Public Transportation Efficiency Analysis

510421205059

PHASE-3

DEVELOPMENT PART 1

Public Transportation Efficiency Analysis

Introduction

This document provides a structured guide for building a Public Transportation Efficiency Analysis. The analysis aims to understand the trend of Public Transport over time and to compare Public Transport Analysis across different regions or countries.

Step 1: Define Analysis Objectives

Clearly outline the objectives of your Public Transportation Efficiency Analysis. Some example objectives include:

- **Trend Analysis:** Understand the trend of Public Transportation Efficiency over time.
- **Regional Comparison:** Compare Public Transportation statistics across different regions or countries.
- **Correlation Analysis:** Identify factors that may correlate with the spread of the virus.

Step 2: Obtain Public Transportation Data

Acquire a reliable and up-to-date Public Transportation dataset.

Step 3: Data Cleaning and Processing

Ensure the data is clean, accurate, and well-structured before loading it into IBM Cognos.

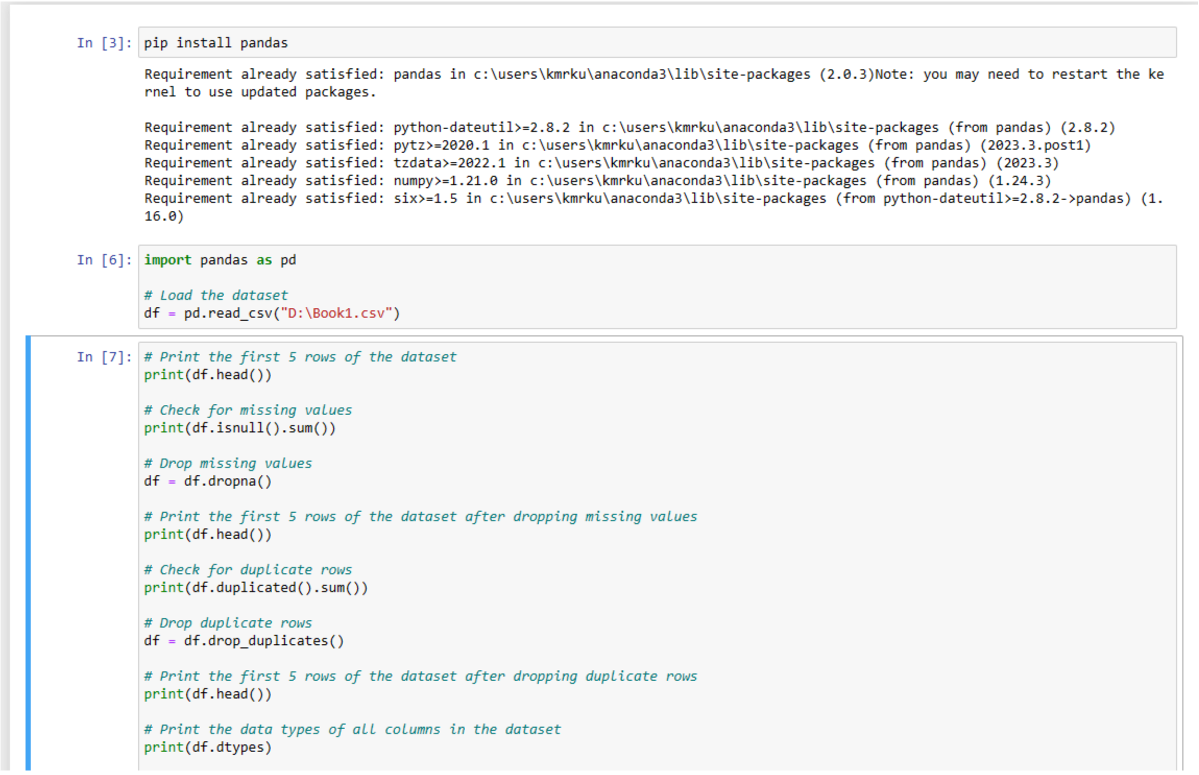
- **Handle Missing Values:** Identify and decide how to handle missing values (remove, impute, etc.).
- **Data Types:** Ensure that data types are appropriate (e.g., date fields should be in datetime format).
- **Remove Duplicates:** Eliminate duplicate records from the dataset.
- **Data Transformation:** Perform necessary transformations (aggregations, filtering, create calculated fields).

In this session we have used Jupyter Notebook to clean the data.

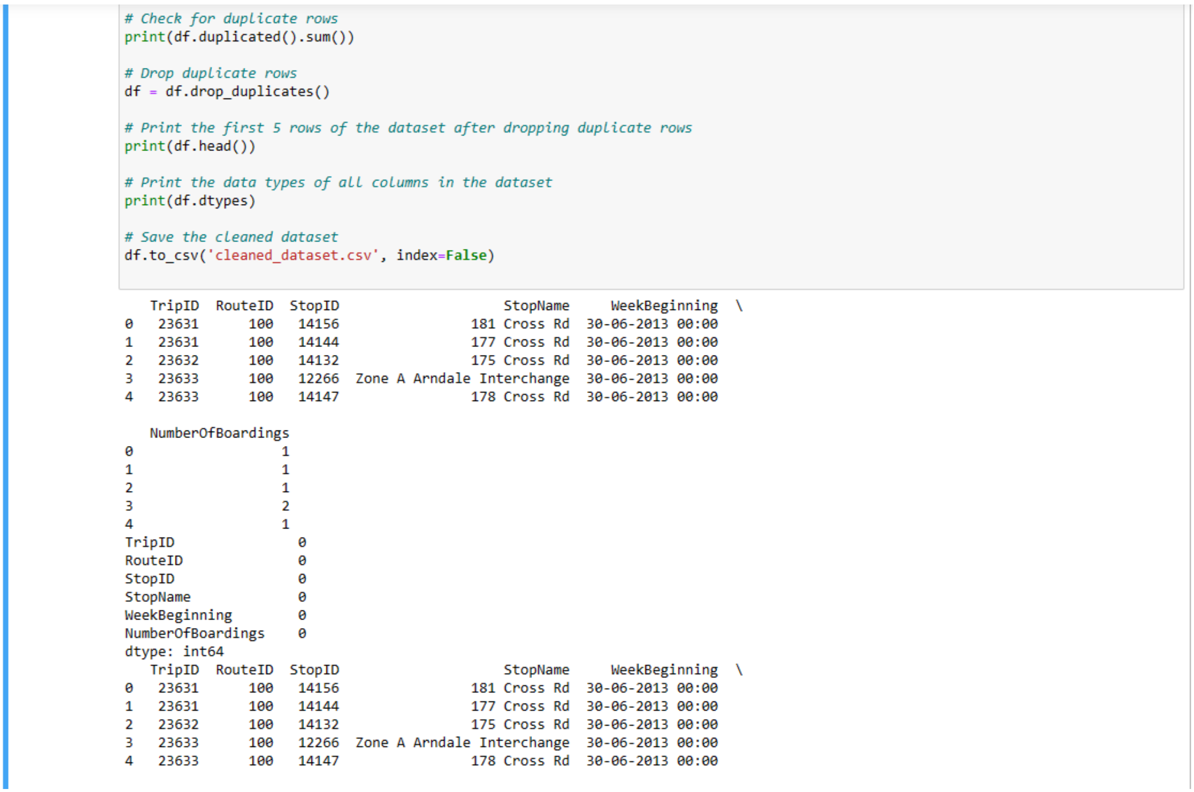
Python code for cleaning dataset:

```
import pandas as pd

# Load the dataset
df = pd.read_csv('dataset.csv')

# Print the first 5 rows of the dataset
print(df.head())

# Check for missing values
print(df.isnull().sum())

# Drop missing values
df = df.dropna()
```

```
# Print the first 5 rows of the dataset after dropping missing values
print(df.head())

# Check for duplicate rows
print(df.duplicated().sum())

# Drop duplicate rows
df = df.drop_duplicates()

# Print the first 5 rows of the dataset after dropping duplicate rows
print(df.head())

# Print the data types of all columns in the dataset
print(df.dtypes)

# Save the cleaned dataset
df.to_csv('cleaned_dataset.csv', index=False)
```

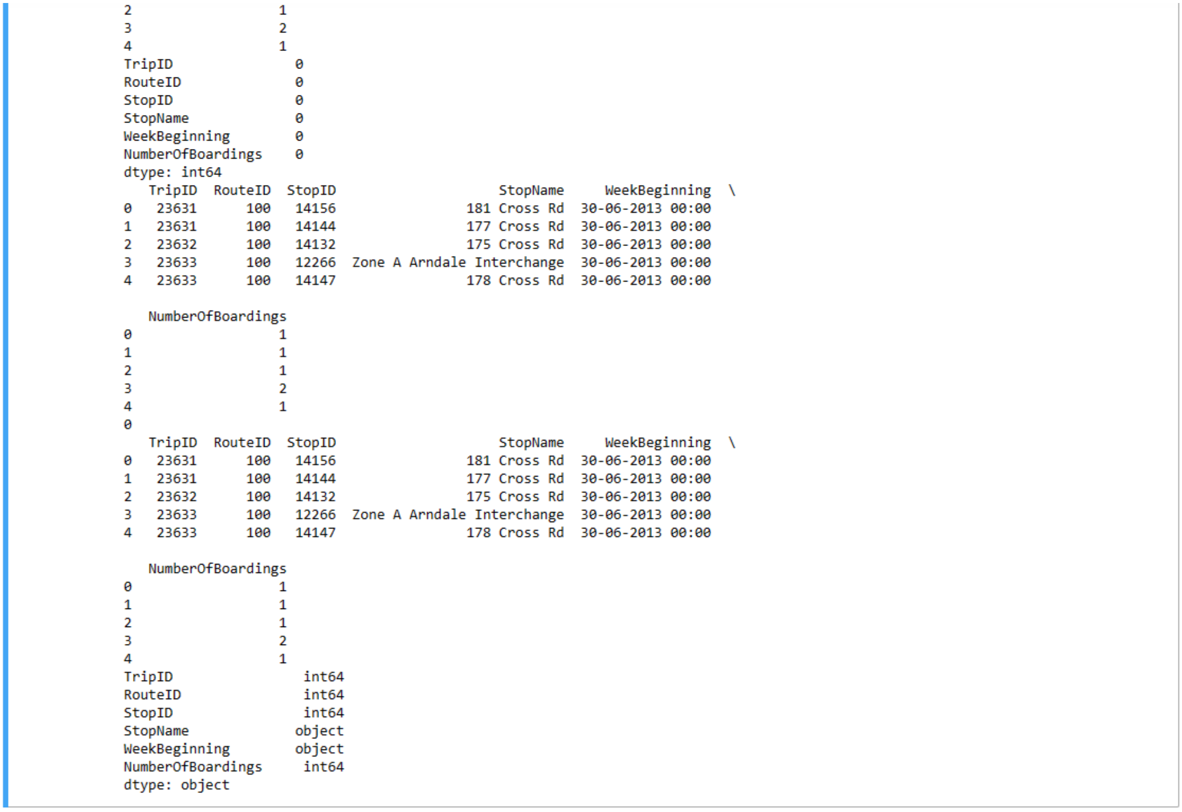


```
In [3]: pip install pandas

        Requirement already satisfied: pandas in c:\users\kmrku\anaconda3\lib\site-packages (2.0.3)Note: you may need to restart the ke
        rnel to use updated packages.

        Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\kmrku\anaconda3\lib\site-packages (from pandas) (2.8.2)
        Requirement already satisfied: pytz>=2020.1 in c:\users\kmrku\anaconda3\lib\site-packages (from pandas) (2023.3.post1)
        Requirement already satisfied: tzdata>=2022.1 in c:\users\kmrku\anaconda3\lib\site-packages (from pandas) (2023.3)
        Requirement already satisfied: numpy>=1.21.0 in c:\users\kmrku\anaconda3\lib\site-packages (from pandas) (1.24.3)
        Requirement already satisfied: six>=1.5 in c:\users\kmrku\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.
        16.0)

In [6]: import pandas as pd

        # Load the dataset
        df = pd.read_csv("D:\Book1.csv")

In [7]: # Print the first 5 rows of the dataset
        print(df.head())

        # Check for missing values
        print(df.isnull().sum())

        # Drop missing values
        df = df.dropna()

        # Print the first 5 rows of the dataset after dropping missing values
        print(df.head())

        # Check for duplicate rows
        print(df.duplicated().sum())

        # Drop duplicate rows
        df = df.drop_duplicates()

        # Print the first 5 rows of the dataset after dropping duplicate rows
        print(df.head())

        # Print the data types of all columns in the dataset
        print(df.dtypes)
```



```
# Check for duplicate rows
print(df.duplicated().sum())

# Drop duplicate rows
df = df.drop_duplicates()

# Print the first 5 rows of the dataset after dropping duplicate rows
print(df.head())

# Print the data types of all columns in the dataset
print(df.dtypes)

# Save the cleaned dataset
df.to_csv('cleaned_dataset.csv', index=False)
```

```
   TripID  RouteID  StopID                StopName     WeekBeginning  \
0   23631      100   14156             181 Cross Rd  30-06-2013 00:00
1   23631      100   14144             177 Cross Rd  30-06-2013 00:00
2   23632      100   14132             175 Cross Rd  30-06-2013 00:00
3   23633      100   12266  Zone A Arndale Interchange  30-06-2013 00:00
4   23633      100   14147             178 Cross Rd  30-06-2013 00:00

   NumberOfBoardings
0                  1
1                  1
2                  1
3                  2
4                  1
TripID               0
RouteID              0
StopID               0
StopName             0
WeekBeginning        0
NumberOfBoardings    0
dtype: int64
   TripID  RouteID  StopID                StopName     WeekBeginning  \
0   23631      100   14156             181 Cross Rd  30-06-2013 00:00
1   23631      100   14144             177 Cross Rd  30-06-2013 00:00
2   23632      100   14132             175 Cross Rd  30-06-2013 00:00
3   23633      100   12266  Zone A Arndale Interchange  30-06-2013 00:00
4   23633      100   14147             178 Cross Rd  30-06-2013 00:00
```



```
2                  1
3                  2
4                  1
TripID               0
RouteID              0
StopID               0
StopName             0
WeekBeginning        0
NumberOfBoardings    0
dtype: int64
   TripID  RouteID  StopID                StopName     WeekBeginning  \
0   23631      100   14156             181 Cross Rd  30-06-2013 00:00
1   23631      100   14144             177 Cross Rd  30-06-2013 00:00
2   23632      100   14132             175 Cross Rd  30-06-2013 00:00
3   23633      100   12266  Zone A Arndale Interchange  30-06-2013 00:00
4   23633      100   14147             178 Cross Rd  30-06-2013 00:00

   NumberOfBoardings
0                  1
1                  1
2                  1
3                  2
4                  1
0
   TripID  RouteID  StopID                StopName     WeekBeginning  \
0   23631      100   14156             181 Cross Rd  30-06-2013 00:00
1   23631      100   14144             177 Cross Rd  30-06-2013 00:00
2   23632      100   14132             175 Cross Rd  30-06-2013 00:00
3   23633      100   12266  Zone A Arndale Interchange  30-06-2013 00:00
4   23633      100   14147             178 Cross Rd  30-06-2013 00:00

   NumberOfBoardings
0                  1
1                  1
2                  1
3                  2
4                  1
TripID                int64
RouteID               int64
StopID                int64
StopName             object
WeekBeginning        object
NumberOfBoardings     int64
dtype: object
```

Step 4: Load Data into IBM Cognos

- **Open IBM Cognos Analytics:** Log in to IBM Cognos Analytics.
- **Create a Data Module or Connect to Data Source:** Create a new data module. Connect to your data source directly.
- **Import Data:** Import the cleaned Public Transportation data into IBM Cognos.

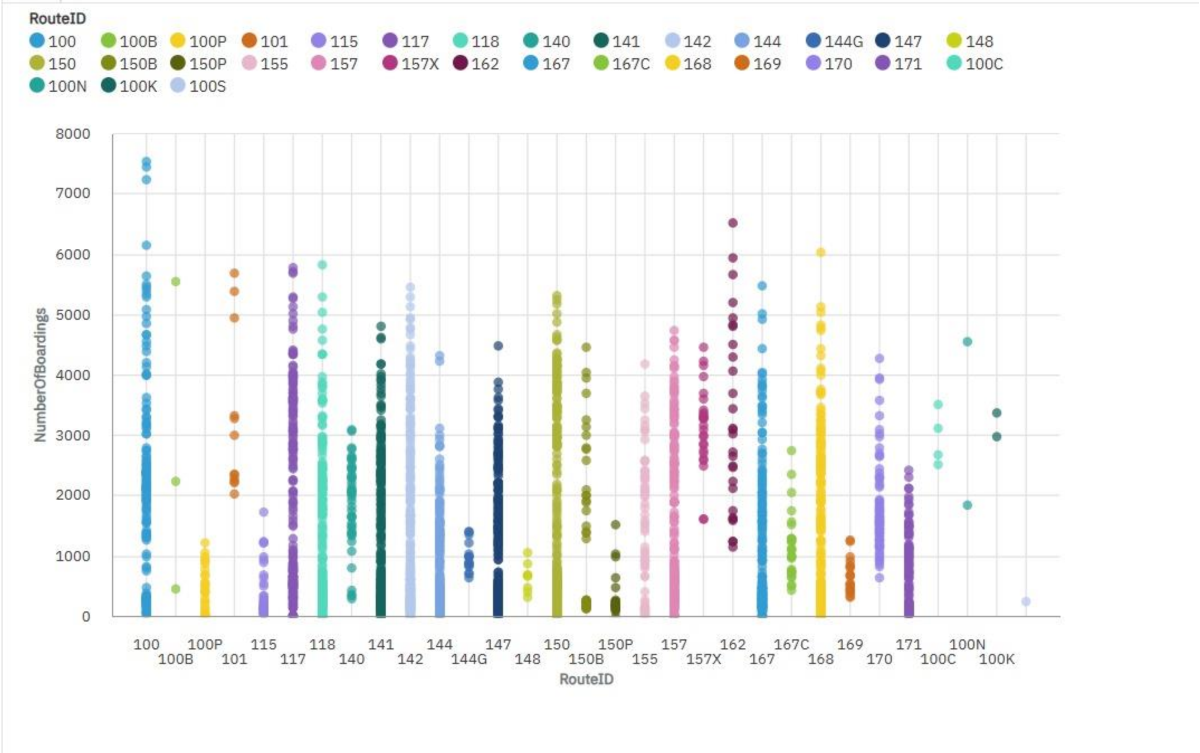
Step 5: Build Visualizations

- **Create Reports and Dashboards:** Build reports and dashboards in IBM Cognos Analytics.
- **Choose Appropriate Chart Types:** Select chart types based on your analysis objectives (line charts, bar charts, maps, etc.).
- **Use Features like Filtering and Grouping:** Utilize features such as filtering, grouping, and drilling down for deeper insights.

Step 6: Analyse and Share

- **Conduct Analysis:** Analyse the visualizations to answer your defined objectives.
- **Share Findings:** Share your findings through interactive dashboards or static reports.
- **Schedule Data Refresh:** Consider scheduling or automating the data refresh process for up-to-date insights.

Data Visualization and Analysis



Conclusion

In conclusion, this guide provides a systematic approach to initiate a comprehensive Public Transportation analysis using IBM Cognos for effective visualization. The outlined steps aim to facilitate the understanding of both temporal trends and regional variations in Public Transportation statistics.