming languages that can be used for data analysis, such as Python, R, and SQL. The best programming language for we will depend on our skills and experience.

Once we have implemented all of these steps, we will have a comprehensive solution for analyzing public transportation data. This solution can be used to assess service efficiency, on-time performance, and passenger feedback. The insights gained from the analysis can be used to support transportation improvement initiatives and enhance the overall public transportation experience.

Data loading:

Data loading for public transport analysis is a crucial initial step in the data analysis pipeline. This process involves the collection, cleaning, integration, transformation, and loading of data for further analysis.

Data collection begins by gathering information from diverse sources, including transit agencies, GPS trackers, ticketing systems, and open data portals. This data encompasses details about routes, stops, schedules, ticket sales, and vehicle tracking.

The subsequent step involves data cleaning, where datasets are scrutinized for duplicates, missing values, and outliers. This cleaning process ensures that the data is consistent and accurate, providing a reliable foundation for analysis.

Data integration may be necessary to merge data from various sources to create a comprehensive dataset. This step is essential for understanding the complete public transport network.

Data transformation is performed to prepare the data for analysis. This often includes aggregating data by time intervals (e.g., hourly or daily), geospatial analysis, and feature engineering to make it suitable for analytical tools and techniques.

Importance of loading and processing dataset:

Loading and preprocessing the dataset is an important first step in building any machine learning model. However, it is especially important for vaccine analysis, as the datasets are often complex and noisy. By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and accurately.

Missing Data

Another common issue that we face in real-world data is the absence of data points. Most machine learning models can’t handle missing values in the data, so you need to intervene and adjust the data to be properly used inside the model.

Scaling the features:

It is often helpful to scale the features before training a machine learning model. This can help to improve the performance of the model and make it more robust to outliers. There are a variety of ways to scale the features, such as min-max scaling and standard scaling.

1.Loading the dataset

Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.

Identify the dataset:

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

Load the Dataset:

Load your dataset into a Pandas DataFrame**.** The quality and reliability of data can significantly impact the outcomes of vaccine analysis, making it imperative to have robust data loading procedures in place.

Program:

```
import pandas as pd
import numpy as np
import csv
import matplotlib.pyplot as plt
route_details=pd.read_csv("/content/20140711.csv",quoting=csv.QUOTE_NONE)
route_details= route_details.sample(n=50,replace=True)
route_details.describe()
```

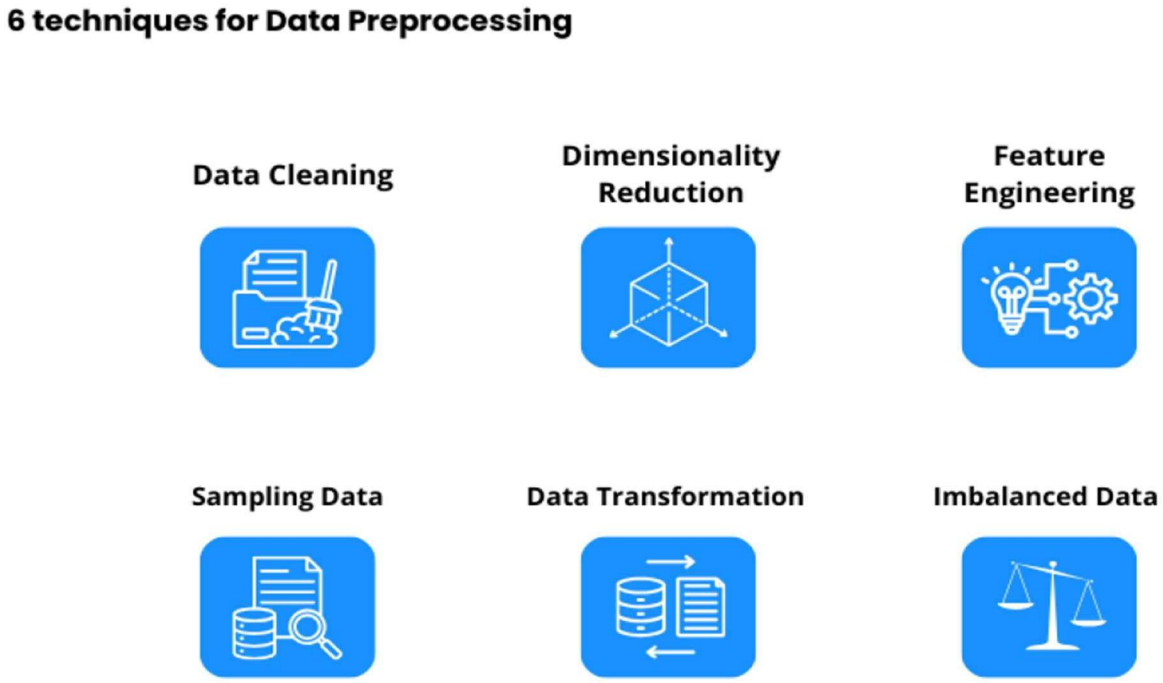
Output:

1 to 4 of 4 entries Filter

index	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
count	50	50	50	50	50	50
unique	23	1	27	23	9	7
top	"44702"	"100"	"12404"	"176 Cross Rd"	"2013-08-18 00:00:00"	"1"
freq	5	50	4	5	14	18

Preprocess the dataset:

Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format.



Data cleaning:

This involves identifying and correcting errors and inconsistencies in the data. For example, this may involve removing duplicate records, correcting typos, and filling in missing values.

Feature Scaling:

Normalize or standardize numerical features to bring them to a common scale. Common methods include Min-Max scaling (scaling features to a specific range) and z-score normalization (scaling features to have a mean of 0 and a standard deviation of 1).

Feature Engineering:

Create new features or modify existing ones to capture more meaningful information from the data. This may involve mathematical transformations, interaction terms, or aggregations.

Data transformation:

It is a critical aspect of data preprocessing that involves converting and modifying the data to make it more suitable for analysis. It can help improve the performance of machine

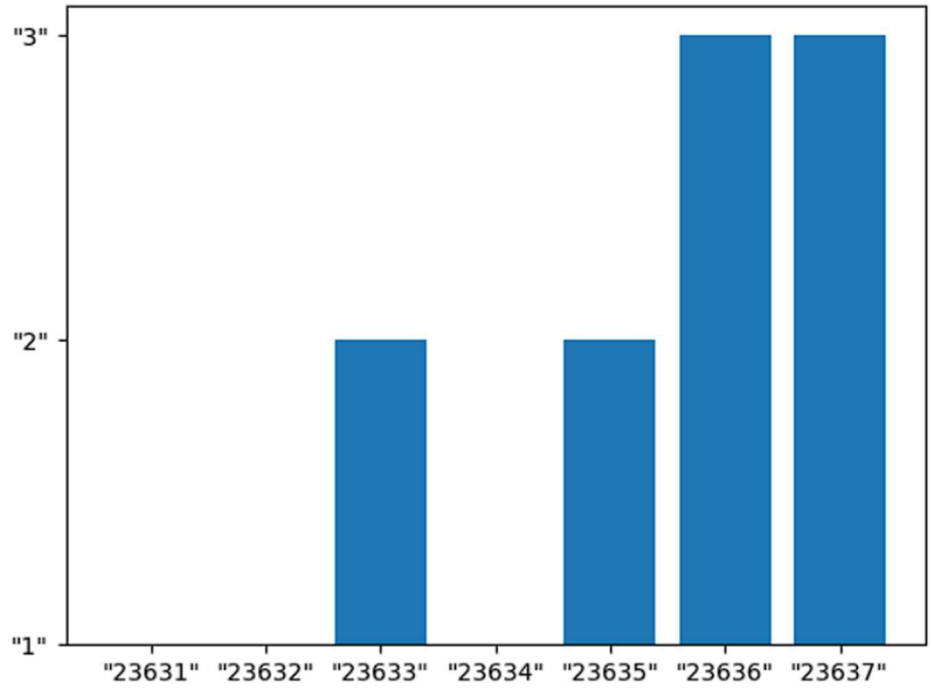
learning models, enhance the interpretability of the data, and ensure that it aligns with the assumptions of certain statistical techniques.

Exploring the data:

Program:

```
chart1=pd.read_csv("/content/20140711.csv",quoting=csv.QUOTE_NONE,nrows=25)
xaxix = chart1['TripID']
yaxix = chart1['NumberOfBoardings']
plt.bar(x=xaxix,height=yaxix)
plt.show()
```

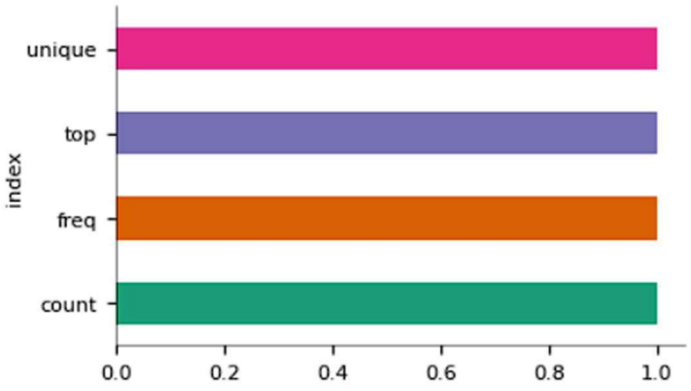
Output:

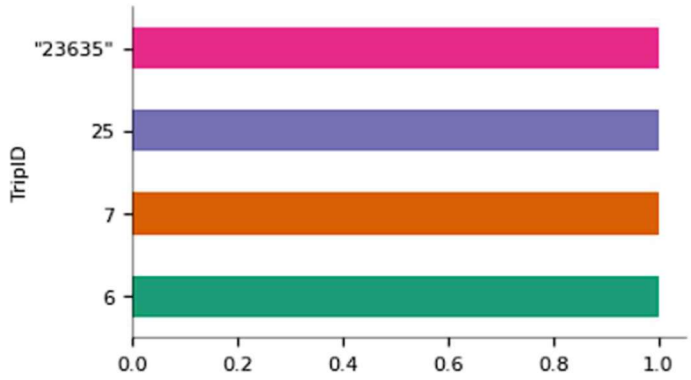


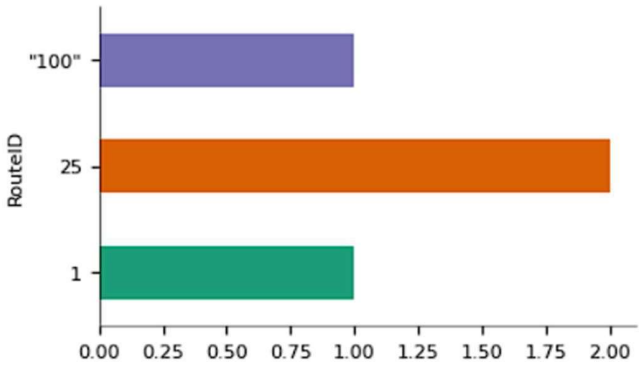
Program:

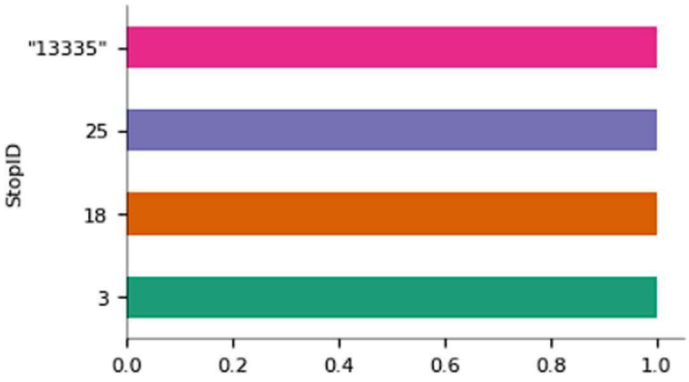
chart1.describe()

Output:

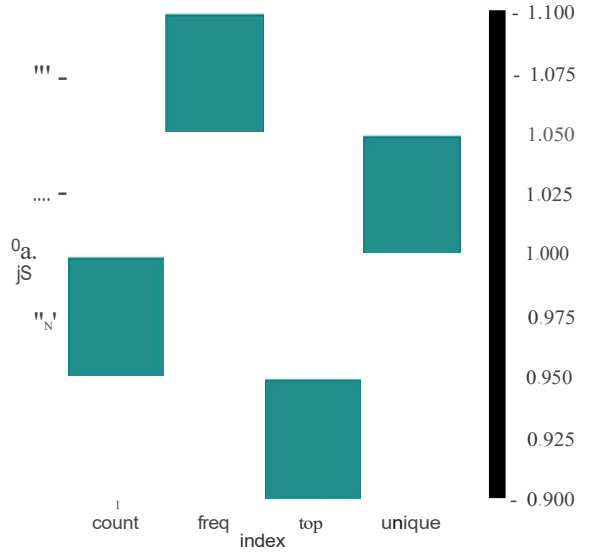


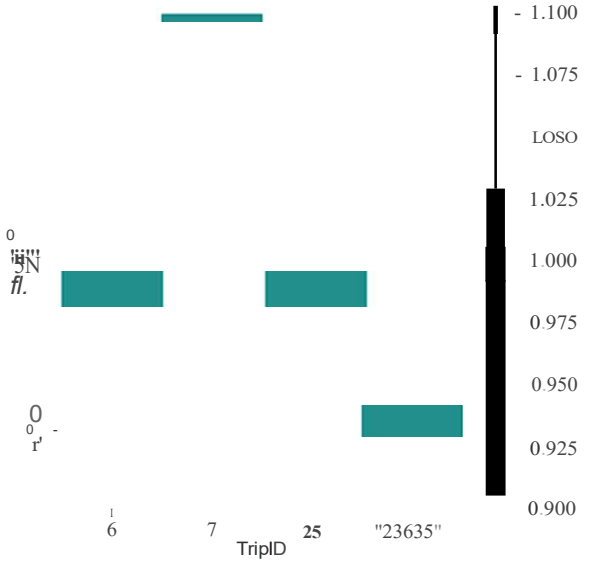


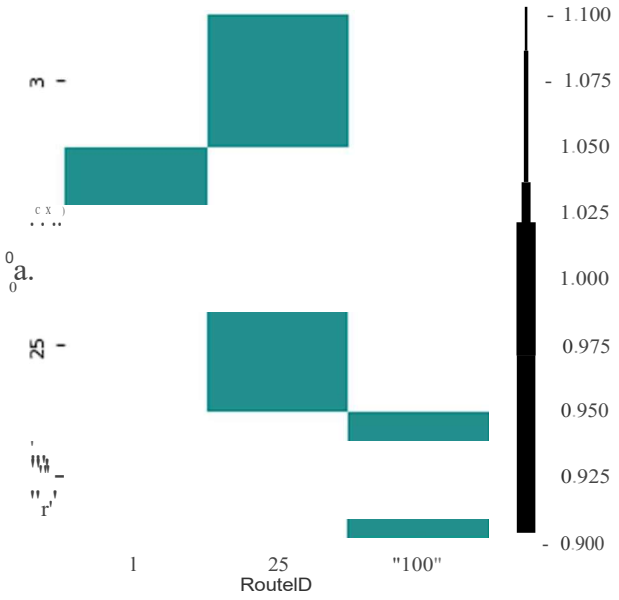


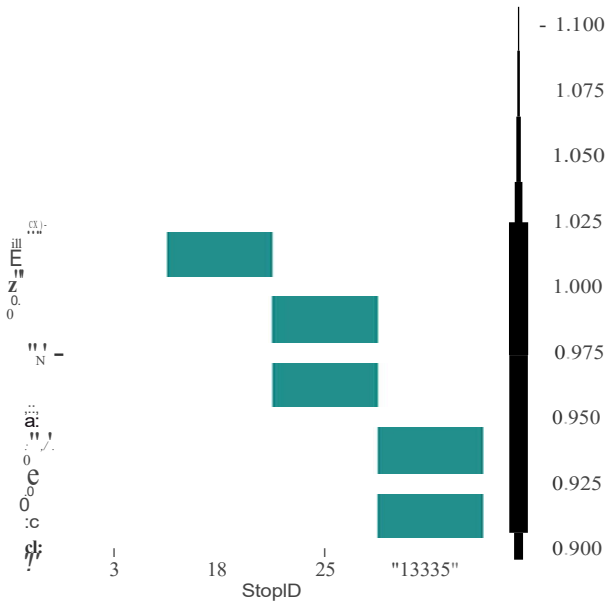


2-d categorical distributions





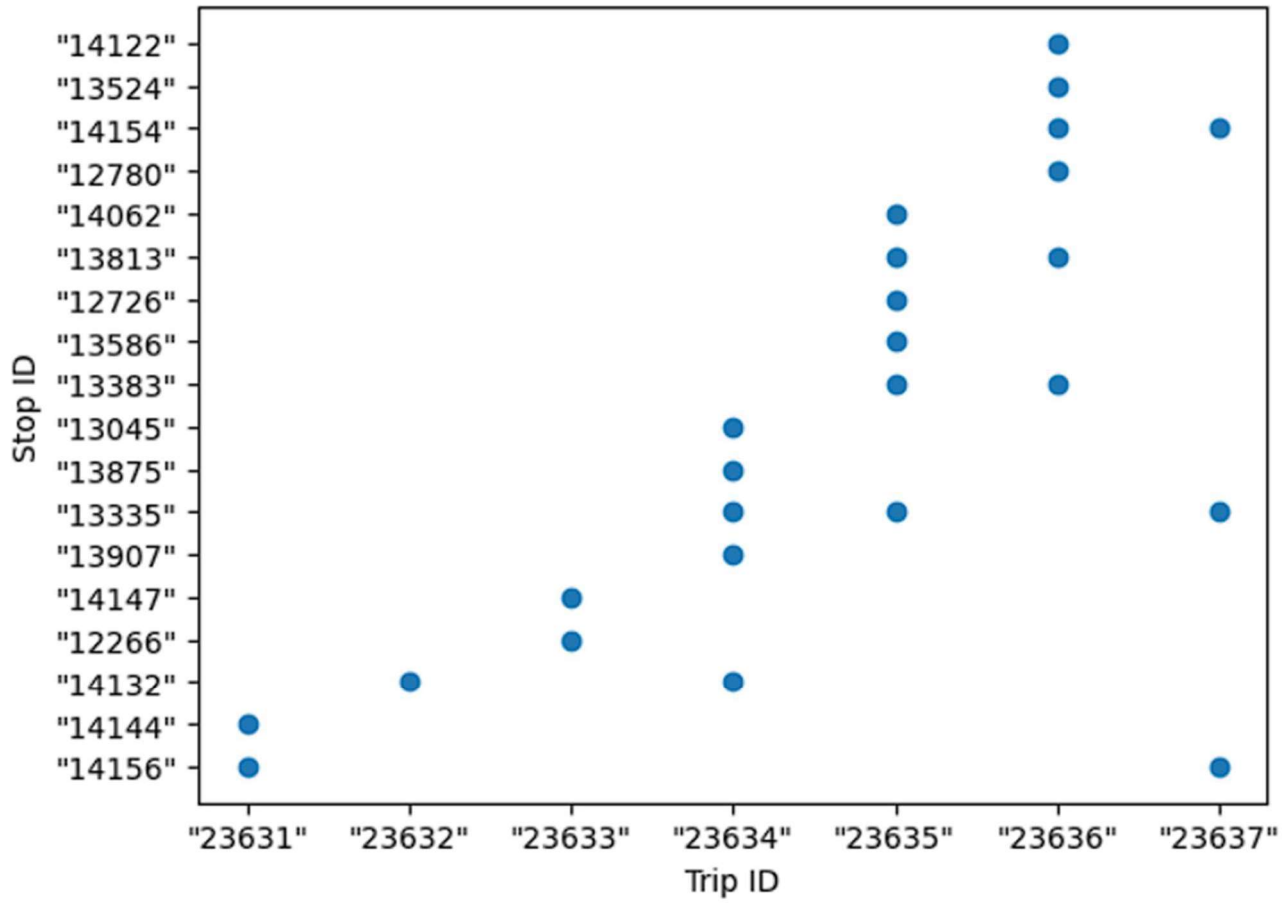




Program:

```
xaxix = chart1['TripID']
yaxix = chart1['StopID']
plt.scatter(xaxix,yaxix)
plt.xlabel("Trip ID")
plt.ylabel("Stop ID")
plt.show()
```

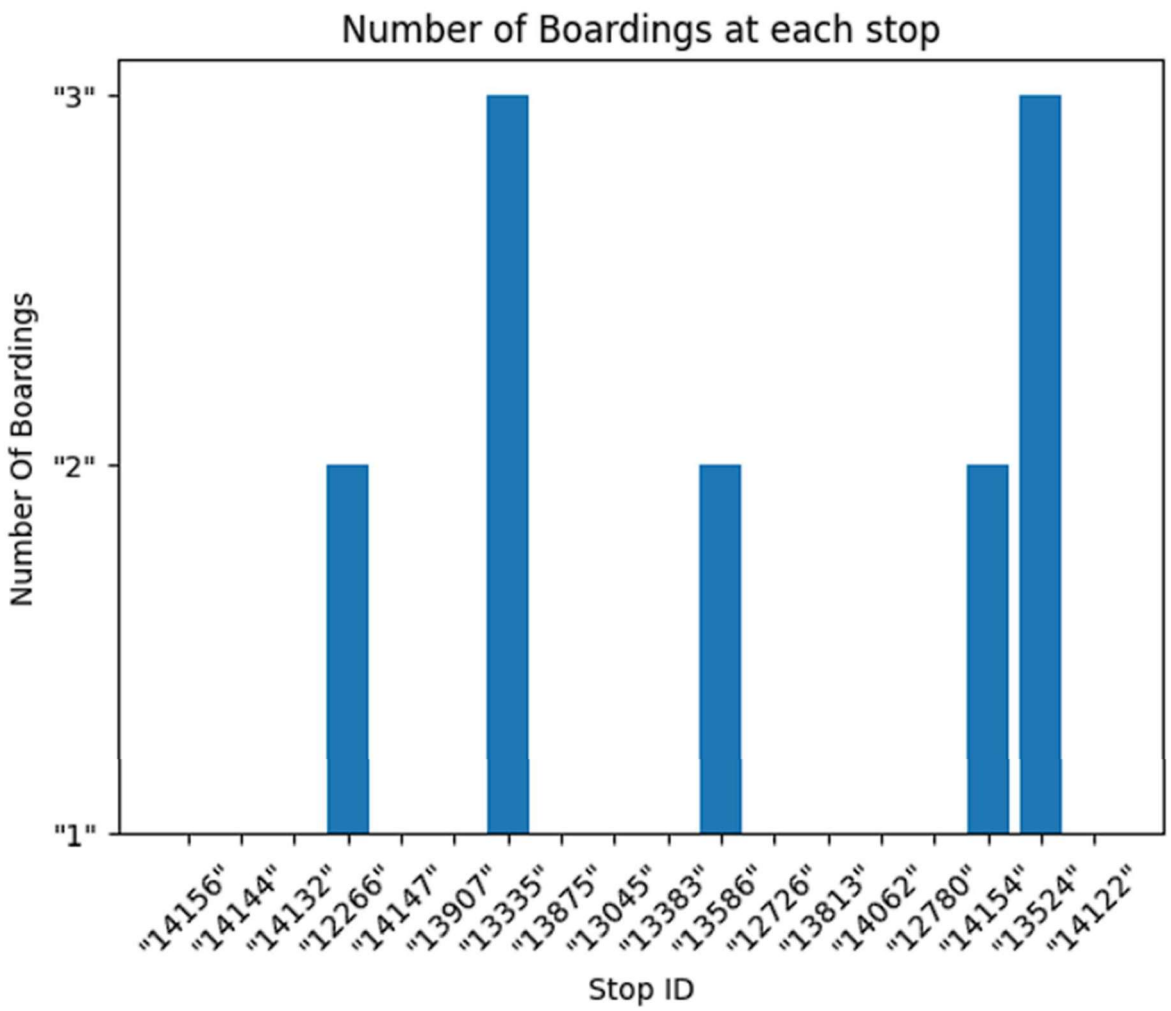
Output:



Program:

```
xaxix = chart1['StopID']
yaxix = chart1['NumberOfBoardings']
plt.bar(xaxix,yaxix)
plt.title("Number of Boardings at each stop")
plt.xlabel("Stop ID")
plt.ylabel("Number Of Boardings")
plt.xticks(rotation=45)
plt.show()
```

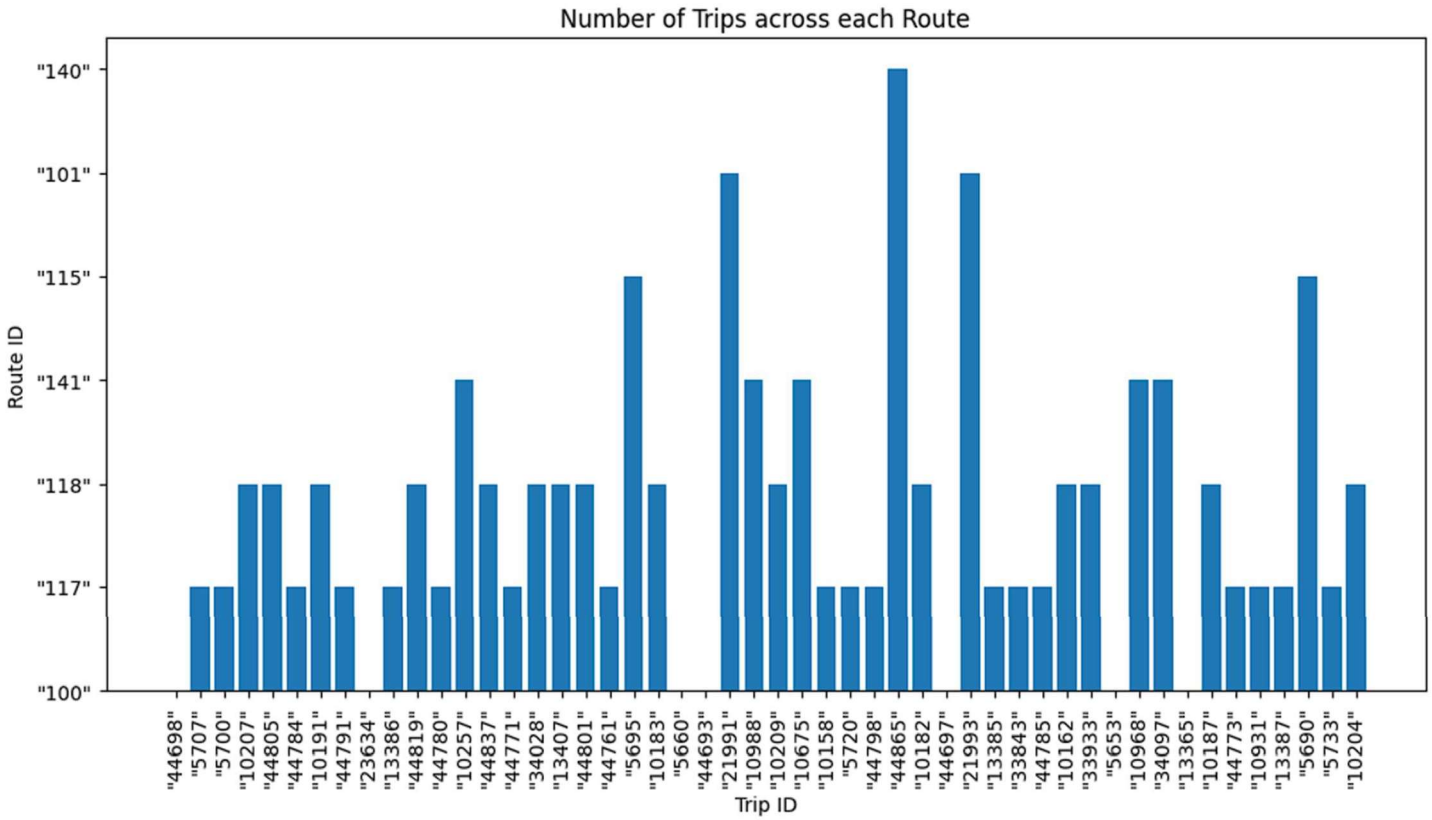
Output:



Program:

```
yaxix = route_details['RouteID']
xaxix = route_details['TripID']
plt.figure(figsize=(12,6))
plt.bar(xaxix,yaxix)
plt.title("Number of Trips across each Route")
plt.xlabel("Trip ID")
plt.ylabel("Route  ID")
plt.xticks(rotation=90)
plt.show()
```

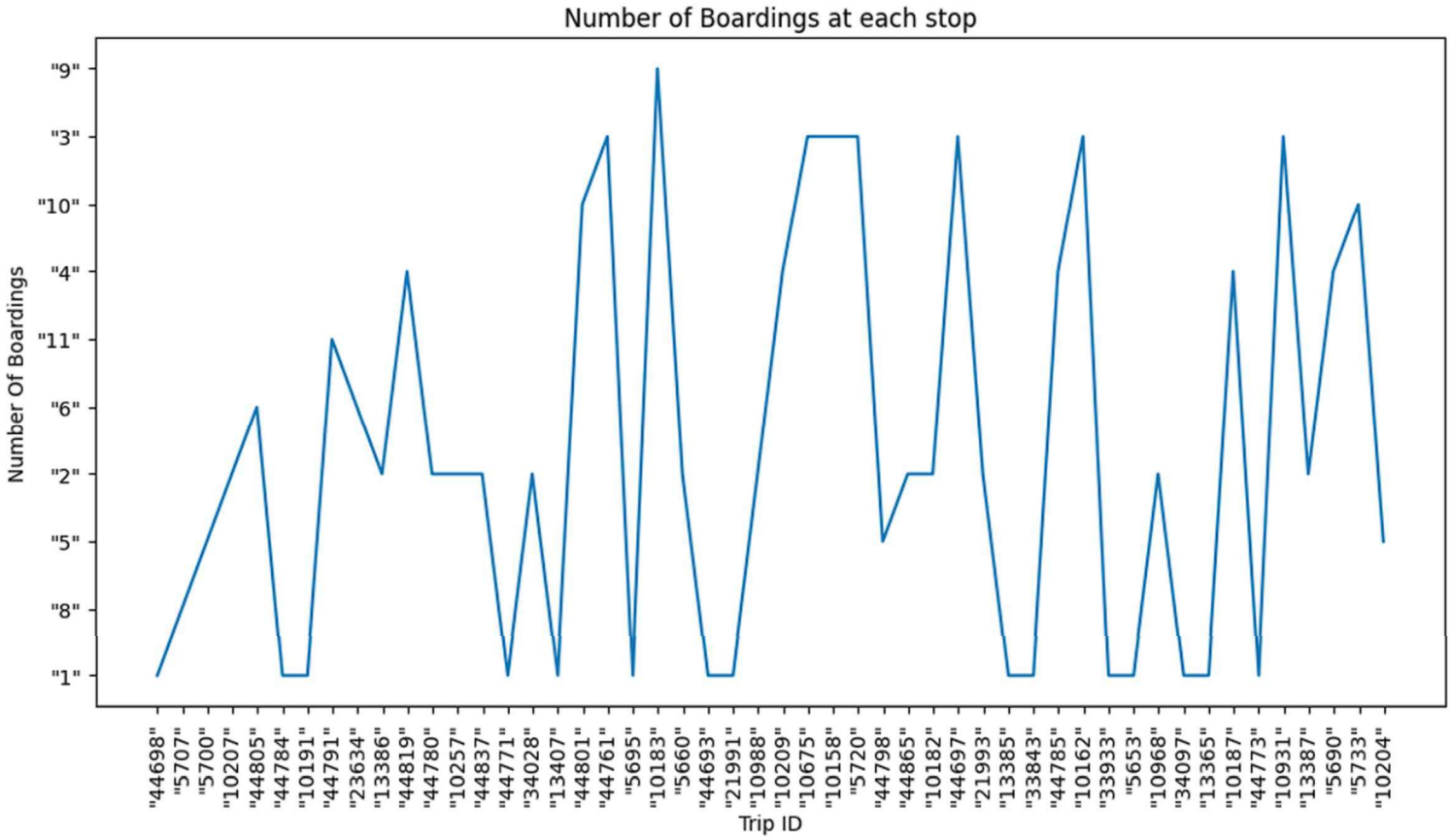
Output:



Program:

```
xaxix = route_details['TripID']
yaxix = route_details['NumberOfBoardings']
plt.figure(figsize=(12,6))
plt.plot(xaxix,yaxix)
plt.title("Number of Boardings at each stop")
plt.xlabel("Trip ID")
plt.ylabel("Number Of Boardings")
plt.xticks(rotation=90)
plt.show()
```

Output:



Importance of data Loading:

Data loading and preprocessing for public transport efficiency analysis serve as the critical initial steps that empower researchers and government traffic department to make informed decisions about traffic situation, daily routes, rush routes, and safety. The quality, accuracy, and suitability of the data at this stage are pivotal in determining the success of subsequent analyses. Through diligent and systematic data handling, we can harness the power of data-driven insights to address public traffic challenges and contribute to the betterment of public safety.

Visualization in IBM Cognos:

Creating a dashboard in IBM Cognos involves several steps, including designing the layout, adding data visualizations, and configuring interactivity. Here's a step-by-step process to create a dashboard in IBM Cognos:

Access IBM Cognos:

- Log in to your IBM Cognos environment using your credentials.

Launch Dashboard Application:

- Once logged in, launch the IBM Cognos Dashboard application.

Create a New Dashboard:

- Start by creating a new dashboard by selecting the "Create Dashboard" option or similar, depending on your version of Cognos.

Choose a Layout:

- Select a layout for your dashboard. Layout options typically include templates with different grid configurations.

Add Data Sources:

- To populate your dashboard with data, you need to add data sources. Common data sources include databases, spreadsheets, or web services.
- Click on the "Add Data" or "Data Source" option.
- Connect to your data source and configure the connection settings.

Design the Layout:

- Customize the layout of your dashboard by dragging and dropping containers, text, and images onto the canvas.
- Arrange components to create the desired structure and style.

Add Widgets:

- To display data, add widgets to your dashboard. Widgets can be charts, tables, text boxes, or other data visualizations.
- Select a container or section, and then add a widget.
- Configure the widget to use the data source you added earlier.

Configure Widgets:

- Customize each widget's settings, including data source, display options, and interactions. This may involve setting filters, sorting, and aggregation.

Apply Filters and Interactivity:

- Enable interactive features like drill-throughs, filtering, and parameterized queries to allow users to explore the data dynamically.
- Configure filter controls and actions to refine data based on user interactions.

Add Prompts (Optional):

- If your dashboard requires user input, you can add prompts. Prompts allow users to select parameters for data filtering.
- Configure prompts to be dynamic and responsive to user input.

Test the Dashboard:

- Before publishing the dashboard, thoroughly test its functionality. Ensure that all widgets display data correctly and that interactivity functions as expected.

Apply Styling:

- Customize the visual style of your dashboard to match your organization's branding and design guidelines.

Save and Publish:

- Save your dashboard project and, when ready, publish it to a location accessible by your target audience. This can be within the Cognos portal or other web applications.

Share and Distribute:

- Share the dashboard with the intended audience by providing them with the necessary access rights or links to the published dashboard.

Maintain and Update:

- Regularly maintain and update the dashboard to ensure that it reflects the most current data and meets the evolving needs of your users.

IBM Cognos offers a range of data visualization tools to help users create compelling and insightful visualizations in their reports and dashboards. These visualization tools allow users to present data in various graphical formats, making it easier to understand and analyze. Here are some of the visualization tools available in IBM Cognos:

1. Charts:

IBM Cognos provides a wide variety of chart types, including bar charts, line charts, pie charts, scatter plots, and more. You can choose the most suitable chart type for your data to convey your message effectively.

2. Tables:

Tabular data is a fundamental visualization tool in Cognos. You can create tables to display data in rows and columns. Tables are highly customizable, allowing you to show data in various ways, such as cross-tabs, lists, and pivot tables.

3. Crosstabs:

Crosstabs are used to display data in a tabular format with rows and columns. They are particularly useful for summarizing data and showing relationships between two or more data dimensions.

4. Maps:

IBM Cognos provides mapping capabilities for geospatial data visualization. You can create maps to display geographic data, including regions, countries, and custom geographic areas. Maps can be interactive and used for drilling down into location-based data.

5. Pivot Tables and Grids:

Pivot tables and grids are tools for multidimensional analysis. Users can interact with data by dragging and dropping dimensions and measures to explore data from different angles.

6. Scorecard:

Scorecards allow you to visualize key performance indicators (KPIs) and metrics using various graphical indicators such as gauges, traffic lights, and progress bars. They provide at-a-glance insights into performance.

7. Advanced Visualizations:

IBM Cognos also supports more advanced visualizations like waterfall charts, heat maps, tree maps, and box plots. These visualizations are suitable for in-depth data analysis and storytelling.

8. Custom Visualizations:

Users can create custom visualizations using JavaScript and D3.js in IBM Cognos. This allows for the creation of unique and specialized visualizations tailored to specific needs.

9. Interactive Dashboards:

Dashboards in IBM Cognos enable you to combine multiple visualizations into a single canvas. You can create interactive dashboards with charts, tables, maps, and other components, and add interactivity like filtering and drill-through actions.

10. Annotations:

Annotations let you add text and graphical elements to your visualizations to provide context and explanations to the viewers.

11. Data Exploration Tools:

IBM Cognos offers data exploration tools like the IBM Cognos Exploration with Smart Data modules, which allow you to explore and visualize data in an intuitive and self-service manner.

12. Active Reports:

Active Reports allow you to create interactive, offline reports that can be used without an internet connection. Users can explore the data and make selections even when disconnected from the Cognos environment.

13. Storytelling:

IBM Cognos Analytics provides storytelling features that allow you to create narratives with visualizations, annotations, and text to guide users through the data and communicate insights effectively.

These visualization tools in IBM Cognos help users design and present data in ways that are informative and actionable, enabling better decision-making and data-driven insights. The choice of visualization depends on the nature of the data and the audience's needs.

Line Graph:

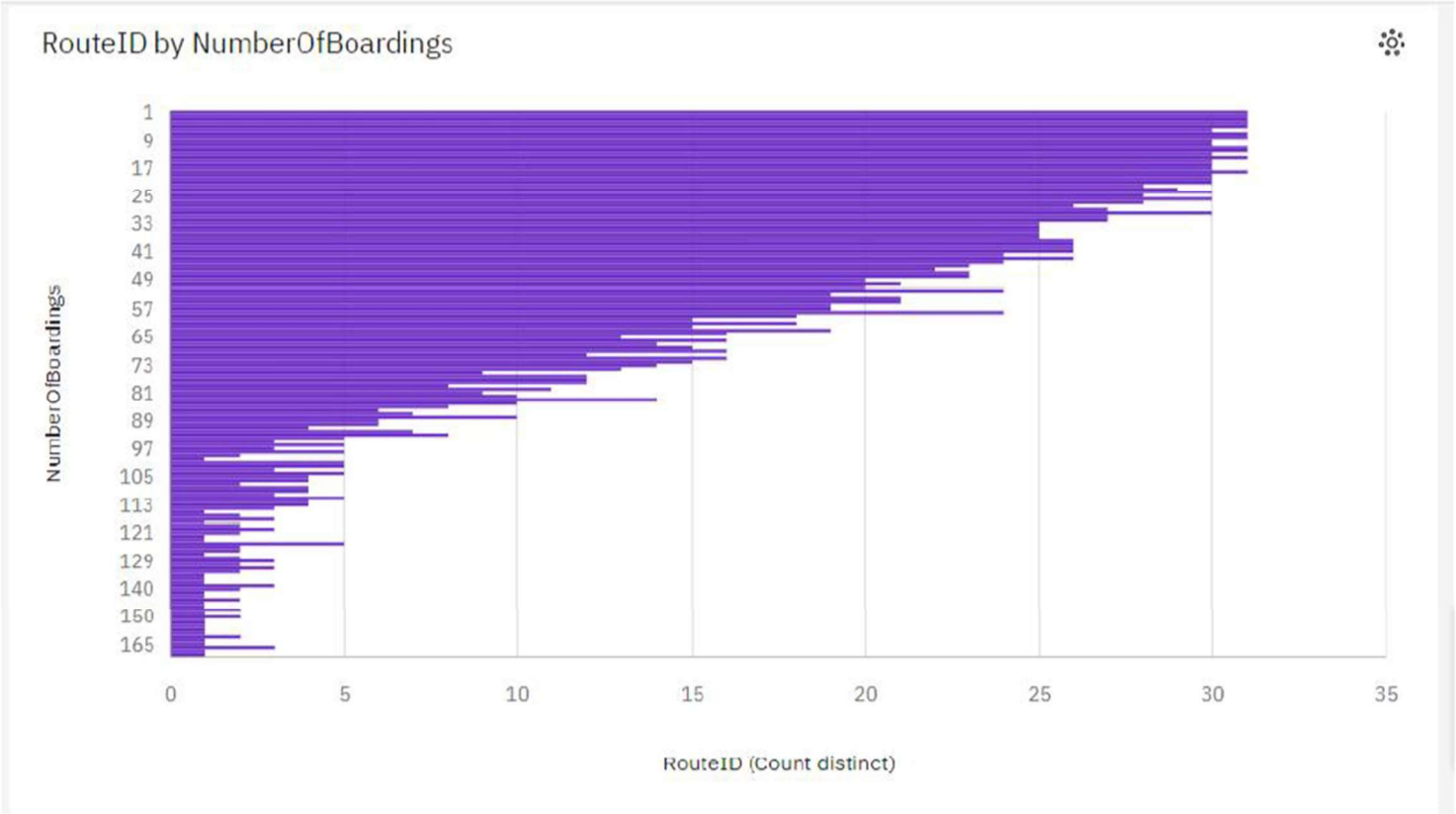
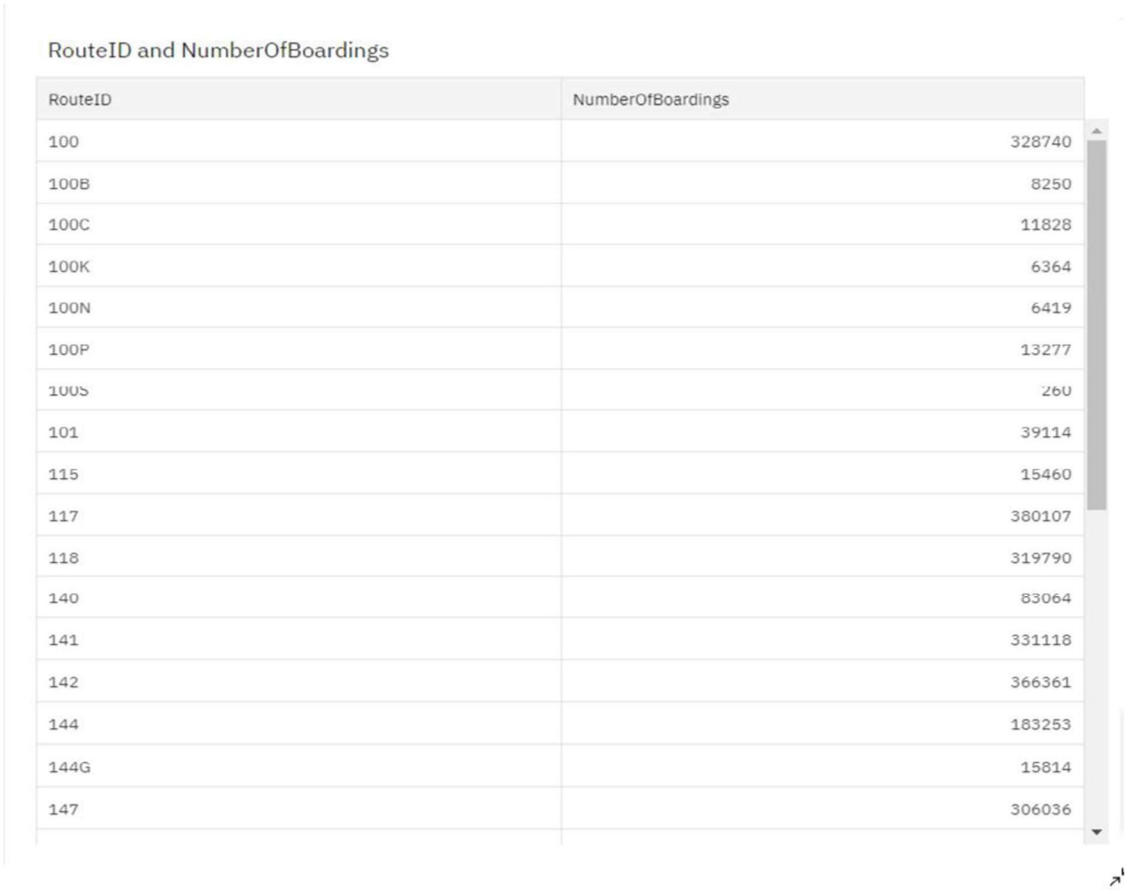


Table Graph:



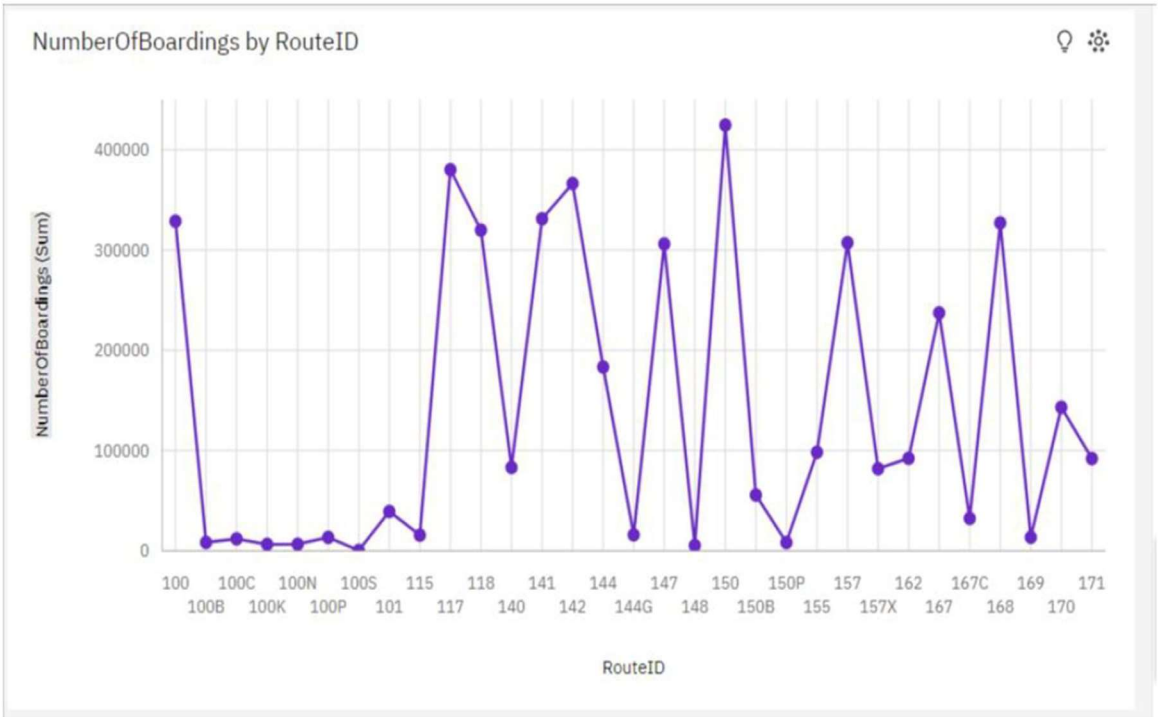
RouteID and NumberOfBoardings

RouteID	NumberOfBoardings
100	328740
100B	8250
100C	11828
100K	6364
100N	6419
100P	13277
100S	260
101	39114
115	15460
117	380107
118	319790
140	83064
141	331118
142	366361
144	183253
144G	15814
147	306036

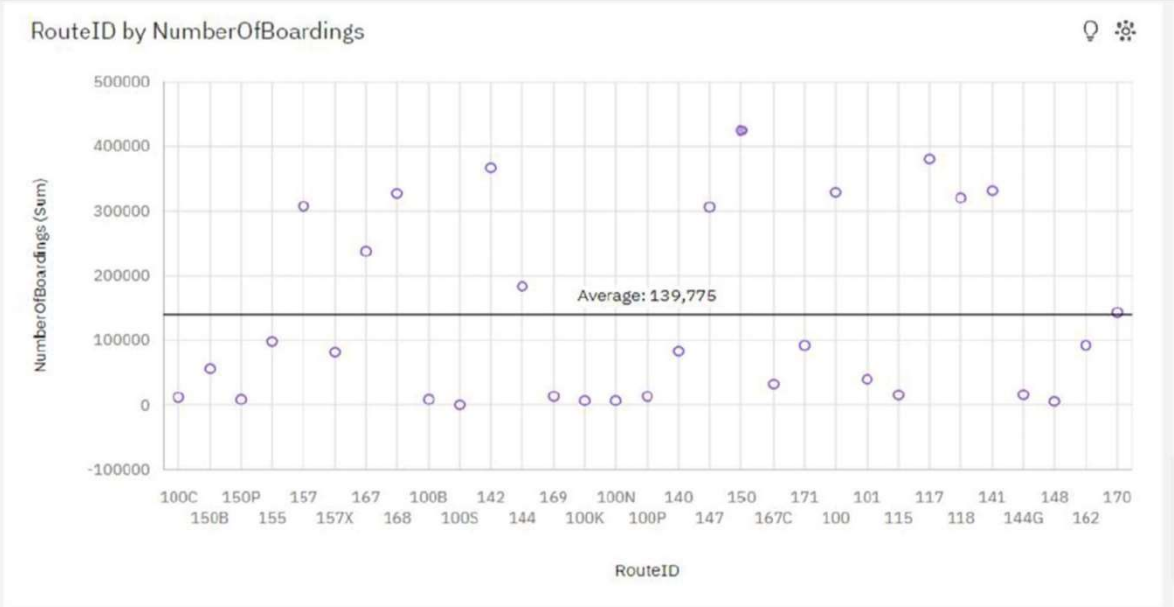
RouteID and NumberOfBoardings

RouteID	NumberOfBoardings
144G	15814
147	306036
148	5190
150	424625
150B	55517
150P	8147
155	98191
157	307301
157X	81745
162	92171
167	237236
167C	32195
168	327057
169	13397
170	143076
171	91911
summary	4333016

Line Graph:



Scatter Plot:



Conclusion:

In conclusion, the analysis of public transport efficiency reveals that it is a multifaceted and dynamic system that plays a crucial role in urban mobility. Efficiency in public transport is not solely about minimizing costs but also about maximizing benefits for passengers, the environment, and society as a whole.

It is evident that efficient public transport systems offer numerous advantages, such as reduced traffic congestion, lower greenhouse gas emissions, increased accessibility, and improved quality of life for residents. However, achieving optimal efficiency requires ongoing evaluation, adaptation, and collaboration among stakeholders, including government authorities, transit agencies, and the public.

In a rapidly urbanizing world, addressing the challenges of public transport efficiency is vital for creating sustainable, inclusive, and resilient cities. It is clear that continued efforts and innovation in this area are essential for the betterment of urban transportation systems and the well-being of communities worldwide.