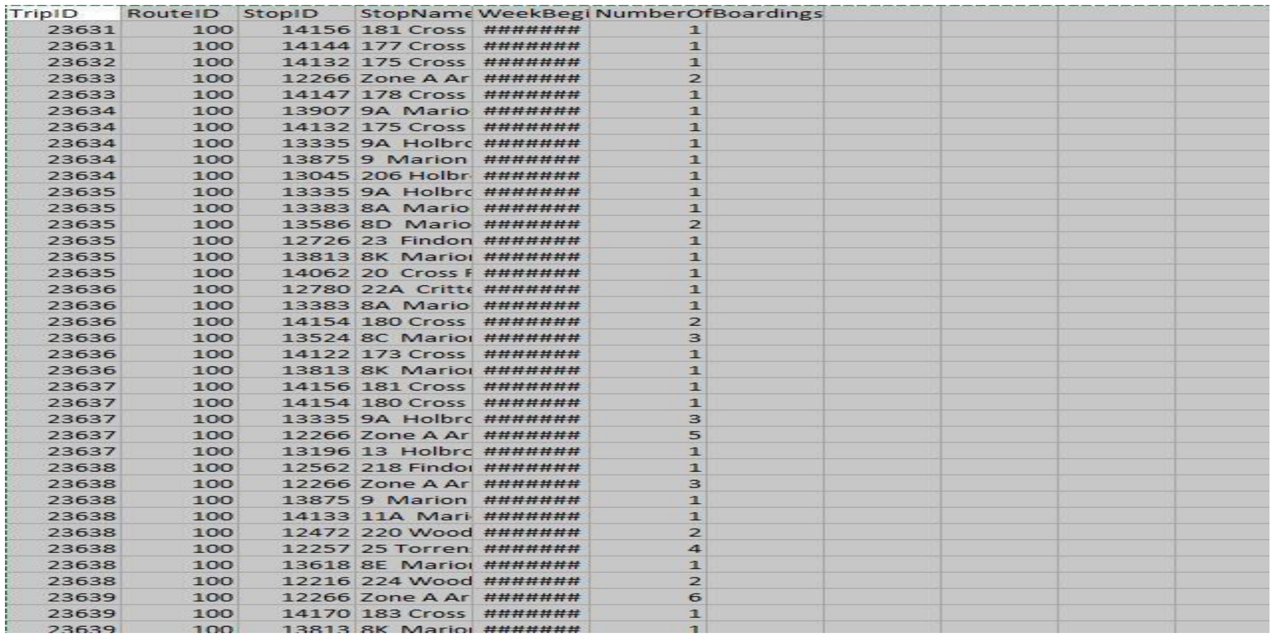
Project Title: Public Transport Analysis

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.



TripID	RouteID	StopID	StopName	WeekBegi	NumberOfBoardings
23631	100	14156	181 Cross	########	1
23631	100	14144	177 Cross	########	1
23632	100	14132	175 Cross	########	1
23633	100	12266	Zone A Ar	########	2
23633	100	14147	178 Cross	########	1
23634	100	13907	9A Mario	########	1
23634	100	14132	175 Cross	########	1
23634	100	13335	9A Holbro	########	1
23634	100	13875	9 Marion	########	1
23634	100	13045	206 Holbr	########	1
23635	100	13335	9A Holbro	########	1
23635	100	13383	8A Mario	########	1
23635	100	13586	8D Mario	########	2
23635	100	12726	23 Findon	########	1
23635	100	13813	8K Marior	########	1
23635	100	14062	20 Cross F	########	1
23636	100	12780	22A Critte	########	1
23636	100	13383	8A Mario	########	1
23636	100	14154	180 Cross	########	2
23636	100	13524	8C Marior	########	3
23636	100	14122	173 Cross	########	1
23636	100	13813	8K Marior	########	1
23637	100	14156	181 Cross	########	1
23637	100	14154	180 Cross	########	1
23637	100	13335	9A Holbro	########	3
23637	100	12266	Zone A Ar	########	5
23637	100	13196	13 Holbro	########	1
23638	100	12562	218 Findor	########	1
23638	100	12266	Zone A Ar	########	3
23638	100	13875	9 Marion	########	1
23638	100	14133	11A Mari	########	1
23638	100	12472	220 Wood	########	2
23638	100	12257	25 Torren	########	4
23638	100	13618	8E Marior	########	1
23638	100	12216	224 Wood	########	2
23639	100	12266	Zone A Ar	########	6
23639	100	14170	183 Cross	########	1
23639	100	13813	8K Marior	########	1

Design Thinking:

Based on the provided information, here's a structured approach to address the project objectives:

1. Project Objectives:

On-Time Performance Assessment:

Measure and evaluate the punctuality of public transportation services, including buses, trains, and trams.
Identify patterns of delays and their causes, such as traffic congestion, maintenance issues, or operational inefficiencies.
Set benchmarks for on-time performance and develop strategies to improve reliability.

Passenger Satisfaction Analysis:

Gather and analyze passenger feedback through surveys, focus groups, or online reviews.
Assess the factors that contribute to passenger satisfaction, such as cleanliness, safety, courtesy of staff, and comfort.
Use insights to implement improvements, enhance the overall passenger experience, and retain or attract riders.

Service Efficiency Optimization:

Evaluate the efficiency of public transport services, considering factors like route optimization, vehicle utilization, and energy consumption.
Identify areas where resources can be allocated more effectively to reduce costs and environmental impact.
Implement changes to schedules, routes, and maintenance procedures to enhance service efficiency and sustainability.

2. Data collection :

Schedule Data:

Official Timetables: Obtain official schedules from transportation agencies or providers. These timetables are typically available online or in printed format and provide planned departure and arrival times for various routes.

GTFS (General Transit Feed Specification): Many transit agencies publish GTFS data, a standardized format that includes schedule information. This data can be accessed through data feeds and APIs, making it suitable for analysis and real-time updates.

Archived Data: Historical schedule data can be collected from previous years to analyze trends and performance over time. These archives can help identify long-term patterns and areas for improvement.

Real-Time Updates:

GPS and AVL Systems: Many modern public transportation vehicles are equipped with GPS (Global Positioning System) and AVL (Automatic Vehicle Location) systems. These systems provide real-time location and movement data, which can be accessed through APIs or data feeds to track vehicle positions and predict arrival times accurately.

Mobile Apps and Websites: Public transportation providers often have mobile apps and websites that offer real-time updates to passengers. Data from these sources can be scraped or accessed through official APIs to provide real-time information on vehicle locations and delays.

Crowdsourced Data: Some mobile apps and platforms collect real-time data from passengers' smartphones. Passengers willingly or unknowingly contribute data on vehicle locations and travel times, which can be aggregated for real-time updates.

Passenger Feedback:

Surveys and Questionnaires: Transportation agencies can conduct surveys and distribute questionnaires to collect feedback from passengers. These surveys may be paper-based or conducted online through the agency's website or mobile apps.

Social Media Monitoring: Monitor social media platforms like Twitter, Facebook, and Instagram for passenger feedback and complaints. Passengers often share their experiences and concerns, providing valuable insights into service quality.

Customer Service and Complaint Logs: Data can be collected from customer service centers and complaint logs. Analyzing these logs can help identify recurring issues and areas for improvement based on direct feedback from passengers.

3. Visualization Strategy:

Step 1: Data Preparation

Before creating dashboards and reports, ensure that your data is properly prepared. This includes data cleaning, transformation, and integration from various sources. Use IBM Cognos to connect to your data sources and create a data model for easy visualization.

Step 2: Define Dashboard Objectives

Clearly define the objectives of your dashboard. What insights are you trying to convey? Common objectives for public transportation analysis might include on-time performance, passenger satisfaction, and service efficiency. Consider the key performance indicators (KPIs) that matter most.

Step 3: Identify Key Metrics

Determine the key metrics and data points that will be included in your dashboard. For example, you might include metrics like on-time performance percentages, customer satisfaction scores, route efficiency data, and more.

Step 4: Design the Dashboard

Design the layout and components of your dashboard. IBM Cognos offers a range of visualization options, including charts, tables, maps, and more. Consider the following components:
KPI Cards: Display important metrics in a summarized format.
Charts and Graphs: Use bar charts, line graphs, and pie charts to visualize trends and comparisons.
Maps: Show route performance and passenger density using geographic data.
Tables: Present detailed data in tabular format.
Filters and Prompts: Allow users to interact with the data by applying filters and prompts to focus on specific aspects.

Step 5: Create Reports

Reports can provide detailed insights and are often used in conjunction with dashboards. Use IBM Cognos to generate reports that support the insights presented in your dashboards. These reports can be in the form of PDFs, Excel sheets, or other formats.

4. Code Integration:

Data Cleaning:

Automated Data Validation: Code can be used to implement data validation checks, flagging or correcting inconsistencies, missing values, and outliers in the dataset.
Standardization and Cleaning Rules: Develop code to apply standardization and cleaning rules consistently across large datasets, ensuring data uniformity.
Data Deduplication: Code can automatically identify and remove duplicate records, reducing errors and redundancy in the data.

Data Transformation:

Feature Engineering: Code can create new features or variables based on existing data, providing richer insights into passenger behavior, route efficiency, and performance metrics.
Aggregation and Summarization: Code allows for efficient aggregation and summarization of data, enabling analyses at different levels (e.g., daily, weekly, or by geographic region).
Geospatial Analysis: Use code to process and analyze geospatial data, such as calculating distances between stops, mapping routes, or identifying service gaps.

Statistical Analysis:

Hypothesis Testing: Code can automate hypothesis tests to assess the significance of factors affecting public transportation, such as weather conditions, service interruptions, or demographics.
Regression Analysis: Implement code for regression analyses to understand the relationships between variables, such as how changes in service frequency impact ridership.
Time Series Analysis: Code can facilitate time series analysis to uncover patterns, seasonality, and trends in transportation data, helping with forecasting and decision-making.

Development part:

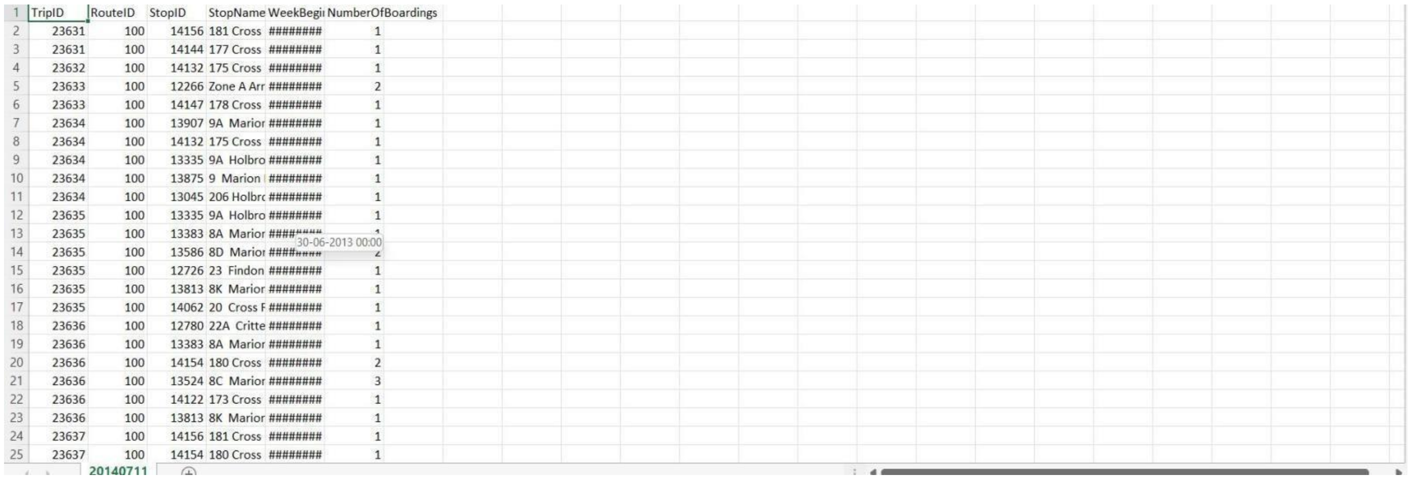
Data loading:

Load public transportation data into analysis tools for evaluating efficiency and making informed decisions for improved transportation services

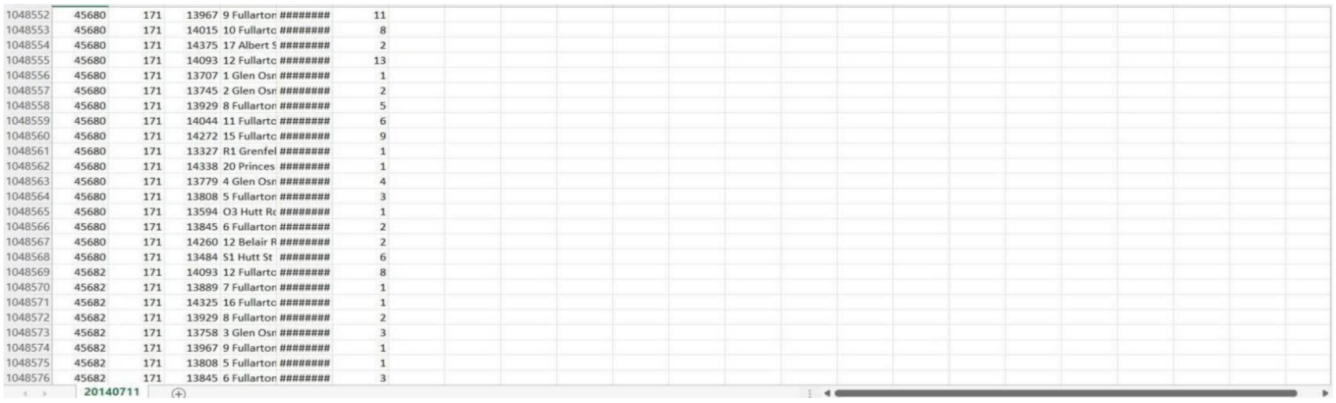
Data Preprocessing:

In data preprocessing for public transport efficiency analysis, the collected data will undergo a series of steps, including handling missing values, outliers, and inconsistencies. This process also involves data integration, where multiple sources are merged into a unified dataset. Data transformations and feature engineering will be applied to make the data suitable for analysis. Quality assurance procedures will ensure data integrity, laying the foundation for insightful analysis and visualization within IBM Cognos.

Given data set:



1	TripID	RouteID	StopID	StopName	WeekBegi	NumberOfBoardings
2	23631	100	14156	181 Cross	########	1
3	23631	100	14144	177 Cross	########	1
4	23632	100	14132	175 Cross	########	1
5	23633	100	12266	Zone A Arr	########	2
6	23633	100	14147	178 Cross	########	1
7	23634	100	13907	9A Marior	########	1
8	23634	100	14132	175 Cross	########	1
9	23634	100	13335	9A Holbro	########	1
10	23634	100	13875	9 Marion	########	1
11	23634	100	13045	206 Holbrc	########	1
12	23635	100	13335	9A Holbro	########	1
13	23635	100	13383	8A Marior	######## 30-06-2013 00:00	1
14	23635	100	13586	8D Marior	########	2
15	23635	100	12726	23 Findon	########	1
16	23635	100	13813	8K Marior	########	1
17	23635	100	14062	20 Cross F	########	1
18	23636	100	12780	22A Critte	########	1
19	23636	100	13383	8A Marior	########	1
20	23636	100	14154	180 Cross	########	2
21	23636	100	13524	8C Marior	########	3
22	23636	100	14122	173 Cross	########	1
23	23636	100	13813	8K Marior	########	1
24	23637	100	14156	181 Cross	########	1
25	23637	100	14154	180 Cross	########	1



1048552	45680	171	13967	9 Fullarton	########	11
1048553	45680	171	14015	10 Fullartc	########	8
1048554	45680	171	14375	17 Albert S	########	2
1048555	45680	171	14093	12 Fullartc	########	13
1048556	45680	171	13707	1 Glen Osn	########	1
1048557	45680	171	13745	2 Glen Osn	########	2
1048558	45680	171	13929	8 Fullarton	########	5
1048559	45680	171	14044	11 Fullartc	########	6
1048560	45680	171	14272	15 Fullartc	########	9
1048561	45680	171	13327	R1 Grenfel	########	1
1048562	45680	171	14338	20 Princes	########	1
1048563	45680	171	13779	4 Glen Osn	########	4
1048564	45680	171	13808	5 Fullarton	########	3
1048565	45680	171	13594	O3 Hutt Rc	########	1
1048566	45680	171	13845	6 Fullarton	########	2
1048567	45680	171	14260	12 Belair R	########	2
1048568	45680	171	13484	S1 Hutt St	########	6
1048569	45682	171	14093	12 Fullartc	########	8
1048570	45682	171	13889	7 Fullarton	########	1
1048571	45682	171	14325	16 Fullartc	########	1
1048572	45682	171	13929	8 Fullarton	########	2
1048573	45682	171	13758	3 Glen Osn	########	3
1048574	45682	171	13967	9 Fullarton	########	1
1048575	45682	171	13808	5 Fullarton	########	1
1048576	45682	171	13845	6 Fullarton	########	3

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 transport 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.

Challenges involved in loading and preprocessing transport efficiency analysis dataset;

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.

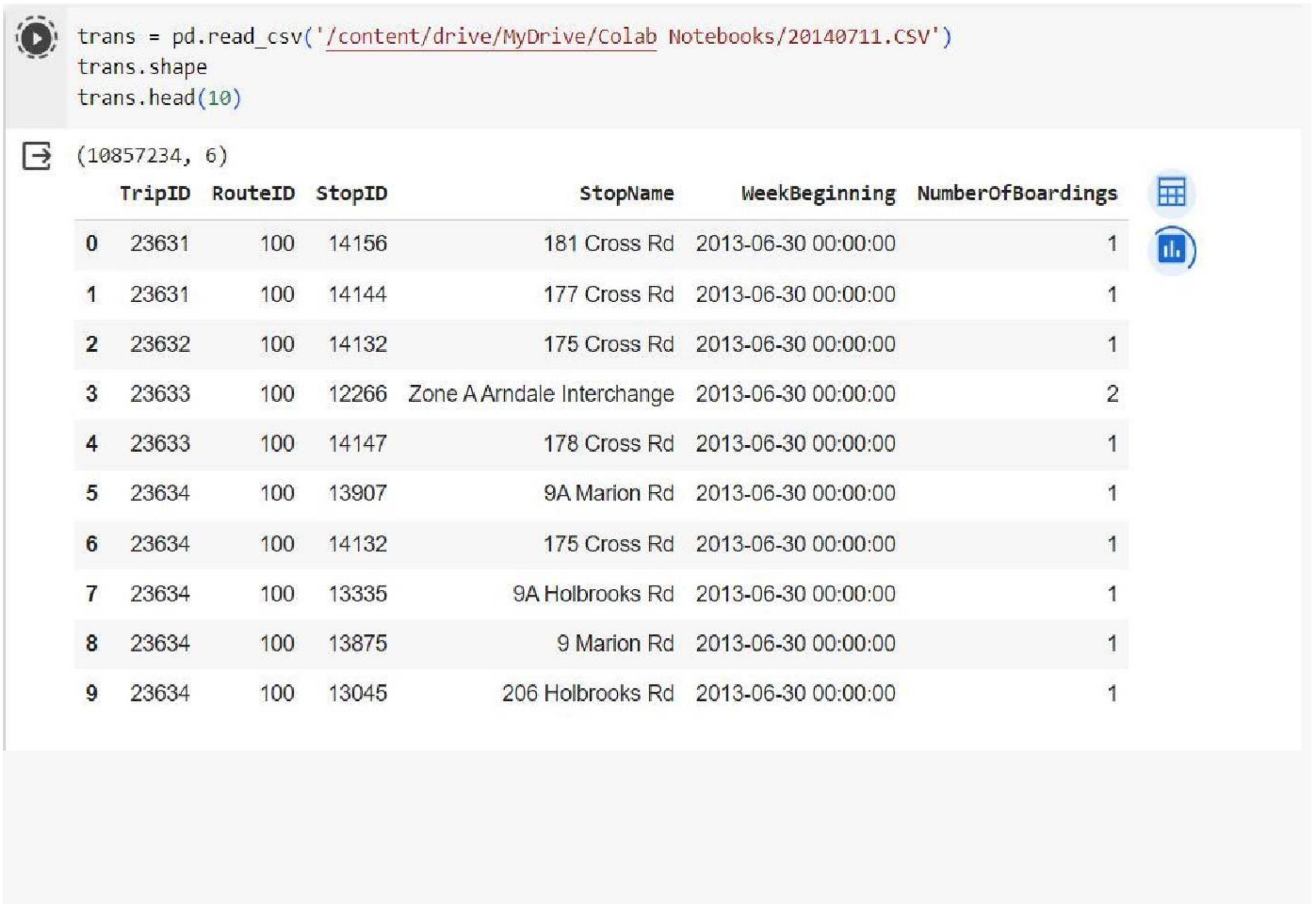
1. Identify the dataset:

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.

2. 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:



```
trans = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/20140711.CSV')
trans.shape
trans.head(10)
```

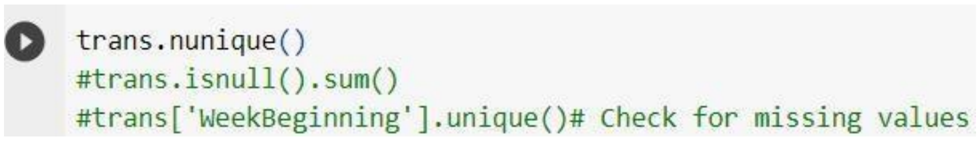
```
(10857234, 6)
```

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings
0	23631	100	14156	181 Cross Rd	2013-06-30 00:00:00	1
1	23631	100	14144	177 Cross Rd	2013-06-30 00:00:00	1
2	23632	100	14132	175 Cross Rd	2013-06-30 00:00:00	1
3	23633	100	12266	Zone A Arndale Interchange	2013-06-30 00:00:00	2
4	23633	100	14147	178 Cross Rd	2013-06-30 00:00:00	1
5	23634	100	13907	9A Marion Rd	2013-06-30 00:00:00	1
6	23634	100	14132	175 Cross Rd	2013-06-30 00:00:00	1
7	23634	100	13335	9A Holbrooks Rd	2013-06-30 00:00:00	1
8	23634	100	13875	9 Marion Rd	2013-06-30 00:00:00	1
9	23634	100	13045	206 Holbrooks Rd	2013-06-30 00:00:00	1

3. Exploring data:

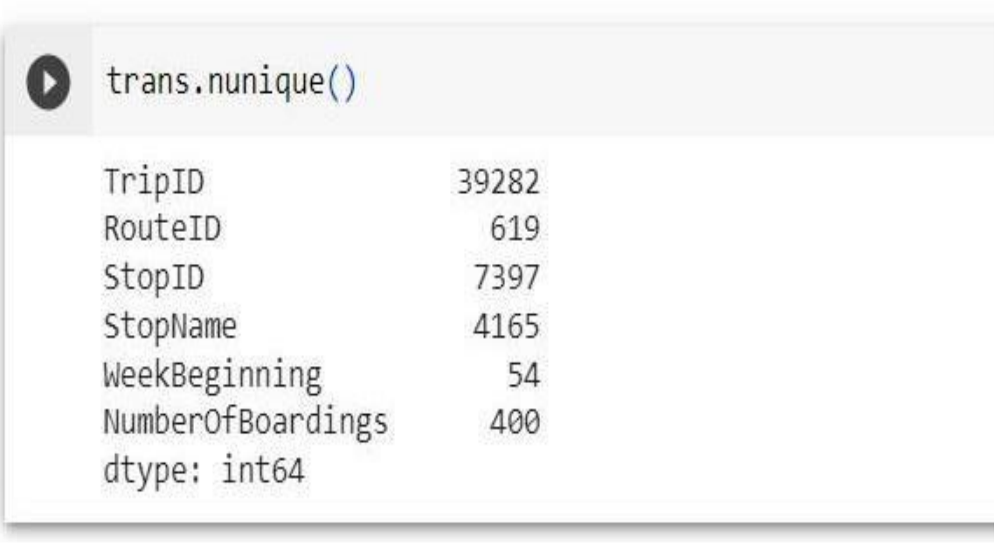
Perform EDA to understand your data better. This includes checking for missing values, exploring the data's statistics, and visualizing it to identify patterns.

Program:



```
trans.nunique()
#trans.isnull().sum()
#trans['WeekBeginning'].unique()# Check for missing values
```

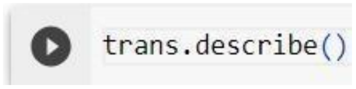
Output



```
trans.nunique()
```

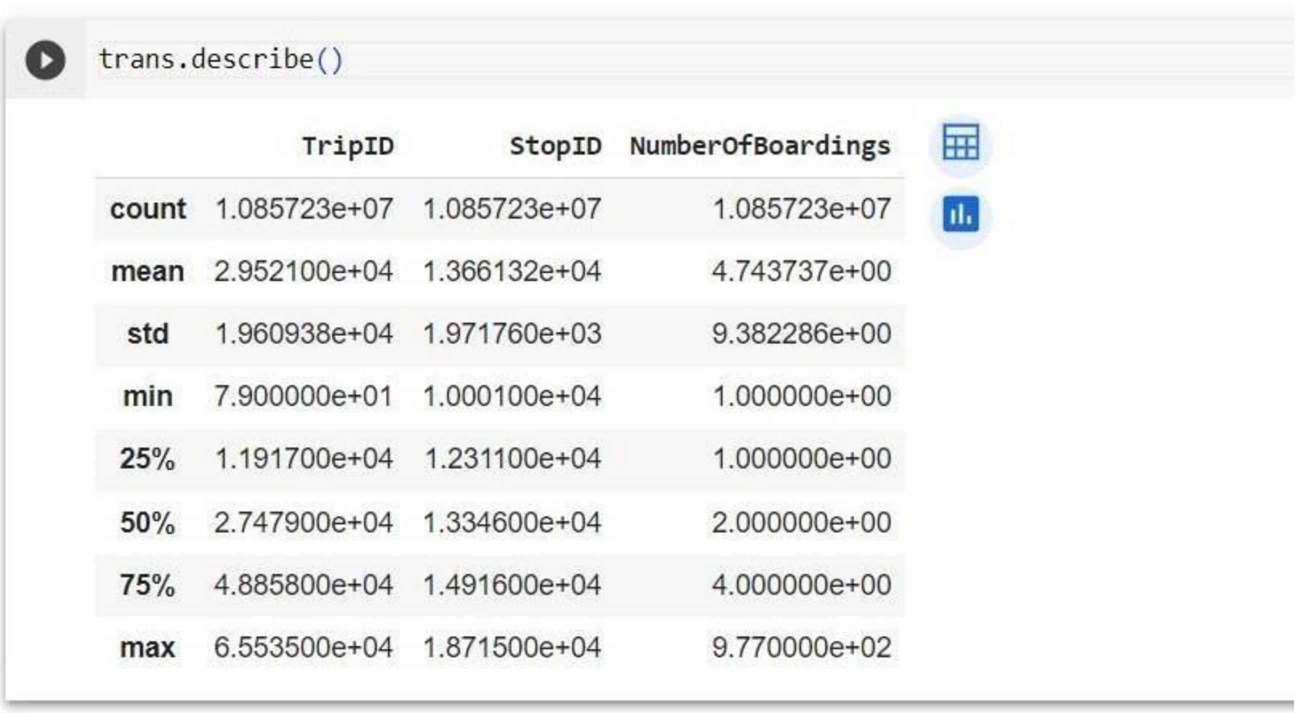
```
TripID               39282
RouteID                619
StopID                7397
StopName              4165
WeekBeginning           54
NumberOfBoardings      400
dtype: int64
```

Program:



```
trans.describe()
```

Output

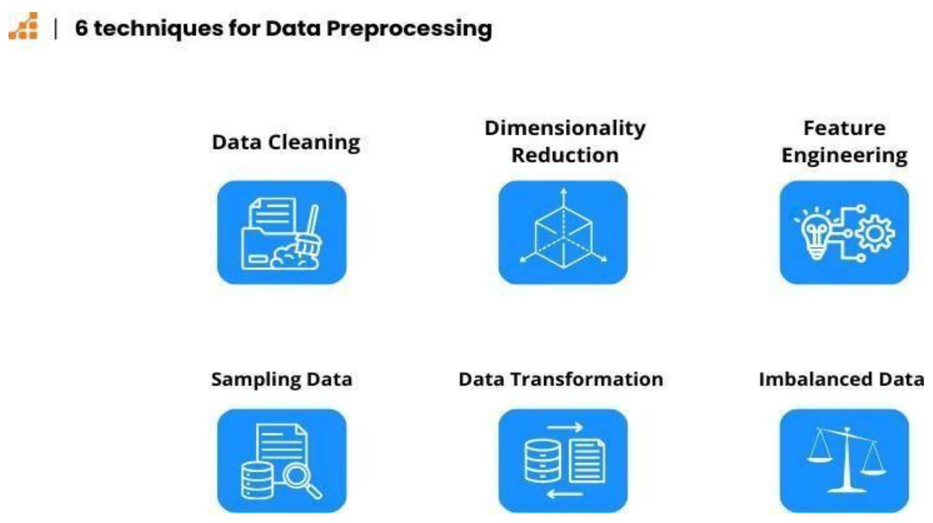


```
trans.describe()
```

	TripID	StopID	NumberOfBoardings
count	1.085723e+07	1.085723e+07	1.085723e+07
mean	2.952100e+04	1.366132e+04	4.743737e+00
std	1.960938e+04	1.971760e+03	9.382286e+00
min	7.900000e+01	1.000100e+04	1.000000e+00
25%	1.191700e+04	1.231100e+04	1.000000e+00
50%	2.747900e+04	1.334600e+04	2.000000e+00
75%	4.885800e+04	1.491600e+04	4.000000e+00
max	6.553500e+04	1.871500e+04	9.770000e+02

3. 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).

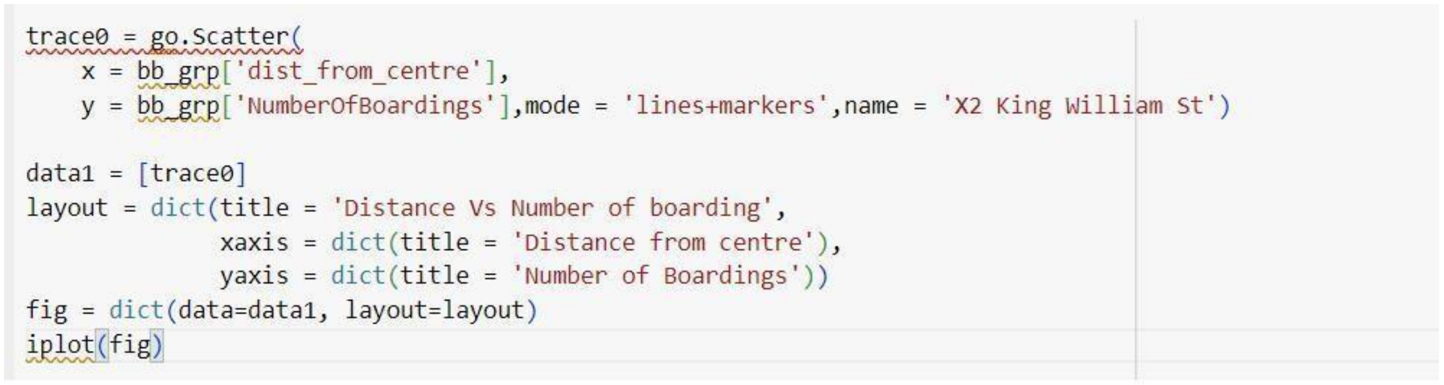
Feature Engineering: Create new features or modify existing ones to capture more meaningful information from the data. This may involve mathematical transformations, interaction terms, or aggregations.

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.

Begin the public transportation efficiency analysis project by loading and preprocessing the dataset. Define objectives, gather transportation data from the source, then rigorously process and cleanse the data for quality and accuracy.

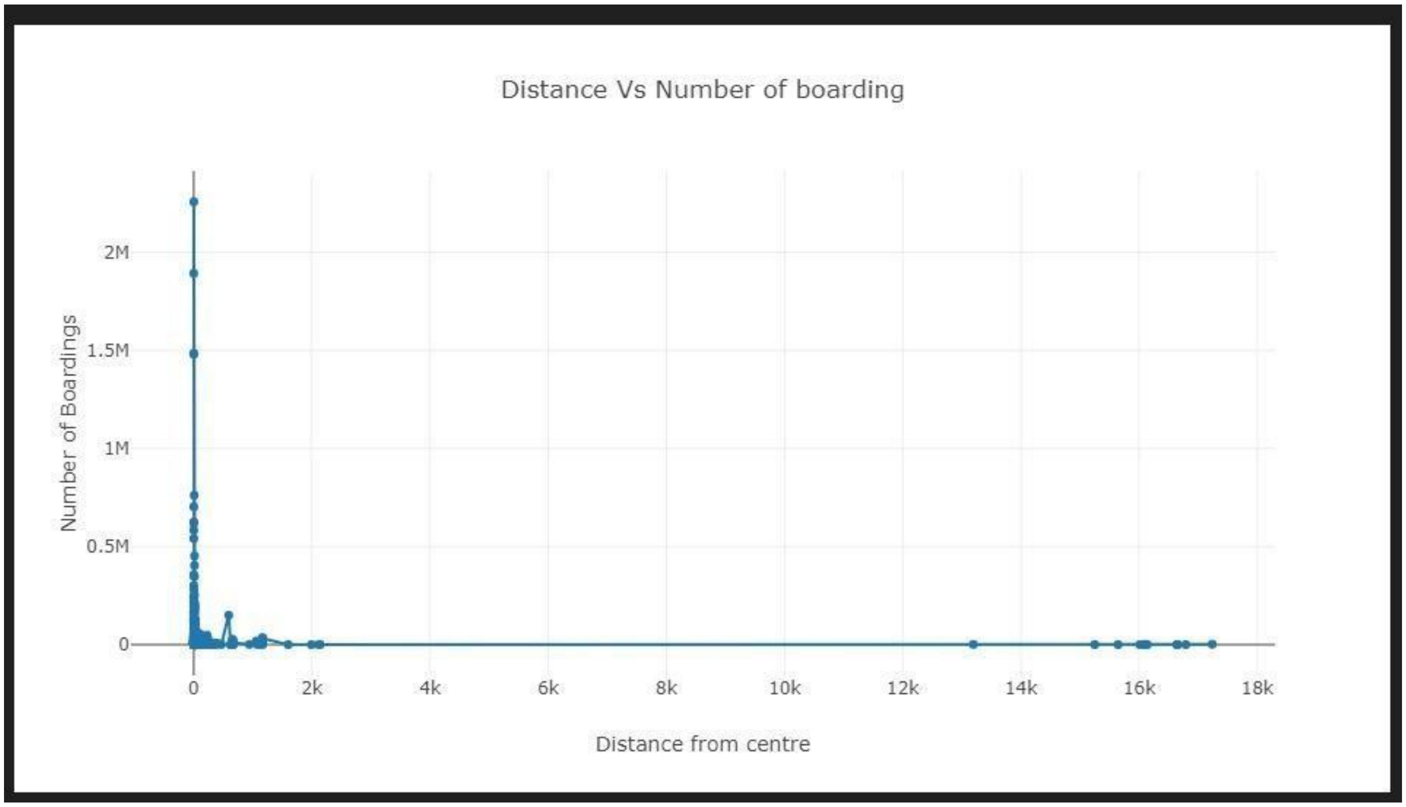
Data visualization:

Program:



```
trace0 = go.Scatter(
    x = bb_grp['dist_from_centre'],
    y = bb_grp['NumberOfBoardings'],mode = 'lines+markers',name = 'X2 King William St')

data1 = [trace0]
layout = dict(title = 'Distance Vs Number of boarding',
              xaxis = dict(title = 'Distance from centre'),
              yaxis = dict(title = 'Number of Boardings'))
fig = dict(data=data1, layout=layout)
iplot(fig)
```

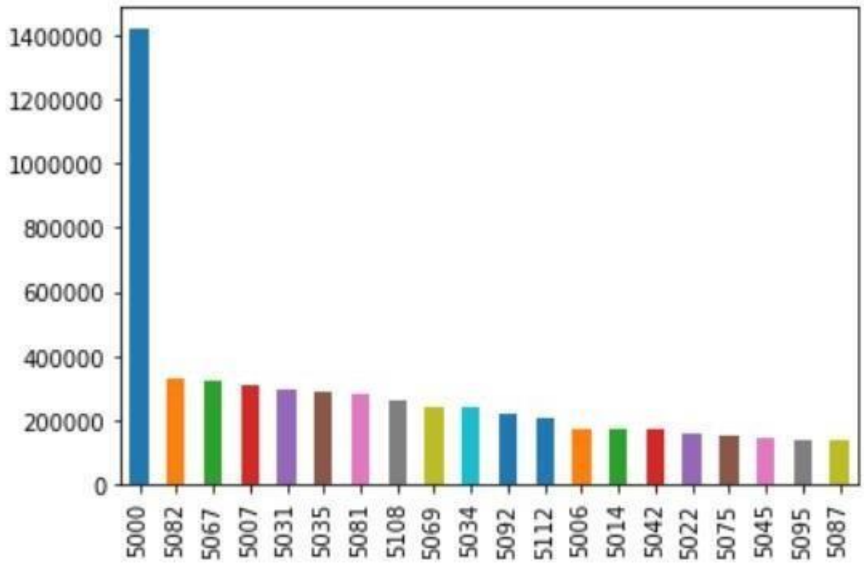
Output:



Program:

```
data['postcode'].value_counts().head(20).plot.bar()
```

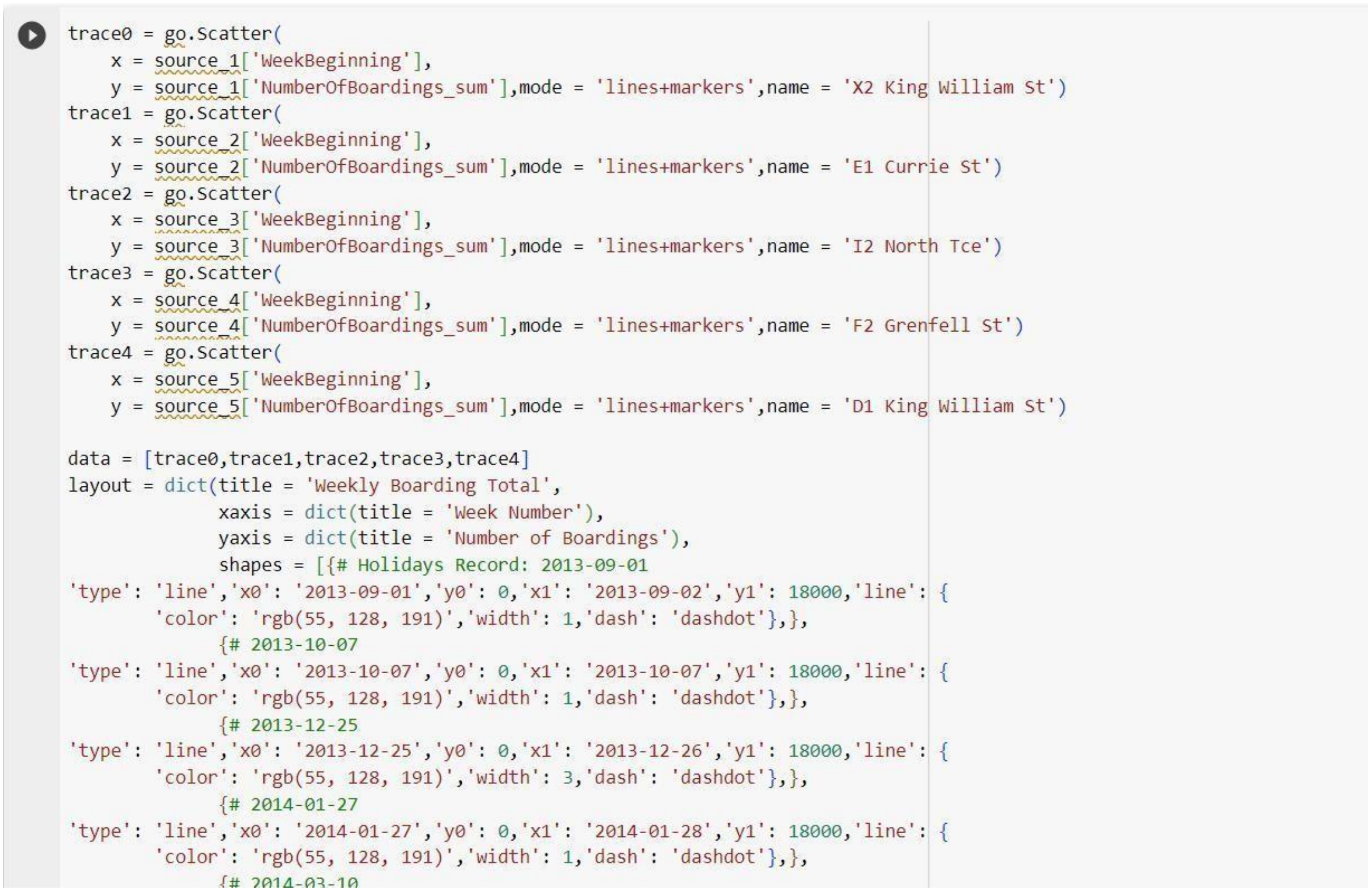
Output:



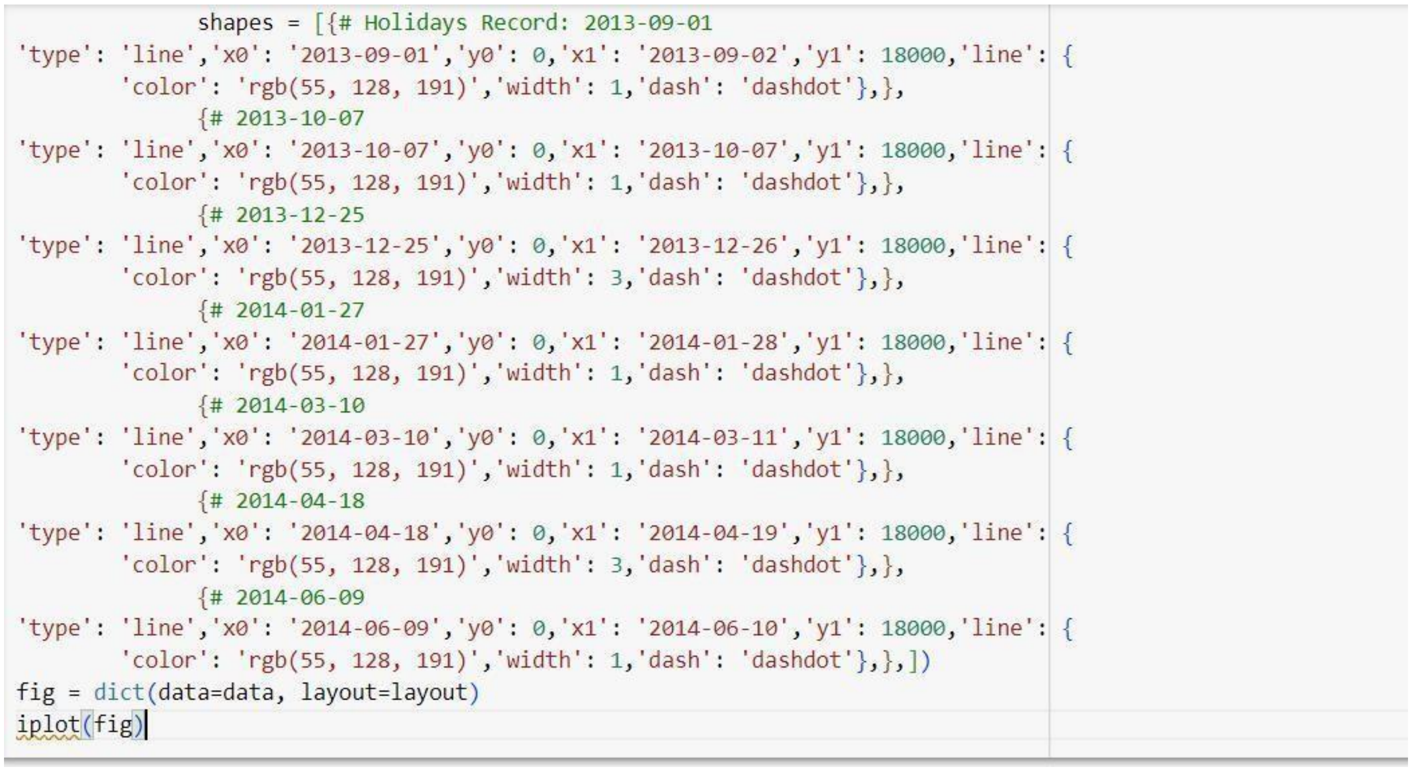
Plot using Plotly:

Program:

```
source_1 = bb[bb['StopName'] == 'X2 King William St'].reset_index(drop = True)
source_2 = bb[bb['StopName'] == 'E1 Currie St'].reset_index(drop = True)
source_3 = bb[bb['StopName'] == 'I2 North Tce'].reset_index(drop = True)
source_4 = bb[bb['StopName'] == 'F2 Grenfell St'].reset_index(drop = True)
source_5 = bb[bb['StopName'] == 'D1 King William St'].reset_index(drop = True)
```



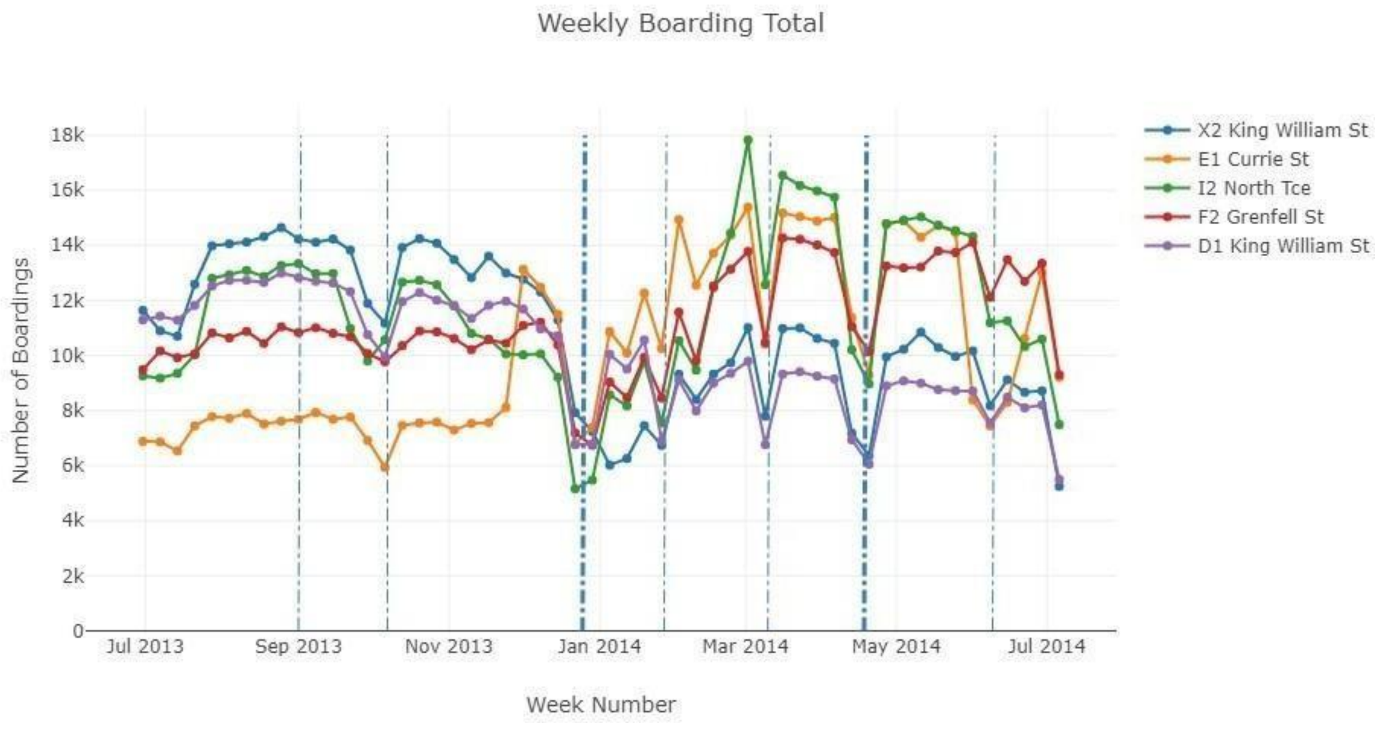
```
trace0 = go.Scatter(
    x = source_1['WeekBeginning'],
    y = source_1['NumberOfBoardings_sum'],mode = 'lines+markers',name = 'X2 King William St')
trace1 = go.Scatter(
    x = source_2['WeekBeginning'],
    y = source_2['NumberOfBoardings_sum'],mode = 'lines+markers',name = 'E1 Currie St')
trace2 = go.Scatter(
    x = source_3['WeekBeginning'],
    y = source_3['NumberOfBoardings_sum'],mode = 'lines+markers',name = 'I2 North Tce')
trace3 = go.Scatter(
    x = source_4['WeekBeginning'],
    y = source_4['NumberOfBoardings_sum'],mode = 'lines+markers',name = 'F2 Grenfell St')
trace4 = go.Scatter(
    x = source_5['WeekBeginning'],
    y = source_5['NumberOfBoardings_sum'],mode = 'lines+markers',name = 'D1 King William St')

data = [trace0,trace1,trace2,trace3,trace4]
layout = dict(title = 'Weekly Boarding Total',
              xaxis = dict(title = 'Week Number'),
              yaxis = dict(title = 'Number of Boardings'),
              shapes = [{# Holidays Record: 2013-09-01
'type': 'line','x0': '2013-09-01','y0': 0,'x1': '2013-09-02','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 1,'dash': 'dashdot'},},
              {# 2013-10-07
'type': 'line','x0': '2013-10-07','y0': 0,'x1': '2013-10-07','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 1,'dash': 'dashdot'},},
              {# 2013-12-25
'type': 'line','x0': '2013-12-25','y0': 0,'x1': '2013-12-26','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 3,'dash': 'dashdot'},},
              {# 2014-01-27
'type': 'line','x0': '2014-01-27','y0': 0,'x1': '2014-01-28','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 1,'dash': 'dashdot'},},
              {# 2014-03-10
```



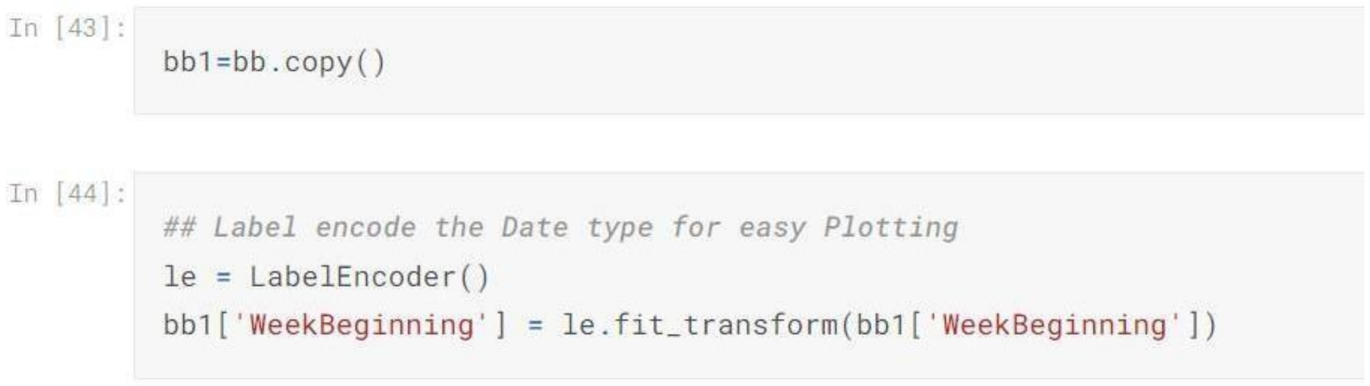
```
              shapes = [{# Holidays Record: 2013-09-01
'type': 'line','x0': '2013-09-01','y0': 0,'x1': '2013-09-02','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 1,'dash': 'dashdot'},},
              {# 2013-10-07
'type': 'line','x0': '2013-10-07','y0': 0,'x1': '2013-10-07','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 1,'dash': 'dashdot'},},
              {# 2013-12-25
'type': 'line','x0': '2013-12-25','y0': 0,'x1': '2013-12-26','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 3,'dash': 'dashdot'},},
              {# 2014-01-27
'type': 'line','x0': '2014-01-27','y0': 0,'x1': '2014-01-28','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 1,'dash': 'dashdot'},},
              {# 2014-03-10
```

```
'type': 'line','x0': '2014-03-10','y0': 0,'x1': '2014-03-11','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 1,'dash': 'dashdot'},},
              {# 2014-04-18
'type': 'line','x0': '2014-04-18','y0': 0,'x1': '2014-04-19','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 3,'dash': 'dashdot'},},
              {# 2014-06-09
'type': 'line','x0': '2014-06-09','y0': 0,'x1': '2014-06-10','y1': 18000,'line': {
        'color': 'rgb(55, 128, 191)','width': 1,'dash': 'dashdot'},},])
fig = dict(data=data, layout=layout)
iplot(fig)
```

Output:



Plot Using Bubbly:



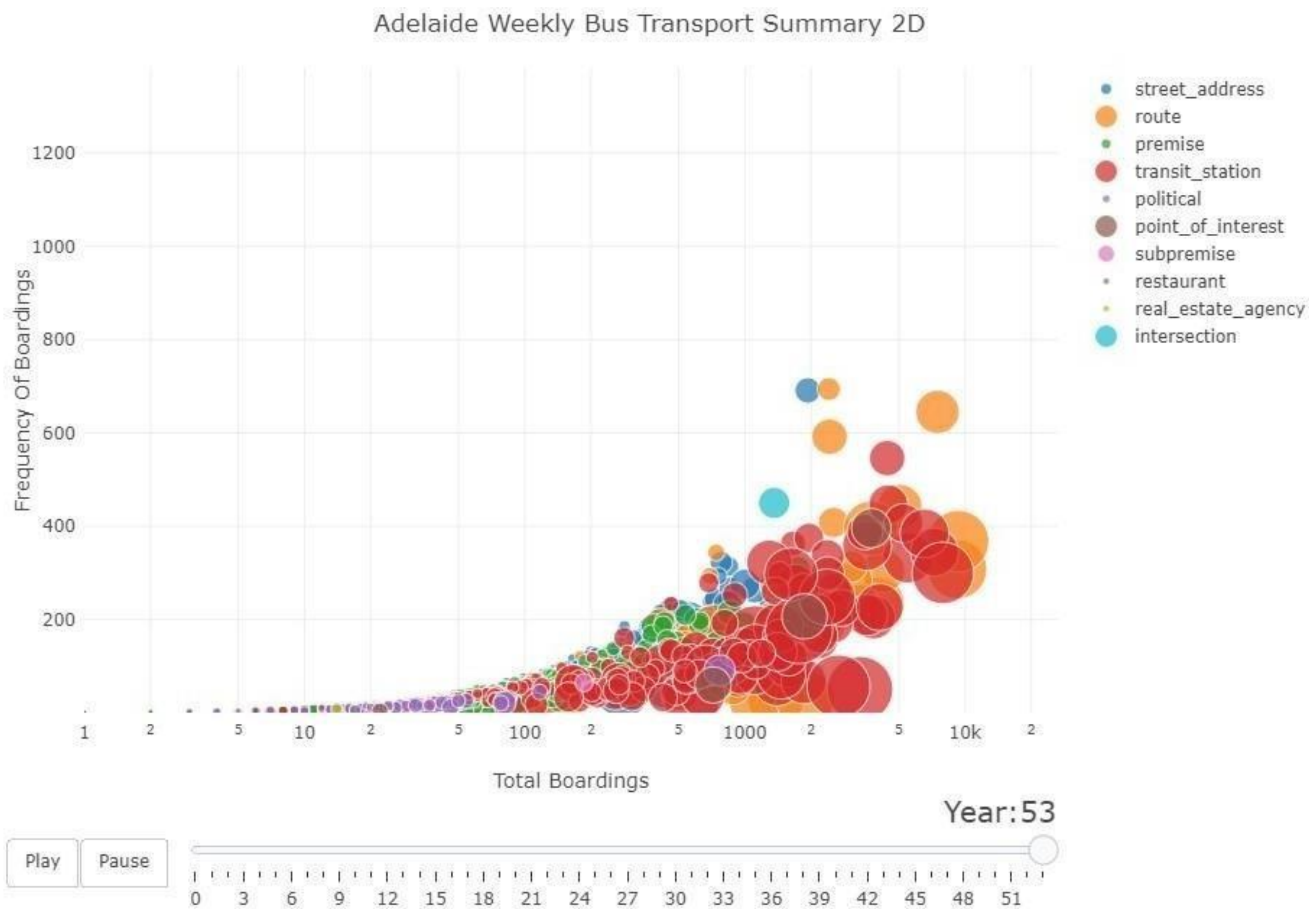
```
In [43]:  bb1=bb.copy()
```

```
In [44]:  ## Label encode the Date type for easy Plotting
          le = LabelEncoder()
          bb1['WeekBeginning'] = le.fit_transform(bb1['WeekBeginning'])
```

2D Plot with 6 different variables:

```
figure = bubbleplot(dataset=bb1, x_column='NumberOfBoardings_sum', y_column='Nu
mberOfBoardings_count',
    bubble_column='StopName', time_column='WeekBeginning', size_column='NumberO
fBoardings_max',
    color_column='type',
    x_title="Total Boardings", y_title="Frequency Of Boardings",show_slider=Tru
e,
    title='Adelaide Weekly Bus Transport Summary 2D',x_logscale=True, scale_bub
ble=2,height=650)

iplot(figure, config={'scrollzoom': True})
```

Output:

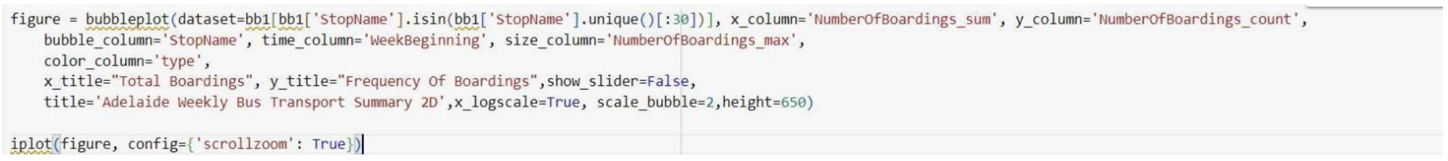


The animated bubble charts convey a great deal of information since they can accomodate upto seven variables in total, namely:

- X-axis (Total Boardings per week)
- Y-axis (Frequency of Bus Boarding)
- Bubbles (Bus stop name)
- Time (in week period)
- Size of bubbles (maximum number of people board at single time)
- Color of bubbles (Type of Bus stop)

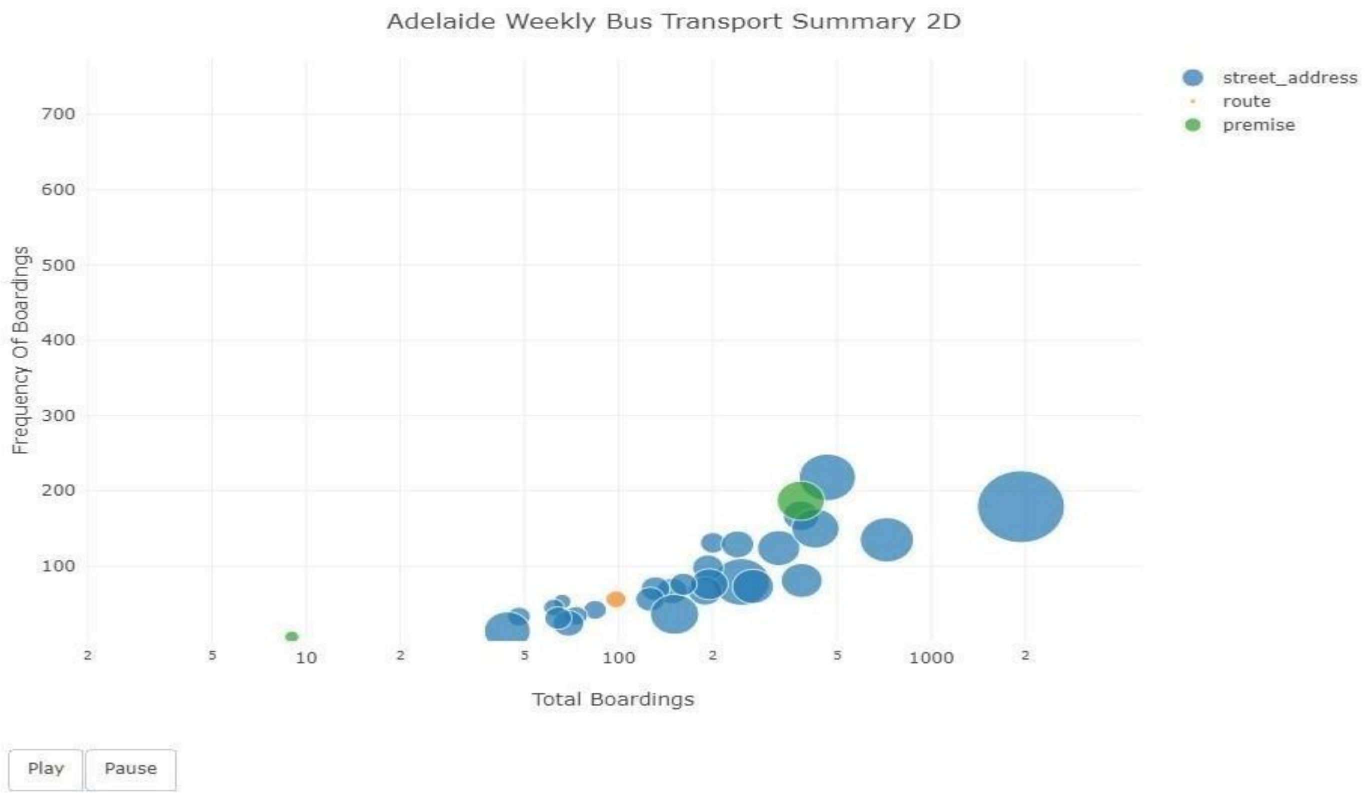
Plot for first 30 stops:

Program:



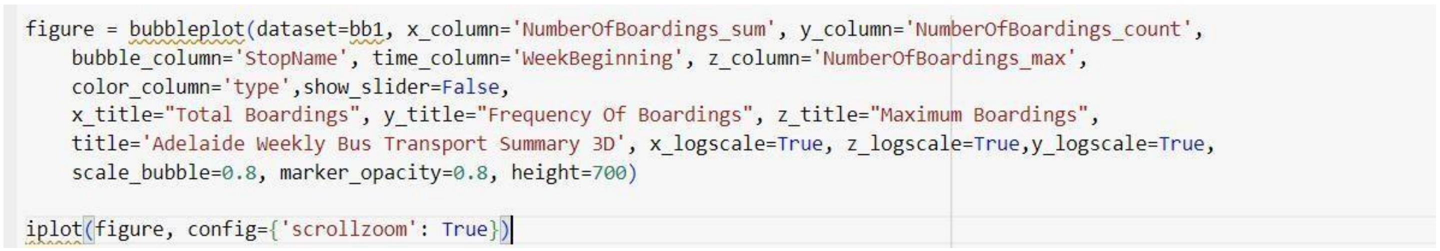
```
figure = bubbleplot(dataset=bb1[bb1['StopName'].isin(bb1['StopName'].unique()[:30])], x_column='NumberOfBoardings_sum', y_column='NumberOfBoardings_count',
    bubble_column='StopName', time_column='WeekBeginning', size_column='NumberOfBoardings_max',
    color_column='type',
    x_title="Total Boardings", y_title="Frequency Of Boardings",show_slider=False,
    title='Adelaide Weekly Bus Transport Summary 2D',x_logscale=True, scale_bubble=2,height=650)

iplot(figure, config={'scrollzoom': True})
```

Output:



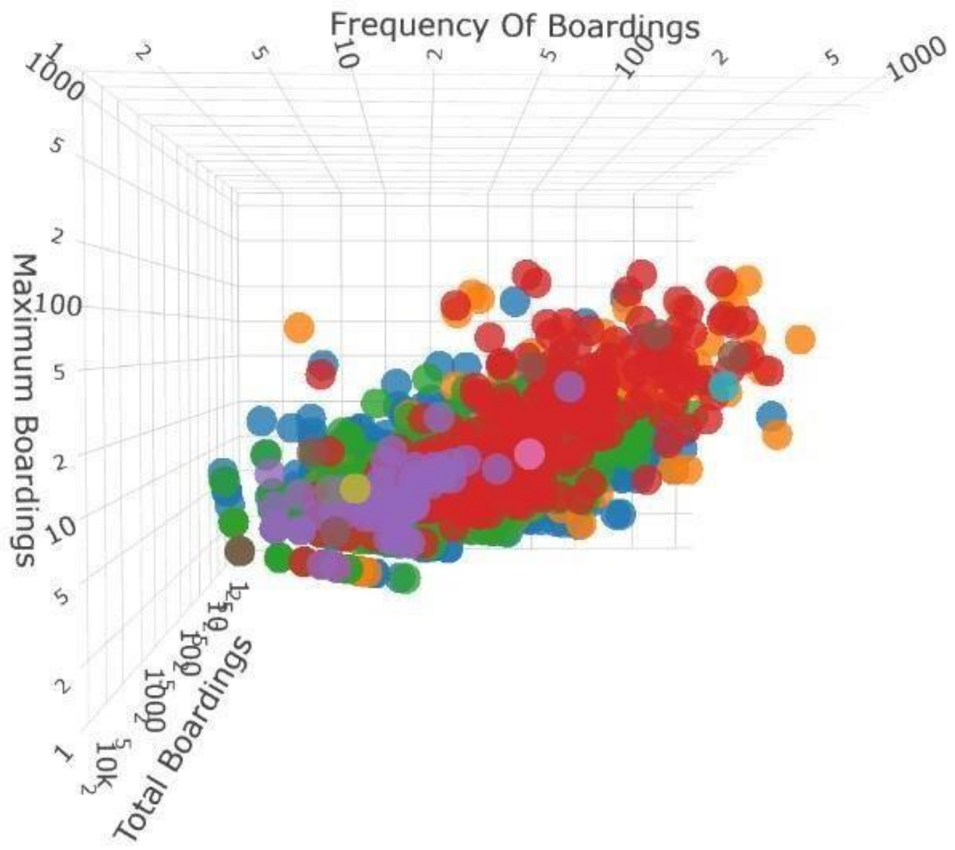
3D Bubble Plot with 6 different variables & there Relationship:

Program:

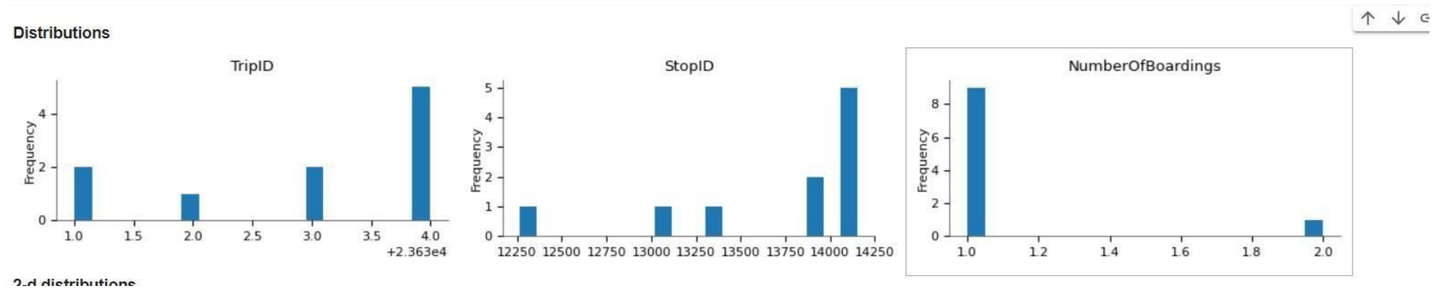


```
figure = bubbleplot(dataset=bb1, x_column='NumberOfBoardings_sum', y_column='NumberOfBoardings_count',
    bubble_column='StopName', time_column='WeekBeginning', z_column='NumberOfBoardings_max',
    color_column='type',show_slider=False,
    x_title="Total Boardings", y_title="Frequency Of Boardings", z_title="Maximum Boardings",
    title='Adelaide Weekly Bus Transport Summary 3D', x_logscale=True, z_logscale=True,y_logscale=True,
    scale_bubble=0.8, marker_opacity=0.8, height=700)

iplot(figure, config={'scrollzoom': True})
```

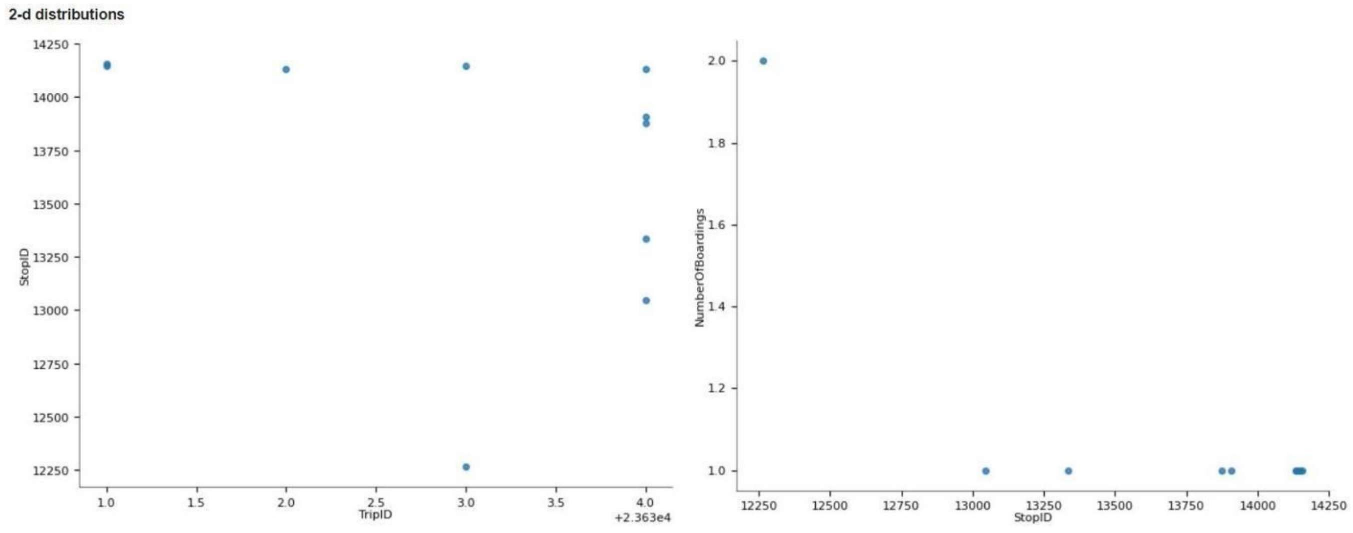
Output:



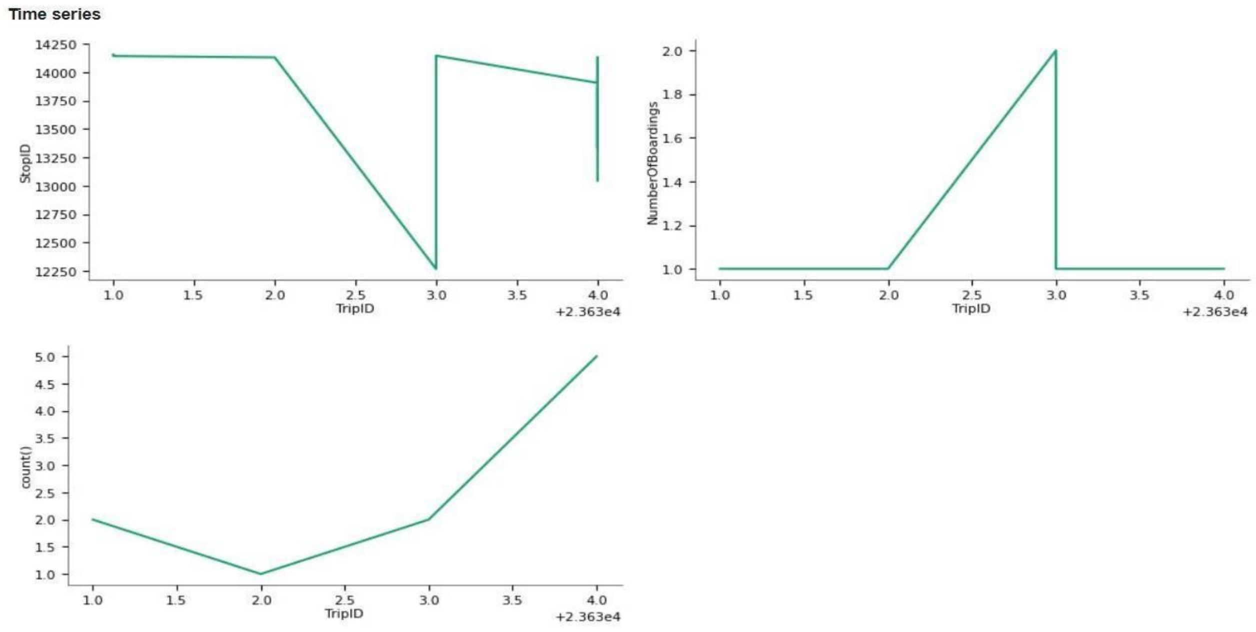
Distributions:



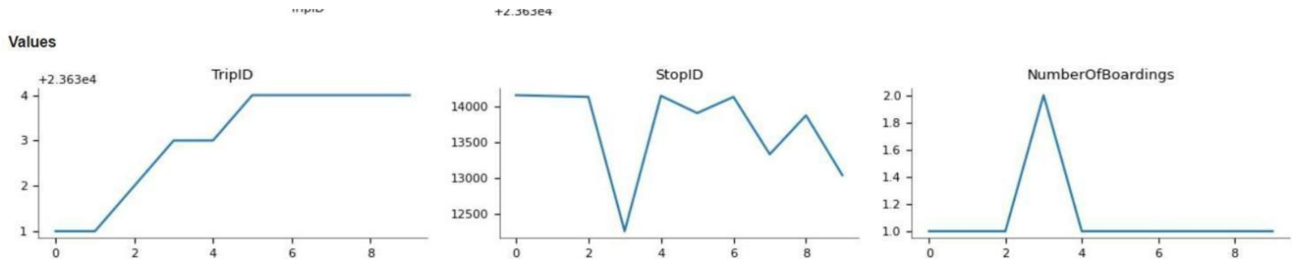
2-d distrubtions:



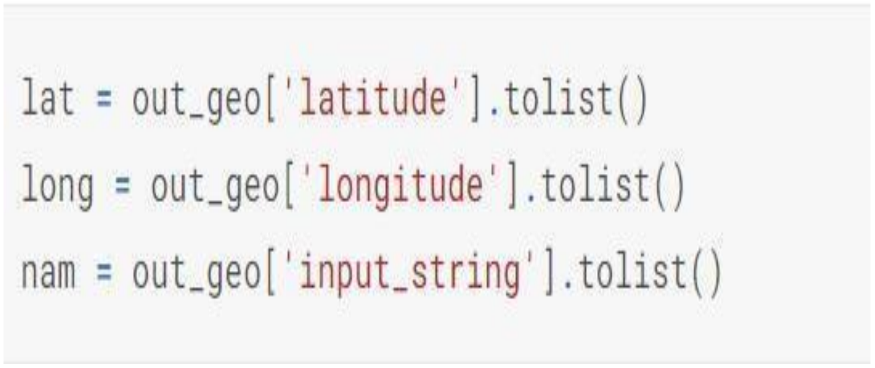
Time series:



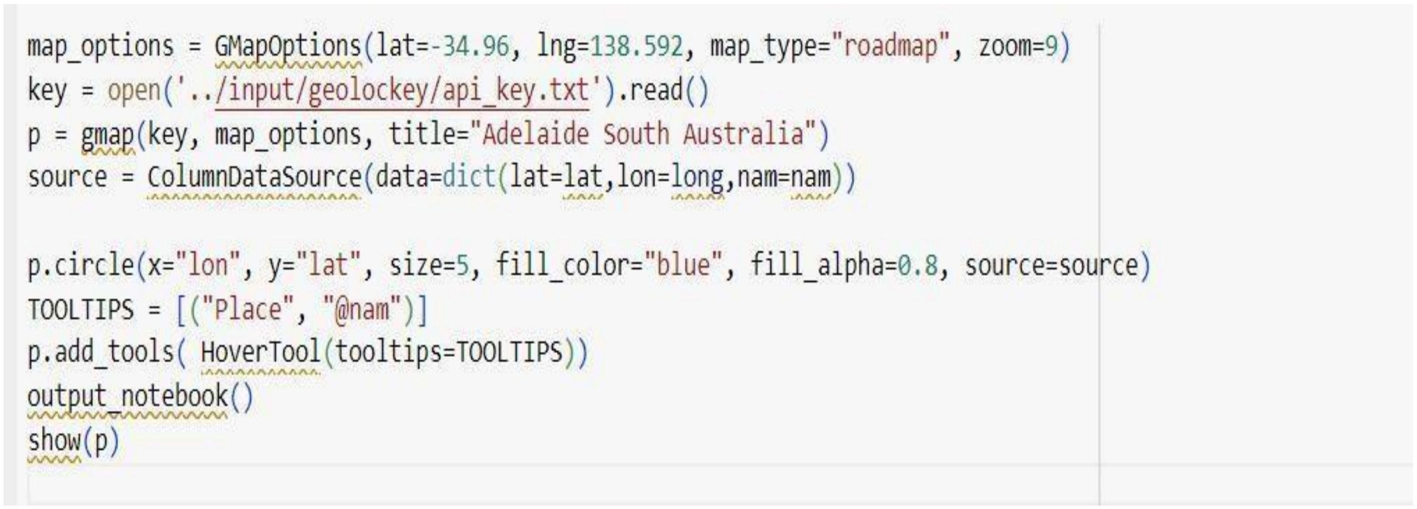
Values:



Using Bokeh:

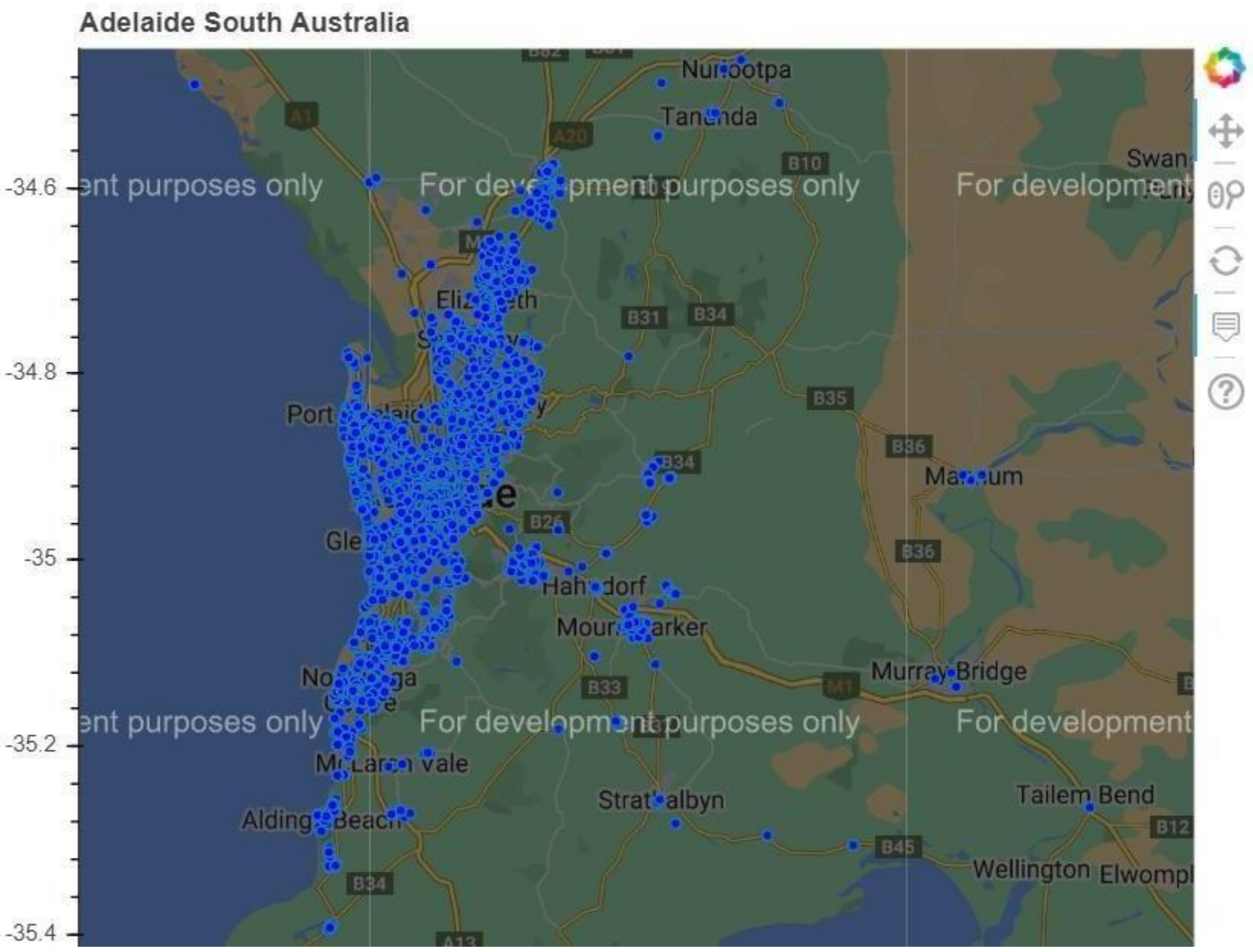


```
lat = out_geo['latitude'].tolist()
long = out_geo['longitude'].tolist()
nam = out_geo['input_string'].tolist()
```



```
map_options = GMapOptions(lat=-34.96, lng=138.592, map_type="roadmap", zoom=9)
key = open('../input/geolockey/api_key.txt').read()
p = gmap(key, map_options, title="Adelaide South Australia")
source = ColumnDataSource(data=dict(lat=lat,lon=long,nam=nam))

p.circle(x="lon", y="lat", size=5, fill_color="blue", fill_alpha=0.8, source=source)
TOOLTIPS = [("Place", "@nam")]
p.add_tools( HoverTool(tooltips=TOOLTIPS))
output_notebook()
show(p)
```

Output:


Adelaide South Australia
-34.6
-34.8
-35
-35.2
-35.4
Nuriootpa
Tanunda
Elizabeth
Port Adelaide
Glenelg
Hahndorf
Mount Barker
Noarlunga Centre
McLaren Vale
Aldinga Beach
Strathalbyn
Mannum
Murray Bridge
Tailem Bend
Wellington
Swan Reach
For development purposes only

Usage of IBM COGNOS:

Create a dashboard:

Creating a dashboard in IBM Cognos typically involves the following step-by-step process:

1. **Data Preparation:**

Ensure that you have the necessary data sources ready and accessible in your IBM Cognos environment.

2. **Launch IBM Cognos:**

Open the IBM Cognos platform or web interface.

3. **Create a New Dashboard:**

Navigate to the area where you can create a new dashboard, typically within the "Authoring" or "Dashboards" section.

3. **Choose the Layout:**

Select a layout for your dashboard. Cognos usually provides various templates to choose from.

4. **Add Data Visualizations:**

Add data visualizations like charts, tables, and graphs to your dashboard. Here's how:

- Click on the area where you want to add a visualization.
- Drag and drop data elements onto the canvas.
- Configure the visualization properties, including data source, dimensions, and measures.

5. **Design the Dashboard:**

Customize the dashboard's appearance by adjusting colours, fonts, and other visual elements.

6. **Add Filters and Interactivity:**

To make the dashboard interactive, add filter elements that allow users to drill down or filter data dynamically.

7. **Create Data Connections:**

Establish connections to your data sources, and set up data queries or import data as needed.

8. **Define Prompts:**

If you want to prompt users for input, create parameterized queries or prompts within your dashboard.

10. **Apply Conditional Formatting:**

Use conditional formatting to highlight specific data points based on predefined rules.

11. **Add Text and Annotations:**

Include text boxes, annotations, or titles to provide context and explanations for the data.

12. **Save and Publish:**

Save your dashboard in your IBM Cognos environment. You can also publish it to make it accessible to others.

13. **Share the Dashboard:**

Share the dashboard with authorized users or groups, and configure access permissions.

14. **Test and Review:**

Thoroughly test the dashboard to ensure that it functions as expected and provides valuable insights.

15. **Schedule Updates:**

If necessary, set up automated data updates or refresh schedules.

16. Documentation:

Document the dashboard's purpose, data sources, and any important details for users.

17. Training:

If the dashboard is intended for a wider audience, provide training or user guides to help users effectively utilize it.

18. Monitor and Maintain:

Continuously monitor the performance of your dashboard and make updates or improvements as needed.

Visualizations in IBM Cognos:

IBM Cognos provides a variety of data visualizations to help users effectively analyze and present their data. These visualizations can be used in reports, dashboards, and other Cognos content. Here are some common types of visualizations available in IBM Cognos:

1. Bar Charts:

- Clustered Bar Chart
- Stacked Bar Chart
- 100% Stacked Bar Chart
- Dual-Axis Bar Chart

2. Column Charts:

- Clustered Column Chart
- Stacked Column Chart
- 100% Stacked Column Chart
- Dual-Axis Column Chart

3. Line Charts:

- Single-Line Chart
- Multi-Line Chart
- Area Chart

4. **Pie Charts:**

- Standard Pie Chart
- Exploded Pie Chart
- Donut Chart

5. **Scatter Plots:**

- Scatter Plot
- Bubble Chart

6. **Area Charts:**

- Area Chart
- Stacked Area Chart
- 100% Stacked Area Chart

7. **Heat Maps:**

- Heat Map Chart

8. **Maps:**

- Geo Map
- Choropleth Map

9. **Tables:**

- Cross-Tab Table
- List Report
- Repeater Table
- Rave Table

10. **Crosstabs:**

- Standard Crosstab
- Pivot Crosstab

11. **Gauges:**

- Linear Gauge
- Radial Gauge
- Knob Gauge

12. **Combination Charts:**

- Combines different chart types in a single visualization.

13. **Waterfall Charts:**

- Visualizes cumulative data in a step-by-step manner.

14. **Bullet Graphs:**

- Compares a primary measure to one or more other measures or targets.

15. **Treemaps:**

- Hierarchical visualization that shows data as nested rectangles.

16. **Radar Charts:**

- Displays data on a spider-web-like chart.

17. **Box Plots:**

- Provides statistical information about the distribution of data.

18. **KPI (Key Performance Indicator) Tiles:**

- Compact visualizations for displaying KPIs and metrics.

19. **Sunburst Charts:**

- Radial visualization for hierarchical data.

20. **Funnel Charts:**

- Used for visualizing stages in a process, such as a sales funnel.

21. **Polar Charts:**

- Radial charts for displaying data in a circular format.

22. **Tag Clouds:**

- Visualizes word frequency by varying font size based on data values.

23. **Histograms:**

- Represents the distribution of data.

24. **Packed Bubbles:**

- Hierarchical bubble chart where bubbles can nest inside one another.

25. **Spider Web Charts:**

- Used for comparing multiple categories of data.

These are just some of the visualization types available in IBM Cognos. Users can choose the most appropriate visualization type based on their data and the insights they want to convey to their audience. The availability of these visualization options may vary depending on the specific version and configuration of IBM Cognos.

Visualization:

Creating visualizations for on-time performance, passenger feedback, and service efficiency metrics with the given dataset requires a creative approach since the dataset does not contain explicit data for these metrics. However, you can still design visualizations based on the available metrics.

1. On-Time Performance Visualization:

On-Time Performance Visualization refers to the graphical representation of data related to the punctuality or timeliness of events, processes, or activities. This type of visualization is often used in various fields to assess and communicate how well something adheres to a schedule or deadline. It can be particularly important in industries where timely performance is critical, such as transportation, logistics, manufacturing, and project management.

Here are some examples of On-Time Performance Visualization:

1. **Flight On-Time Performance:** Airlines and airports use visualizations to display the punctuality of flights, showing metrics like the percentage of flights that depart and arrive on time.

2. **Train or Bus Schedule Adherence:** Public transportation systems use visualizations to illustrate how well their services adhere to published schedules, helping passengers plan their journeys.

3. **Manufacturing Production Efficiency:** Manufacturing companies may use visualizations to track the on-time performance of production processes, highlighting areas where delays occur.

4. **Project Management:** Project managers can create Gantt charts or timelines to visualize project tasks and milestones to ensure they are completed on time.

5. Service Level Agreements (SLAs): Businesses often track and visualize their adherence to SLAs when providing services to clients. Visualizations can show whether response times or service delivery meet agreed-upon deadlines.

6. Academic Timetables: Schools and universities may create visual schedules to help students and faculty keep track of classes and events and ensure they start and end on time.

On-Time Performance Visualizations can take various forms, including bar charts, line graphs, heatmaps, pie charts, and more. These visualizations make it easier to identify trends, patterns, and areas that need improvement when it comes to punctuality and adherence to schedules.

In the absence of direct on-time data, you can create an "On-Time Score" based on the "No of boardings." This score might represent how well a route or trip meets its passenger demand expectations.

X-Axis: Route ID or Trip ID
Y-Axis: On-Time Score (based on No of boardings)
a. Calculate an "On-Time Score" for each Route ID or Trip ID based on the "No of boardings." For example, you can use a threshold value or formula to determine whether a route or trip is considered on-time in terms of meeting its expected demand.

b. Create a bar chart or a scatter plot where the X-axis represents the Route ID or Trip ID and the Y-axis represents the "On-Time Score."

c. Customize the chart, labels, and titles as needed.

Line graph:

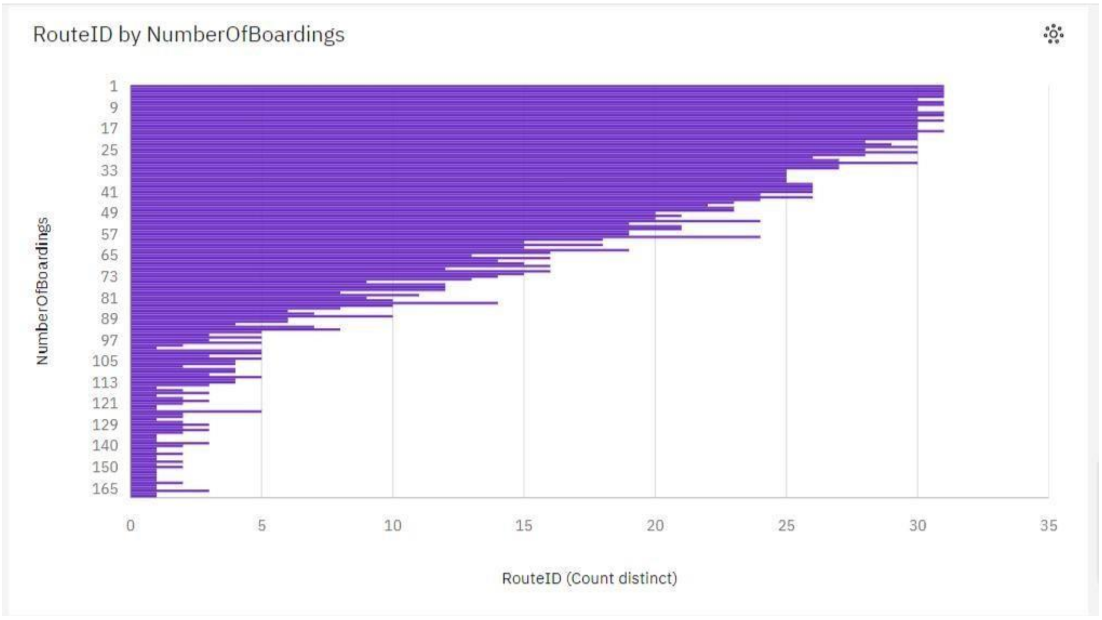
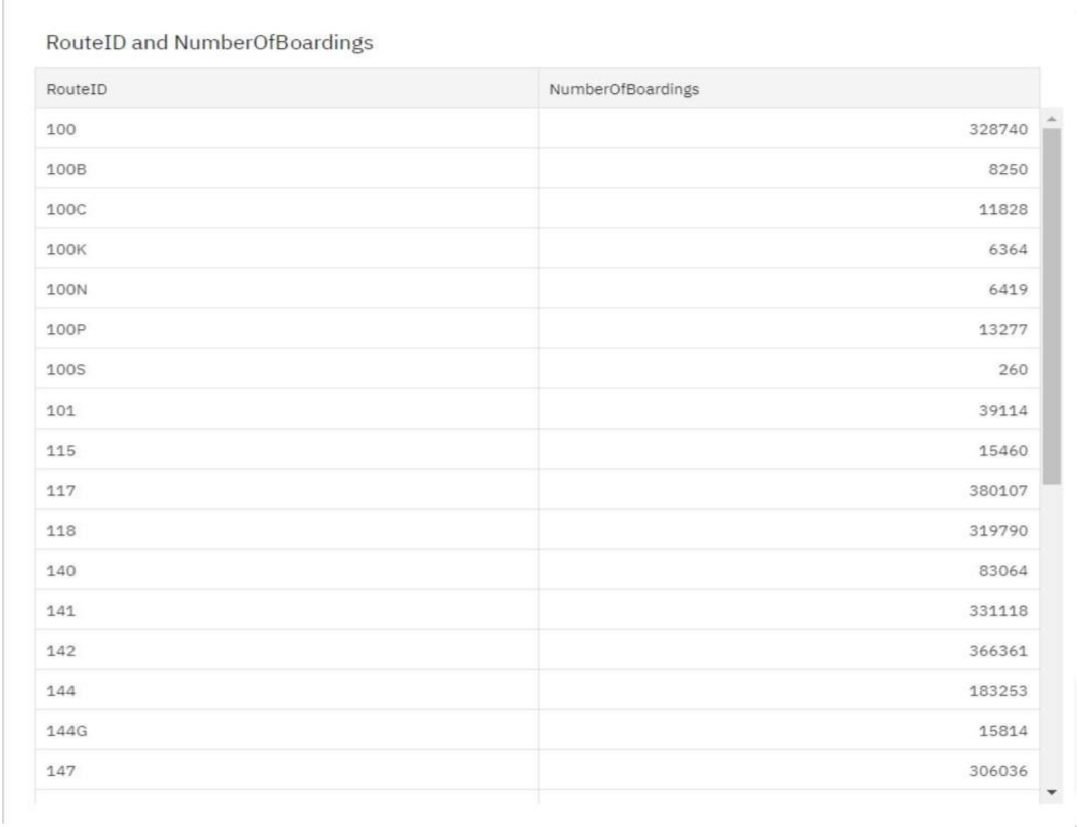
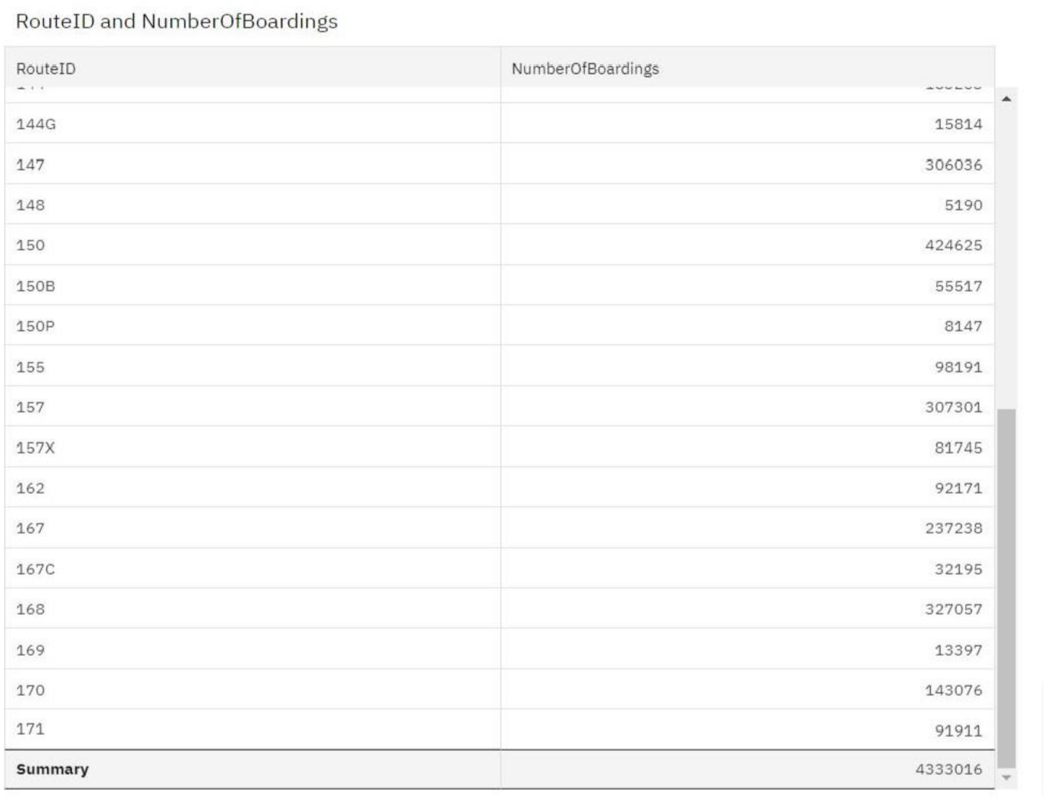


Table graph:



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	237238
167C	32195
168	327057
169	13397
170	143076
171	91911
Summary	4333016

2. Service Efficiency Visualization:

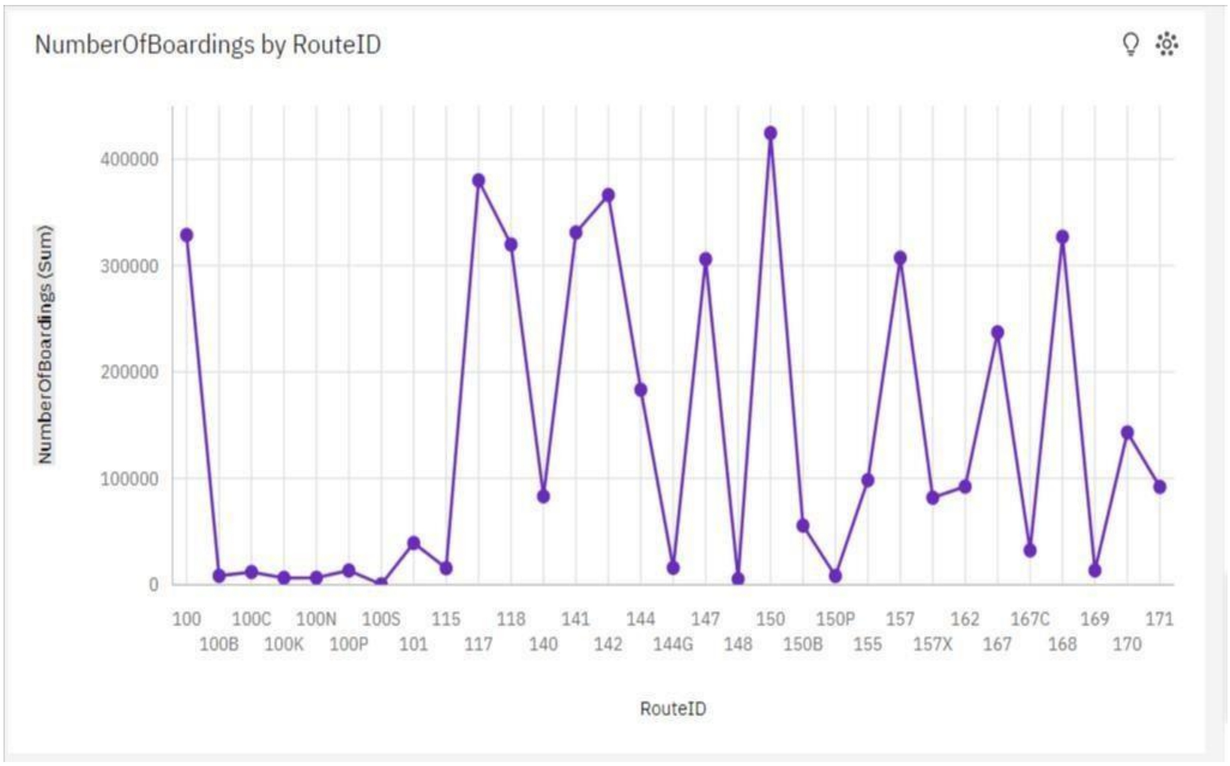
To assess service efficiency, you can continue to use the "No of boardings" metric as an indicator of demand:

X-Axis: Route ID or Stop Name
Y-Axis: No of boardings
a. Create a chart that shows the "No of boardings" for each Route ID or Stop Name. You can use a bar chart or a table.

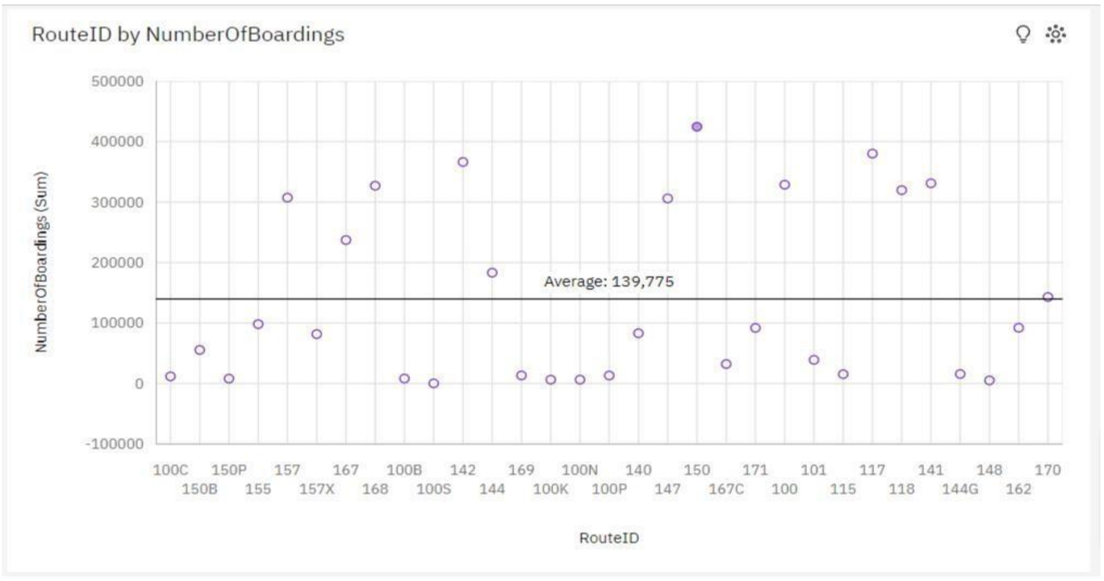
b. Customize the chart to make it visually informative.

While these visualizations don't directly measure traditional on-time performance or passenger feedback, they can provide insights into how well routes or trips are performing based on passenger demand. Remember that the available metrics drive the type of analysis and visualization you can create, and adapting them to your specific use case is important.

Line graph:



Scatterplot:



3. Passenger Feedback Visualization:

Create a visualization to understand passenger feedback based on available data. Assuming you have feedback scores associated with Trip IDs or Route IDs:

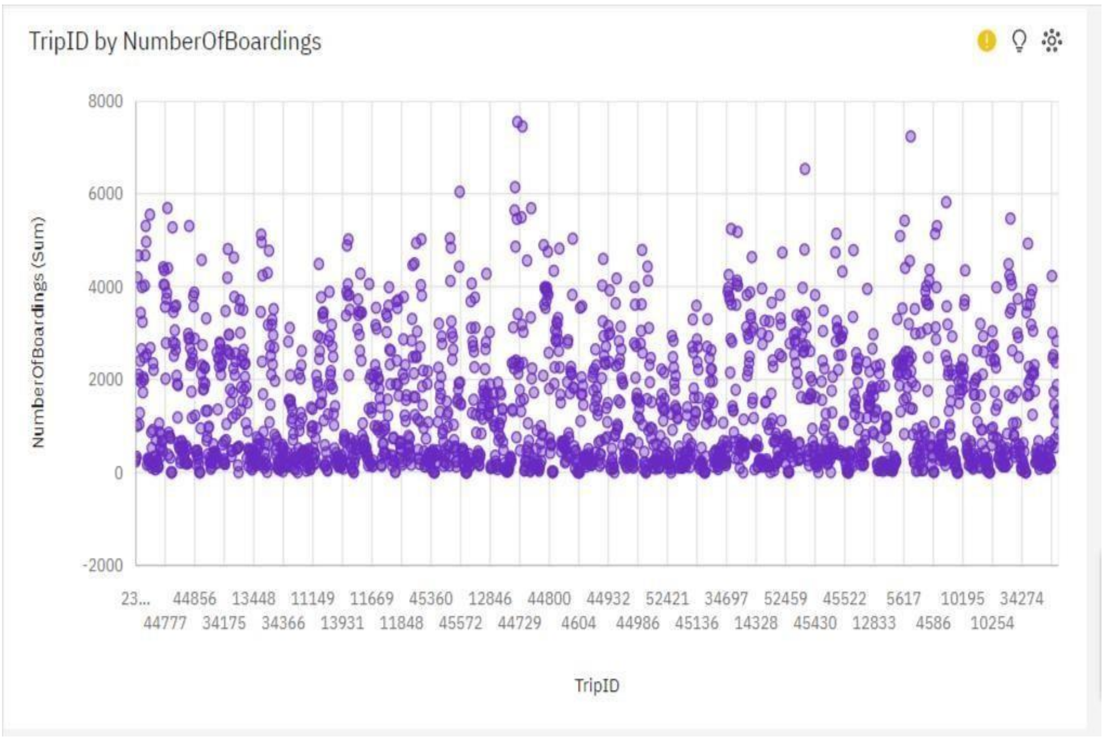
Category Axis: Trip ID or Route ID
Data Point: Average Feedback Score

a. Calculate the average feedback score for each Trip ID or Route ID.

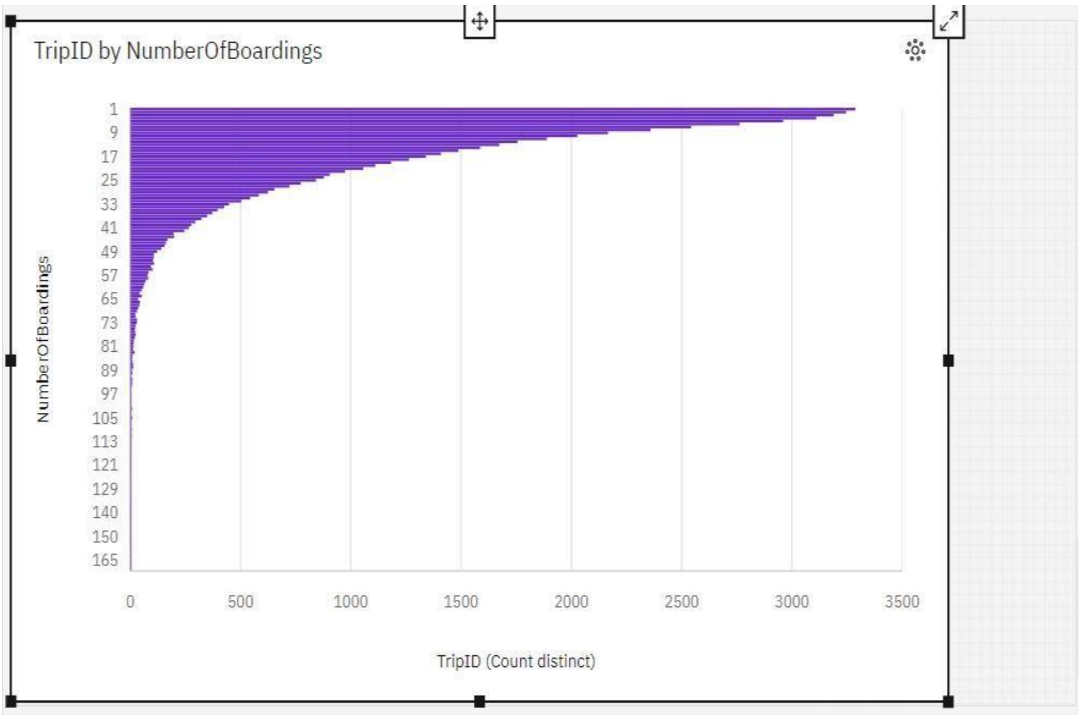
b. Create a bar chart where the X-axis represents Trip ID or Route ID, and the Y-axis represents the average feedback score.

c. Customize the chart and labels.

Scatter plot:



Bar Graph:



Advanced data analysis, such as calculating service punctuality rates or sentiment analysis on passenger feedback:

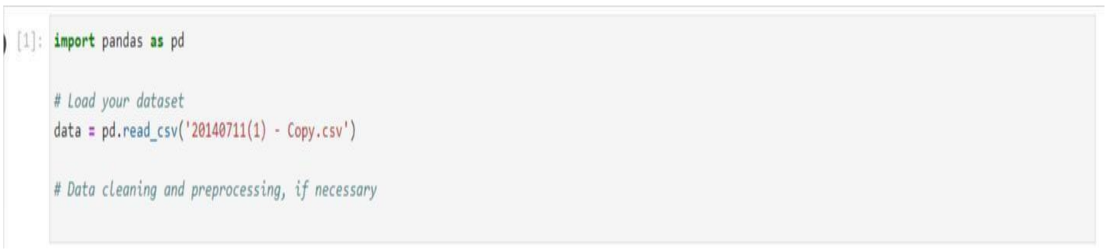
The advanced data analysis in this context is twofold: first, to calculate service punctuality rates by defining a threshold for on-time performance based on passenger demand data; and second, to perform indirect sentiment analysis on passenger feedback by categorizing sentiment based on passenger demand levels. By implementing these analyses, the goal is to assess transportation service performance and passenger satisfaction indirectly, leveraging available metrics to gain insights into service punctuality and passenger sentiment, even in the absence of explicit feedback scores. This data-driven approach enables data-informed decision-making and service quality improvement for the transportation system.

To perform advanced data analysis, including calculating service punctuality rates and sentiment analysis on passenger feedback, you can use Python with various libraries. Below are steps for performing both analyses:

1. Data Preparation:

Before you can perform data analysis, you need to load your dataset into Python and prepare it for analysis. You may use libraries like Pandas for data manipulation and cleaning.

Program :



```
[1]: import pandas as pd

     # Load your dataset
     data = pd.read_csv('20140711(1) - Copy.csv')

     # Data cleaning and preprocessing, if necessary
```

Service punctuality rates:

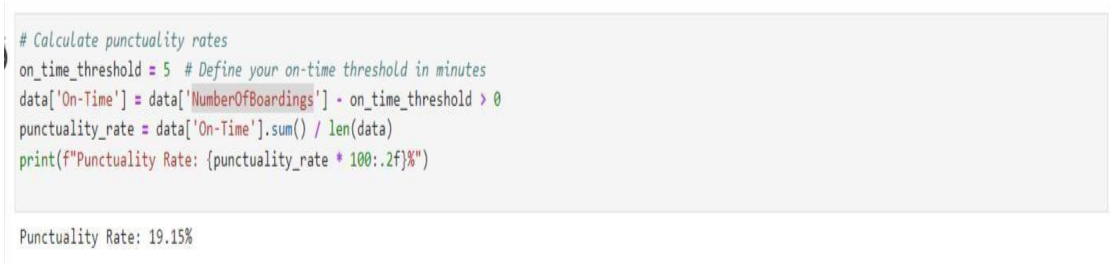
Service punctuality rates refer to the measure of how often a service, such as public transportation, arrives or departs on time according to a predefined schedule. Punctuality rates are used to assess the reliability and adherence of the service to its scheduled timetable.

For example, if a bus service is scheduled to depart from a stop at 9:00 AM, and it consistently departs within a few minutes of 9:00 AM, it is considered punctual. The punctuality rate is a metric that quantifies the percentage of times a service adheres to its schedule within an acceptable time window.

Punctuality rates are often used in the transportation industry to evaluate the on-time performance of buses, trains, flights, and other modes of transportation. They are crucial for ensuring efficient and reliable services, as deviations from the schedule can inconvenience passengers and disrupt travel plans.

For service punctuality rates, you need to define what you consider "on-time" and then calculate the rate based on your criteria. Let's assume a simple example where you consider a route on time if it arrives within 5 minutes of the scheduled time.

Program and output:



```python
# Calculate punctuality rates
on_time_threshold = 5  # Define your on-time threshold in minutes
data['On-Time'] = data['NumberOfBoardings'] - on_time_threshold > 0
punctuality_rate = data['On-Time'].sum() / len(data)
print(f"Punctuality Rate: {punctuality_rate * 100:.2f}%")
```

```
Punctuality Rate: 19.15%
```

Sentiment analysis on passenger feedback:

Sentiment analysis on passenger feedback, also known as opinion mining or emotion AI, is a process of using natural language processing (NLP) and machine learning techniques to analyze and determine the sentiment or emotional tone expressed in written or spoken feedback provided by passengers. The objective is to understand whether the feedback is positive, negative, or neutral and to what degree.

The sentiment analysis process typically involves:

1. Text Processing: Cleaning and preprocessing the text data, including tokenization (breaking text into words or phrases), removing stop words, and handling special characters.

2. Sentiment Classification: Assigning each text or comment a sentiment label, such as positive, negative, or neutral, based on the content and context of the text.

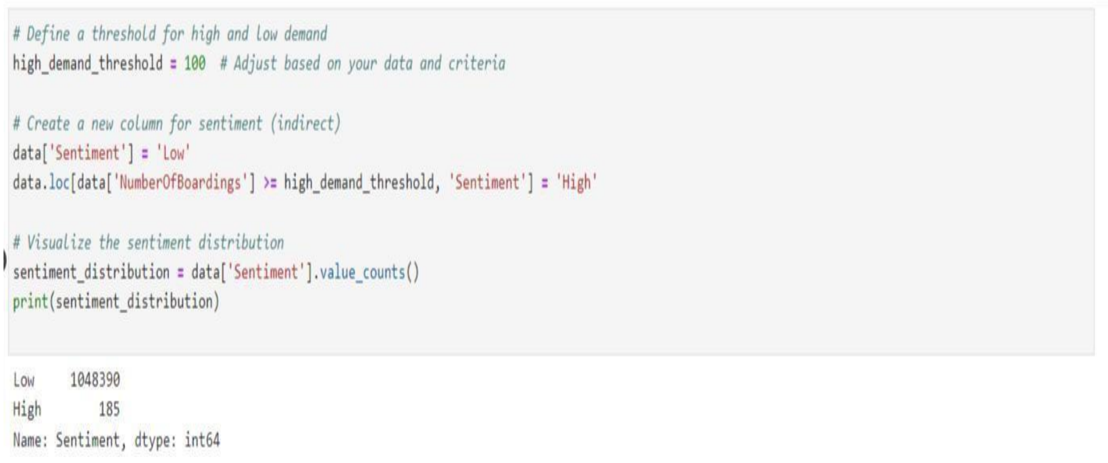
3. Sentiment Score: Assigning a sentiment score or polarity, which quantifies the intensity or degree of sentiment expressed in the feedback. For example, a score of -1 might represent strong negative sentiment, 0 for neutral, and +1 for strong positive sentiment.

4. Text Analysis: Extracting insights and patterns from the sentiment-labeled data, which can be used for decision-making, identifying areas for improvement, and understanding customer satisfaction or dissatisfaction.

Sentiment analysis is widely used in various industries, including customer service, market research, social media monitoring, and product reviews, to gauge public opinion and sentiments towards products, services, or experiences. In the context of passenger feedback, sentiment analysis helps transportation companies and authorities understand how passengers feel about their services, identify issues, and make data-driven improvements to enhance the passenger experience.

Since you mentioned that there's no explicit feedback score, you can perform an indirect sentiment analysis based on the available data, such as passenger demand. Higher demand can be assumed to indicate higher passenger satisfaction.

Program & output:



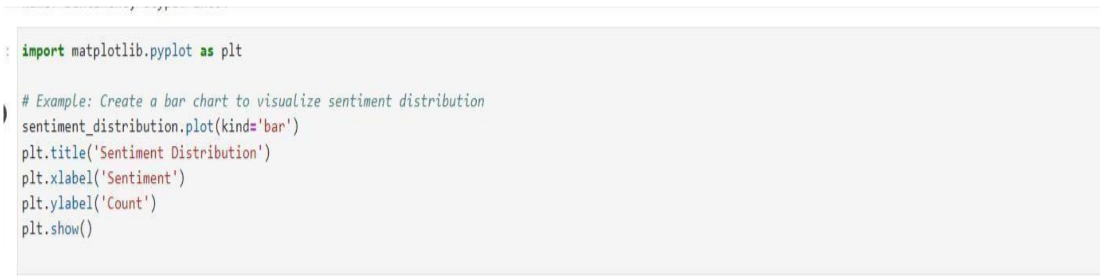
```python
# Define a threshold for high and low demand
high_demand_threshold = 100  # Adjust based on your data and criteria

# Create a new column for sentiment (indirect)
data['Sentiment'] = 'Low'
data.loc[data['NumberOfBoardings'] >= high_demand_threshold, 'Sentiment'] = 'High'

# Visualize the sentiment distribution
sentiment_distribution = data['Sentiment'].value_counts()
print(sentiment_distribution)
```

```
Low     1048390
High        185
Name: Sentiment, dtype: int64
```

4. Data Visualization:

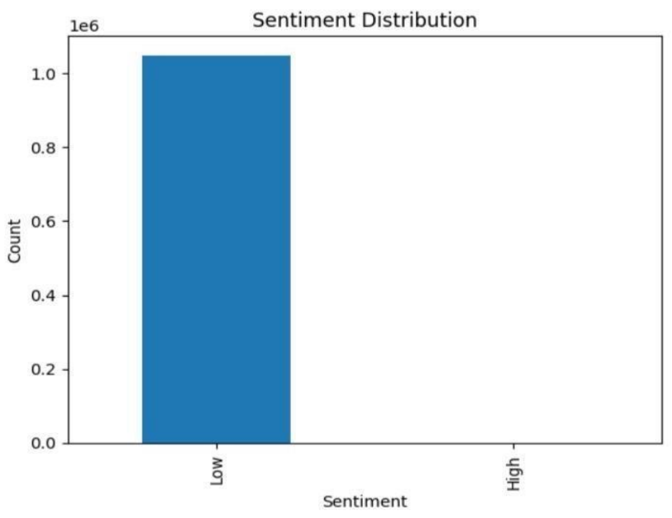
For further analysis, you can use libraries like Matplotlib or Seaborn to create visualizations that represent the calculated metrics or sentiments visually.

Program:



```python
import matplotlib.pyplot as plt

# Example: Create a bar chart to visualize sentiment distribution
sentiment_distribution.plot(kind='bar')
plt.title('Sentiment Distribution')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```

Output:



Outcome:

The outcome of the provided instructions is a comprehensive data analysis and reporting process aimed at addressing the given problem. Let's break down the outcome in detail:

1. Data Analysis Integration:

The instructions guide the process of combining two key aspects of data analysis: data visualization using IBM Cognos and the integration of code (e.g., Python) for advanced data analysis. This integration allows for a comprehensive examination of the problem at hand.

2. Visualizing Key Metrics:

The primary objective is to design dashboards and reports in IBM Cognos to visualize three critical metrics related to the transportation service:

On-Time Performance: This metric evaluates how well the service adheres to its schedule.
Passenger Feedback: Indirect analysis is employed to gauge passenger sentiment based on demand data.
Service Efficiency: The visualization aims to assess the efficiency of the service, primarily based on passenger demand.

3. Reporting in IBM Cognos:

IBM Cognos, as a business intelligence tool, provides the platform for designing interactive and informative reports and dashboards. These reports offer stakeholders a clear and user-friendly way to access and understand the analysis results.

4. Advanced Data Analysis with Code:

The use of code, such as Python, enables advanced data analysis:

Calculating Service Punctuality Rates: By defining on-time criteria and thresholds, the code quantifies how often the service is punctual, providing a measure of its reliability.
Sentiment Analysis on Passenger Feedback: Even without direct feedback scores, code can be used to categorize passenger sentiment based on available data, such as demand levels, offering insights into passenger satisfaction.

5. Data-Driven Decision-Making:

By combining data visualization in IBM Cognos with code-based analysis, the outcome is a powerful approach to data-driven decision-making. Transportation organizations can gain a deep understanding of their service performance and passenger satisfaction, leading to informed improvements.

6. Holistic Problem Solving:

These detailed steps constitute a holistic approach to addressing the problem. By leveraging the strengths of data visualization and code-based analysis, a more comprehensive understanding of the transportation service's performance and passenger sentiment is achieved. This approach is invaluable for making evidence-based decisions and enhancing the overall passenger experience.

Insights from the analysis:

1. **Route and Stop Optimization:**

Analysis of route id and stop name can help identify which routes and stops have the highest and lowest boardings. This information is crucial for optimizing routes and potentially eliminating underperforming routes or stops. It can lead to more efficient resource allocation, potentially saving costs and improving the overall quality of transportation services.

2. **Ridership Trends:**

By examining the number of boardings over time (possibly in relation to week beginnings), you can identify trends and patterns in ridership. This information can be used to adapt transportation schedules to match demand, improving service reliability and passenger satisfaction.

3. **Resource Allocation:**

Understanding the number of boardings at different stops and on various routes can guide resource allocation decisions. It helps in determining where additional buses or trains may be needed during peak times or where services can be reduced during off-peak periods, leading to better resource utilization.

4. **Service Planning:**

Trip id and week beginnings data can be used to analyze and plan for special events, holidays, or seasonal variations in ridership. By identifying when and where demand increases or decreases, transportation authorities can adjust schedules and services accordingly.

5. **Infrastructure Investment:**

Insights from the analysis can highlight areas with consistently high ridership, indicating the need for infrastructure investments, such as building larger stations or adding more boarding platforms. This can enhance the passenger experience and accommodate growing demand.

6. **Marketing and Outreach:**

For areas with low boardings, targeted marketing and outreach efforts can be initiated to raise awareness and encourage more people to use public transportation. Insights from the data can help prioritize marketing campaigns.

7. **Predictive Analysis:**

Historical data, when analyzed, can be used to create predictive models that forecast future ridership patterns. This can aid in long-term planning and decision-making, helping transportation authorities proactively adapt to changing demands.

8. **Customer Experience Improvement:**

By understanding which stops have the highest number of boardings, transportation authorities can focus on improving the quality of services at those locations, making them more attractive for passengers.

9. **Cost Reduction:**

By eliminating or optimizing underperforming routes and stops, transportation authorities can potentially reduce operational costs, which can be redirected towards improving services in other areas.

10. **Safety Enhancement:**

By analyzing the data, transportation authorities can identify stops or routes with a high incidence of safety-related incidents. This information can be used to implement safety measures, such as better lighting, surveillance, or security personnel to improve passenger safety.

11. **Environmental Sustainability:**

Insights into the number of boardings can help identify routes or areas with high ridership, allowing transportation authorities to prioritize the deployment of eco-friendly vehicles or implement green transportation initiatives in those areas to reduce emissions and support environmental sustainability.

12. **Accessibility Improvements:**

Understanding where passengers board can highlight the need for better accessibility options, such as ramps, elevators, and accessible vehicles, at specific stops to cater to passengers with disabilities.

13. **Intermodal Transportation Planning:**

Analysis of route id and stop name can help identify strategic locations for intermodal transportation hubs, where various modes of transportation (buses, trains, trams, etc.) can connect seamlessly, enhancing the overall transportation network's efficiency.

14. **Crowd Management:**

Insights from the number of boardings data can assist in managing crowded conditions during peak hours. Transportation authorities can plan for additional services or implement crowd control measures to improve the passenger experience and safety during busy periods.

Links for code integration:

Code for service efficiency,On-time performance,passenger satisfaction:

https://github.com/Pradeepking19/democode/blob/177385007dd673bba8e9c15b7bd2f05f136371fb/DAC_PHASE5.ipynb

GitHub link:

https://github.com/Pradeepking19/democode/blob/577536cb0aab9e0a7798ee3e976f57ec22919ed7/IBMcognosdashboard%20.pdf