PUBLIC TRANSPORT EFFICIENCY ANALYSIS

Phase 3 :Development part

Objective:

The primary objective of this project is to create a comprehensive public transportation efficiency analysis system using IBM Cognos for visualization. This involves defining analysis goals, collecting transportation data from specified sources, and meticulously processing and cleaning the data to ensure its quality and accuracy. Additionally, the project aims to integrate, transform, and load the data into IBM Cognos, ultimately developing informative visualizations and reports to provide meaningful insights. The ultimate goal is to support stakeholders in making informed decisions to enhance public transportation efficiency, while also ensuring the transparency, reproducibility, and documentation of the entire process.

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	########	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

30-06-2013 00:00

20140711

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

20140711

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.

<u>1.Loading the dataset:</u>

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(‘ ’)

trans.shape

trans.head(10)
```

```
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()
```

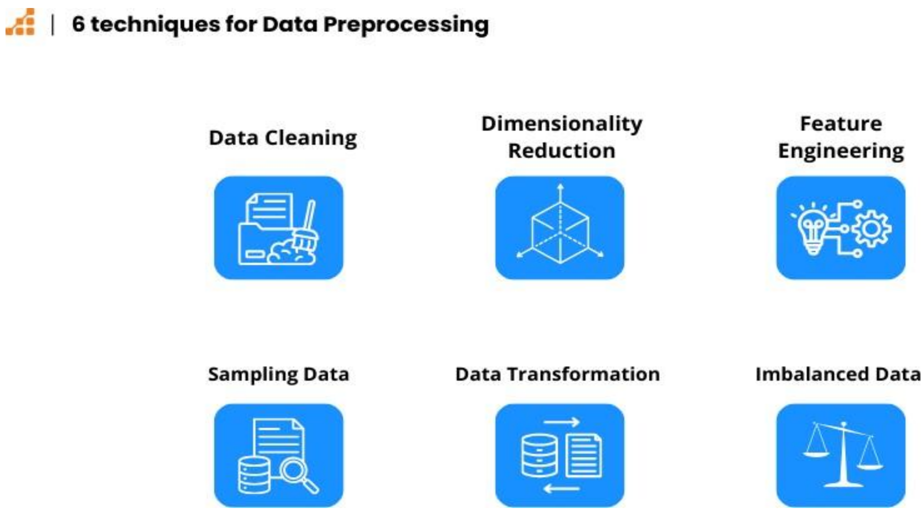
Output

```
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

4. **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.

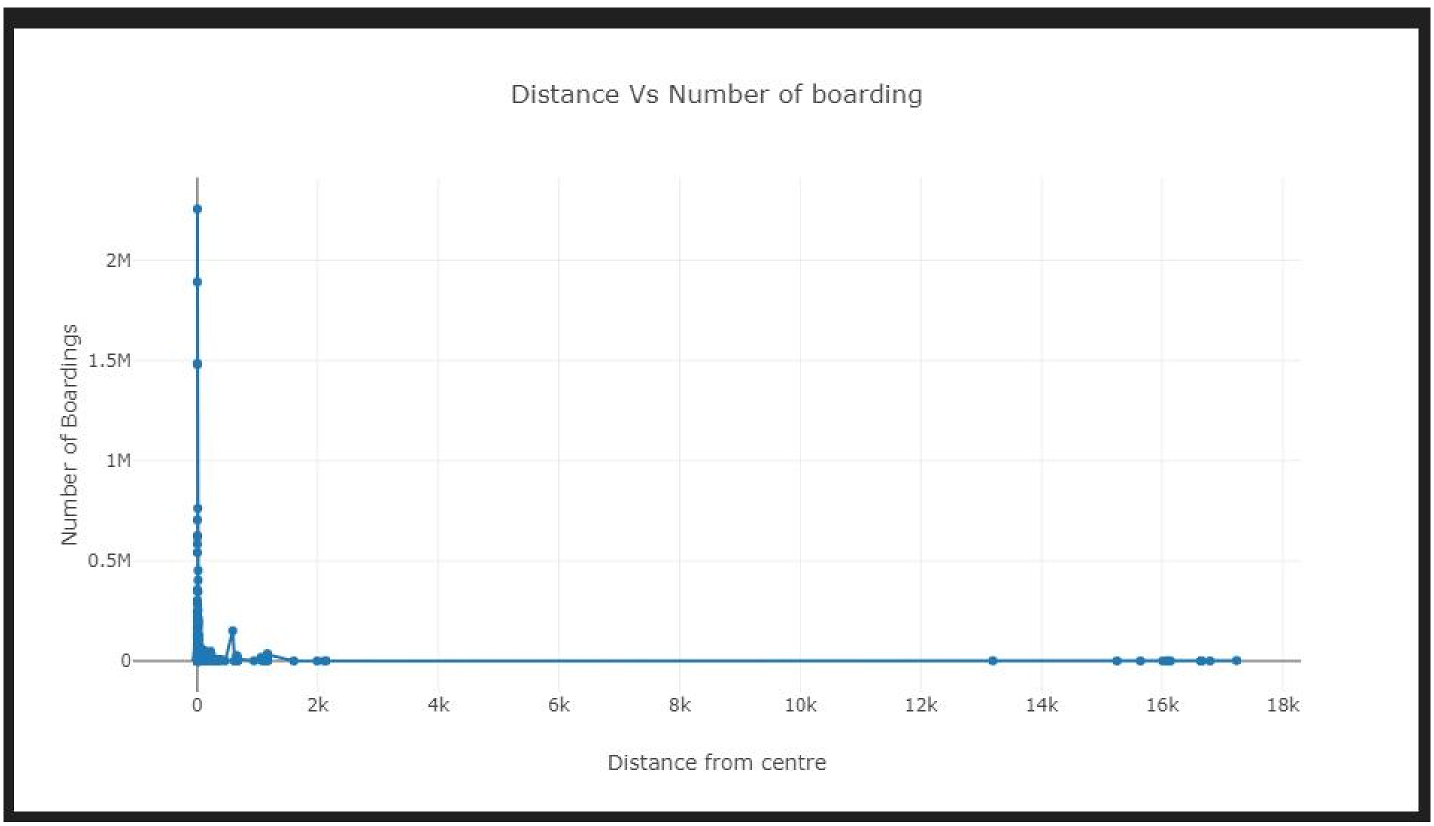
Data visualization:

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)
```

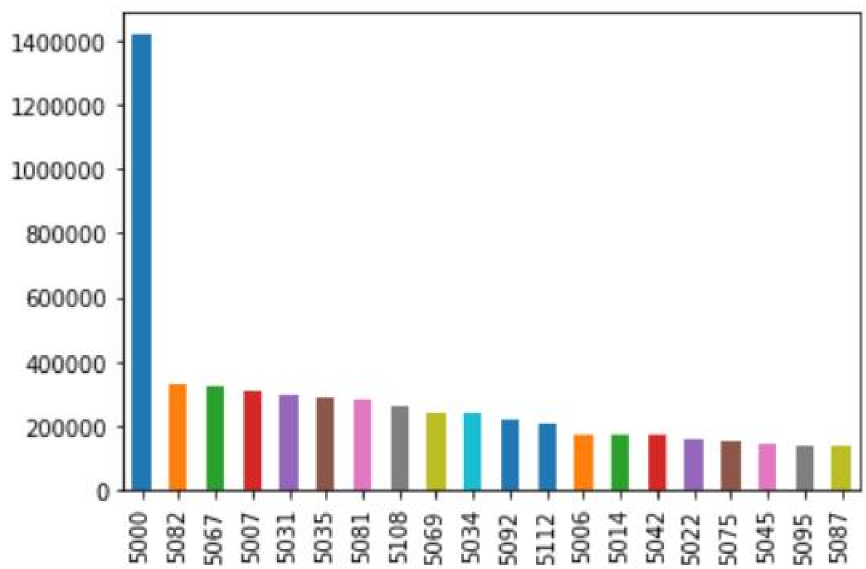
Output:



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()
```

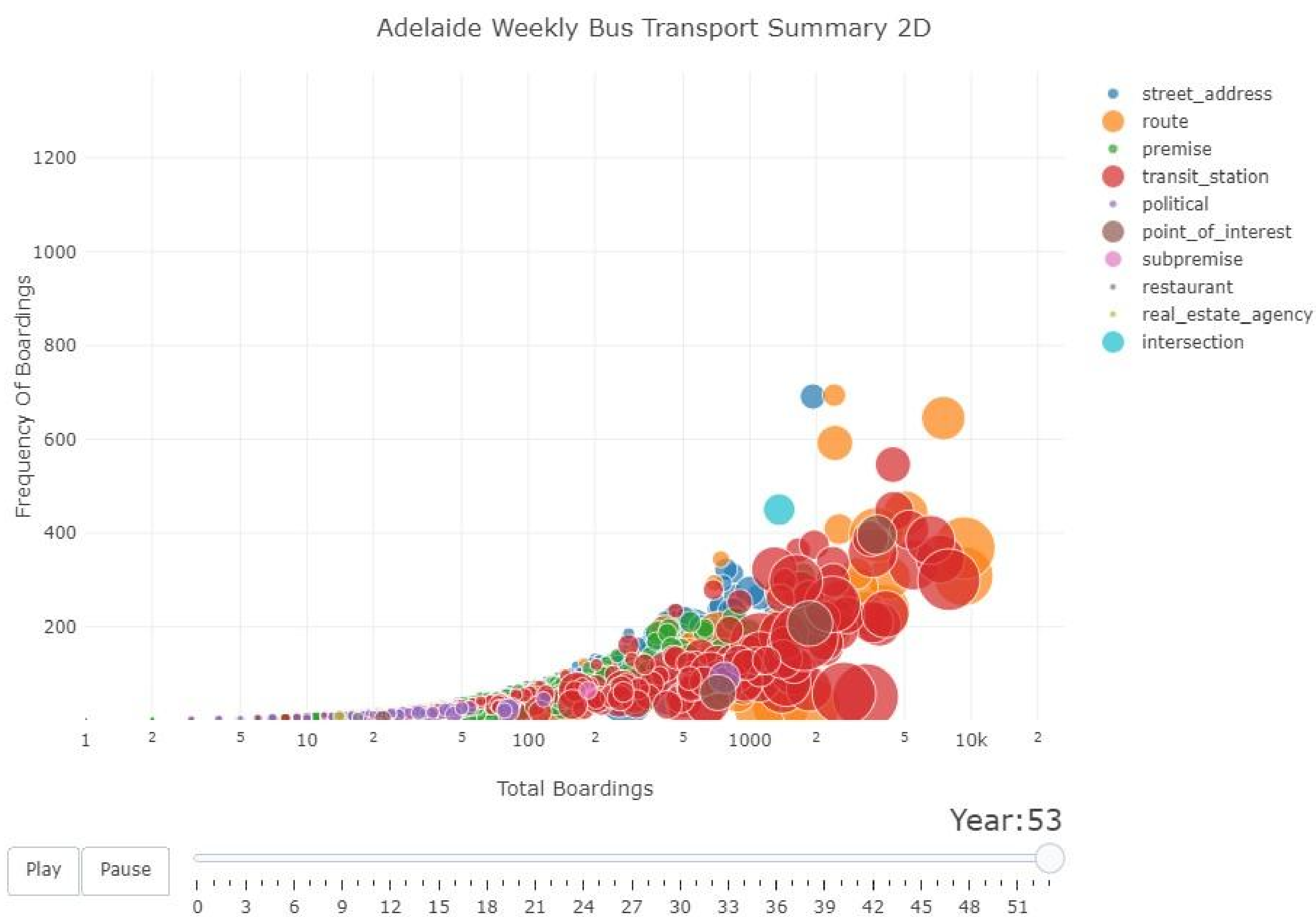
In [44]:
```
## Label encode the Date type for easy Plotting
le = LabelEncoder()
bb1['WeekBeginning'] = le.fit_transform(bb1['WeekBeginning'])
```

2D Plot with 6 different variables:

```
figure = bubbleplot(dataset=bb1, x_column='NumberOfBoardings_sum', y_column='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=True,
    title='Adelaide Weekly Bus Transport Summary 2D',x_logscale=True, scale_bubble=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)

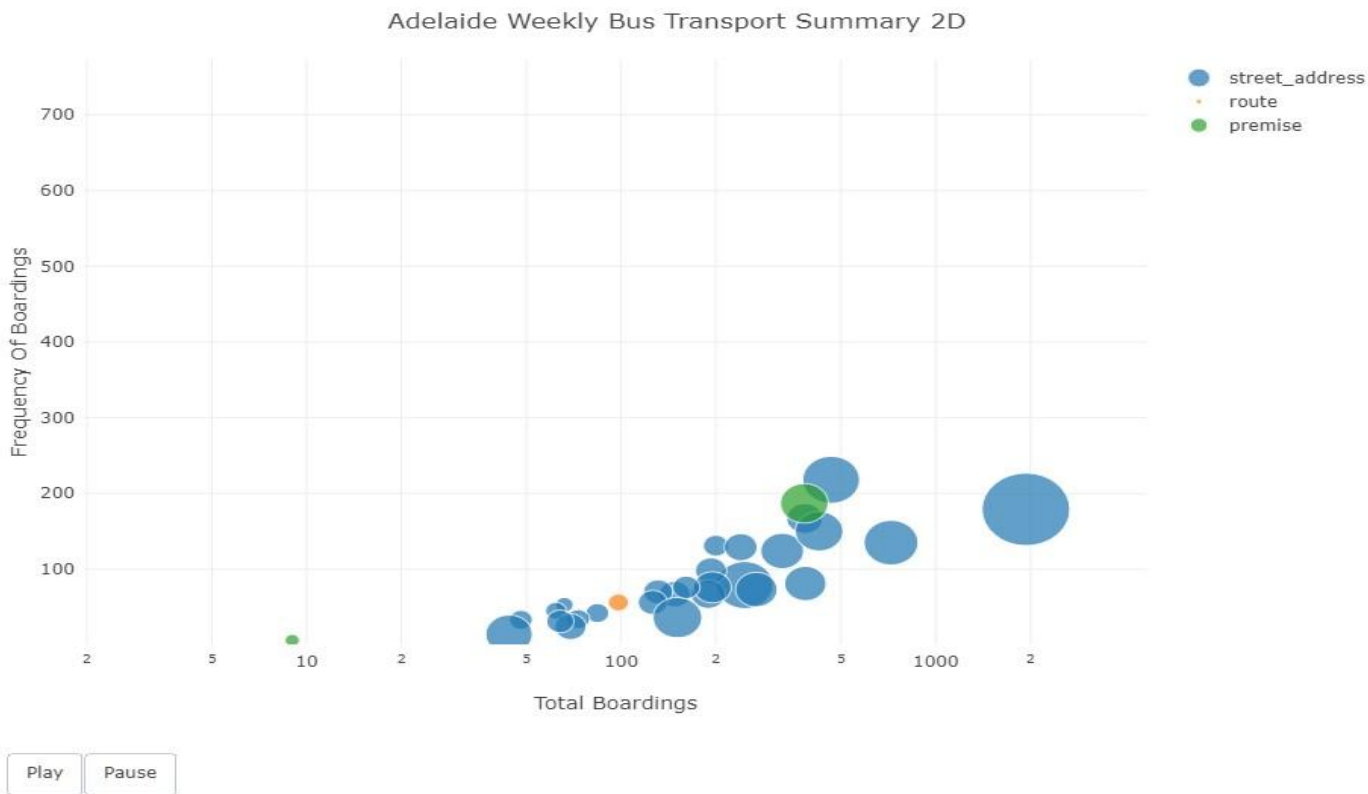
Plot for first 30 stops:

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:



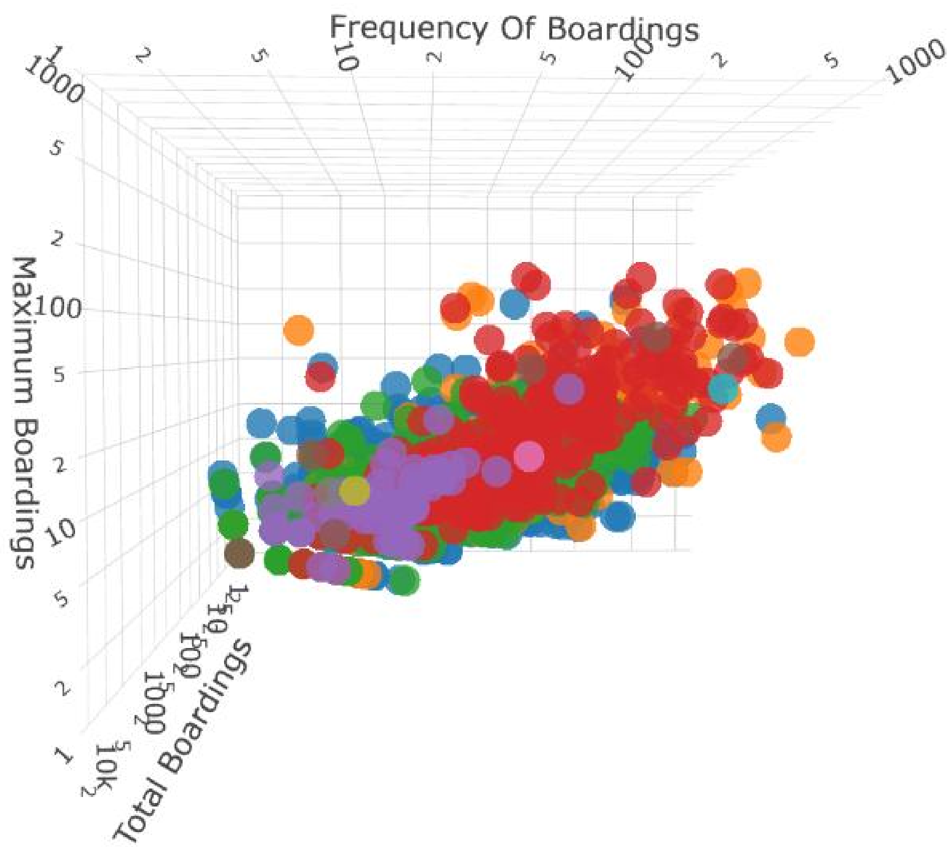
3D Bubble Plot with 6 different variables & there Relationship:

Program:

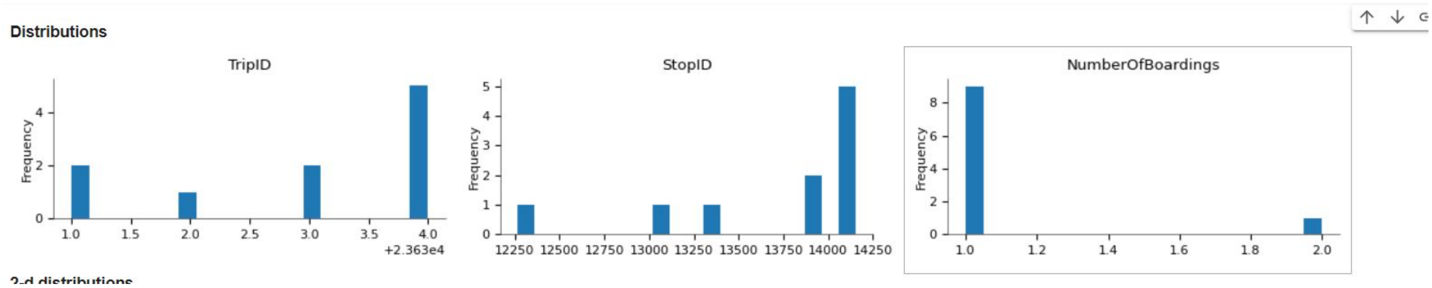
```
figure = bubbleplot(dataset=bb1, x_column='NumberOfBoardings_sum', y_column='NumberOfBoardings_count',
    bubble_column='StopName', time_column='WeekBeginning', z_column='NumberOfBoardings_max',
    color_column='type',show_slider=False,
    x_title="Total Boardings", y_title="Frequency Of Boardings", 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})
```

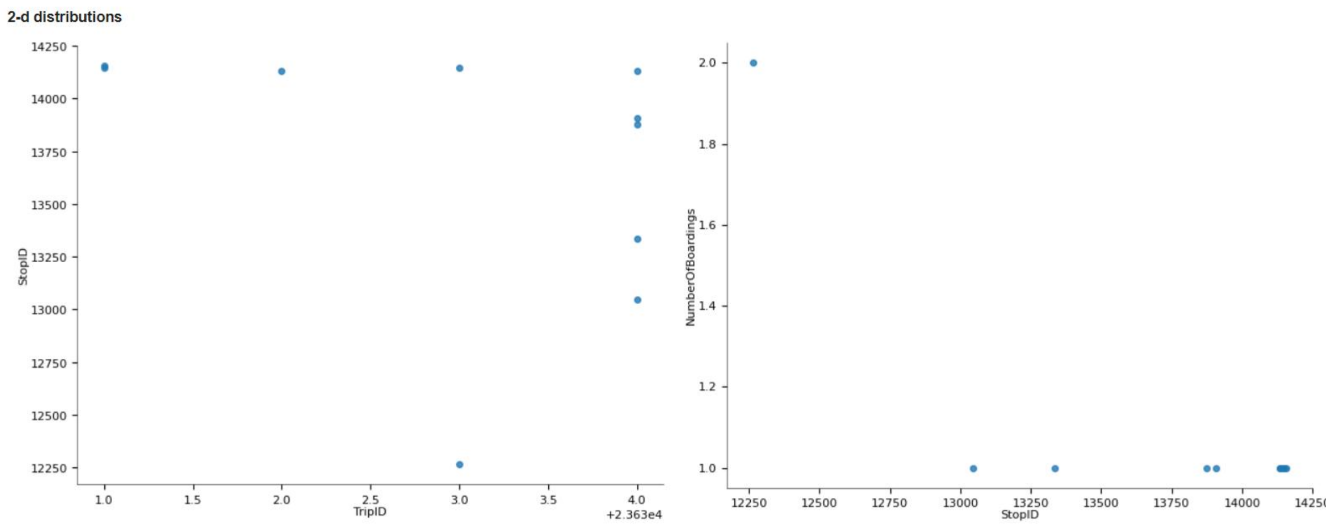
Output:



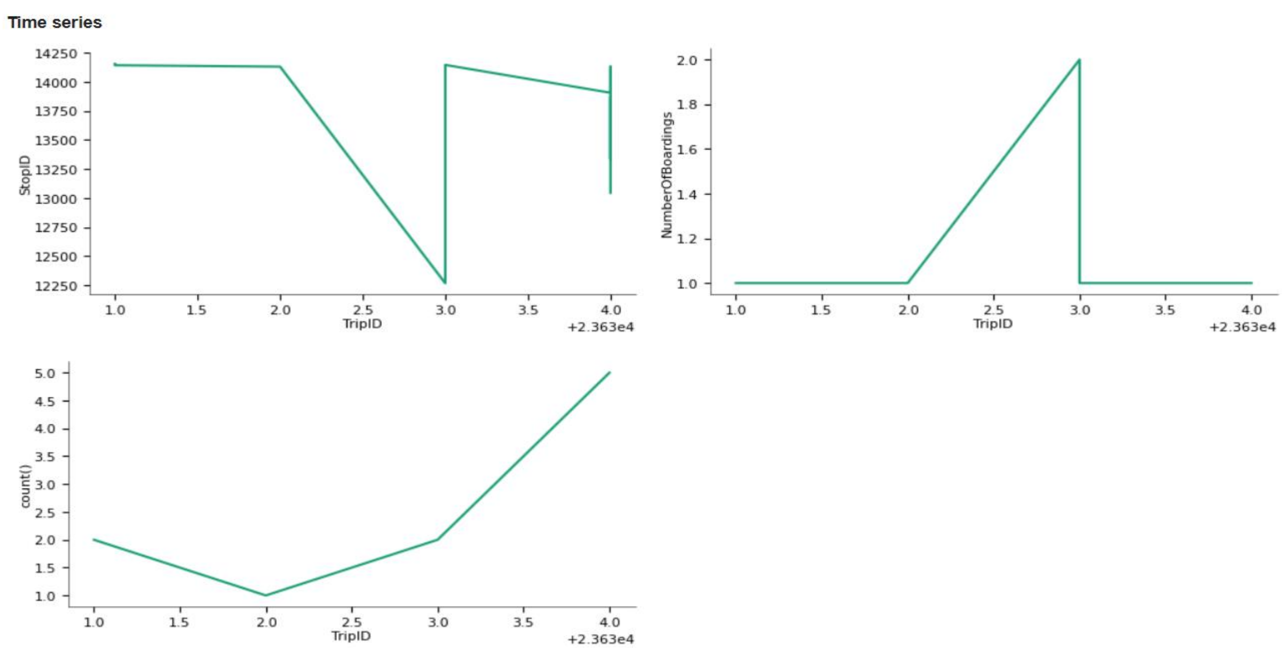
Distributions:



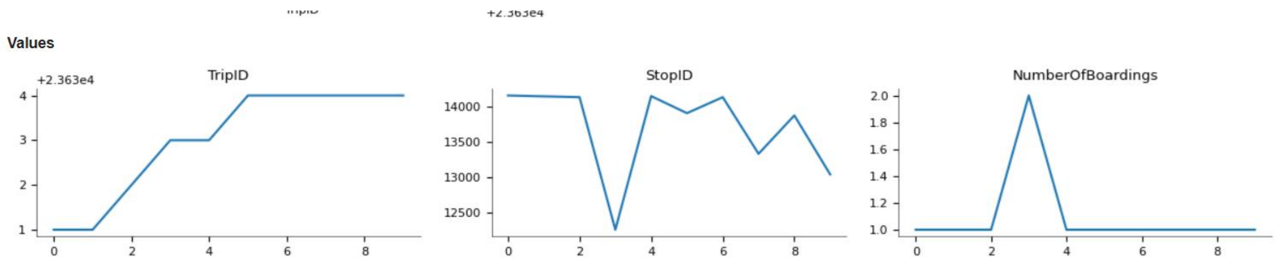
2-d distrubtions:



Time series:



Values:



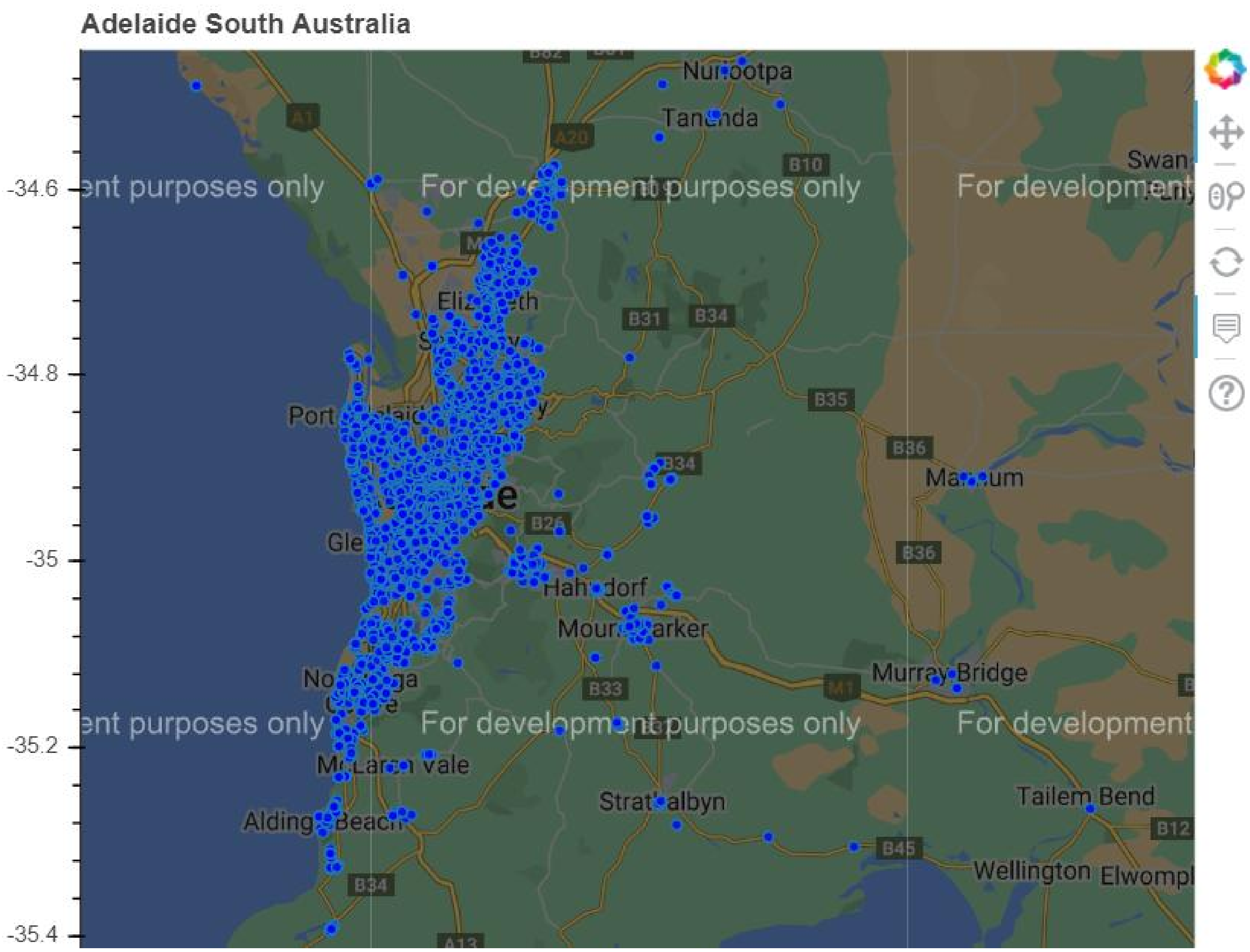
Using Bokeh:

```
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:



Conclusion:

In conclusion, the initial stages of our project, focused on building a comprehensive public transportation efficiency analysis through IBM Cognos for visualization, have been successfully initiated. By meticulously defining our analysis objectives and diligently collecting transportation data from the provided source, we have laid a strong foundation for our research. Equally critical has been the thorough process of cleaning and enhancing the collected dataset, ensuring its quality and accuracy. This crucial step not only mitigates potential data discrepancies but also establishes the reliability of our findings, setting the stage for a robust and insightful analysis of public transportation efficiency.