PUBLIC TRANSPORT EFFICIENCY ANALYSIS

Date	17/10/2023
Team ID	1281
Project Name	Public Transport Efficiency Analysis

```
import numpy as np
import pandas as pd
import os
for dirname, _, filenames in os.walk('dataset.csv'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

print("Load the dataset")
import pandas as pd
data = pd.read_csv('dataset.CSV', low_memory=False)
data.shape
data.head(30)
```

```
Load the dataset

    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
5    23634     100   13907                9A  Marion Rd  30-06-2013
00:00
6    23634     100   14132                 175 Cross Rd  30-06-2013
00:00
7    23634     100   13335              9A  Holbrooks Rd  30-06-2013
00:00
8    23634     100   13875                 9  Marion Rd  30-06-2013
00:00
9    23634     100   13045              206 Holbrooks Rd  30-06-2013
00:00
10   23635     100   13335              9A  Holbrooks Rd  30-06-2013
```

00:00
11 23635 100 13383 8A Marion Rd 30-06-2013
00:00
12 23635 100 13586 8D Marion Rd 30-06-2013
00:00
13 23635 100 12726 23 Findon Rd 30-06-2013
00:00
14 23635 100 13813 8K Marion Rd 30-06-2013
00:00
15 23635 100 14062 20 Cross Rd 30-06-2013
00:00
16 23636 100 12780 22A Crittenden Rd 30-06-2013

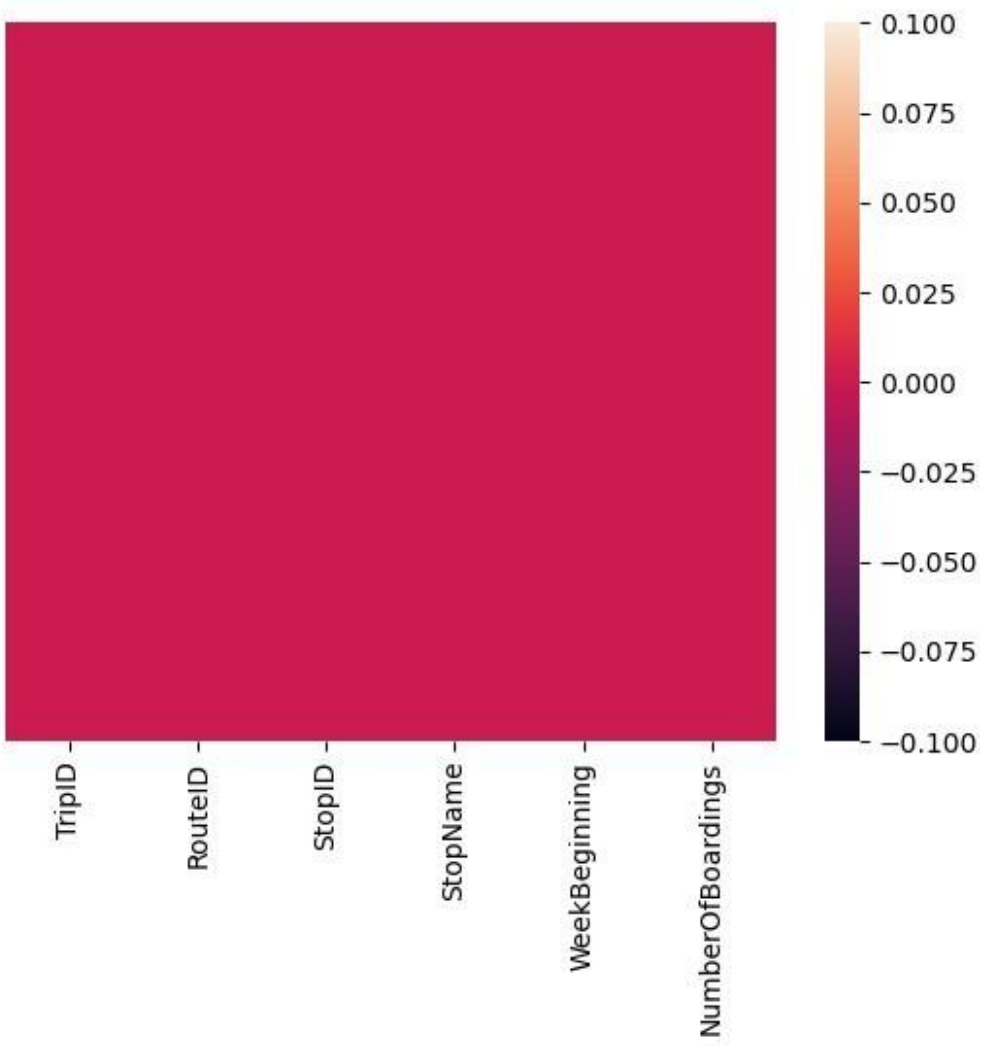
```
00:00
17   23636     100   13383               8A  Marion Rd  30-06-2013
00:00
18   23636     100   14154                180 Cross Rd  30-06-2013
00:00
19   23636     100   13524               8C  Marion Rd  30-06-2013
00:00
20   23636     100   14122                173 Cross Rd  30-06-2013
00:00
21   23636     100   13813               8K  Marion Rd  30-06-2013
00:00
22   23637     100   14156                181 Cross Rd  30-06-2013
00:00
23   23637     100   14154                180 Cross Rd  30-06-2013
00:00
24   23637     100   13335            9A  Holbrooks Rd  30-06-2013
00:00
25   23637     100   12266  Zone A Arndale Interchange  30-06-2013
00:00
26   23637     100   13196            13  Holbrooks Rd  30-06-2013
00:00
27   23638     100   12562               218 Findon Rd  30-06-2013
00:00
28   23638     100   12266  Zone A Arndale Interchange  30-06-2013
00:00
29   23638     100   13875                9  Marion Rd  30-06-2013
00:00
```

	NumberOfBoardings
0	1
1	1
2	1
3	2
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	2
13	1
14	1
15	1
16	1
17	1
18	2
19	3

```
20                   1
21                   1
22                   1
23                   1
24                   3
25                   5
26                   1
27                   1
28                   3
29                   1
```

```
data = data.drop_duplicates()
import seaborn as sns
sns.heatmap(data.isnull(),yticklabels= False)
print("\nCheck data types of columns")
print(data.dtypes)
```

```
Check data types of columns
TripID                int64
RouteID              object
StopID                int64
StopName             object
WeekBeginning        object
NumberOfBoardings     int64
dtype: object
```




```
data['RouteID'] = pd.to_numeric(data['RouteID'], errors='coerce')
print("Handle mixed data types")
print(data.dtypes)
```

```
Handle mixed data types
TripID                 int64
RouteID              float64
StopID                 int64
StopName              object
WeekBeginning         object
NumberOfBoardings      int64
dtype: object
```

```
data = data.dropna()
print("\nHandle missing values")
print(data.shape)
```

```
Handle missing values
(1008700, 6)
```

```
data['WeekBeginning'] = pd.to_datetime(data['WeekBeginning'],
errors='coerce')
print("\nConvert 'WeekBeginning' column to datetime format")
print(data['WeekBeginning'].head())
```

```
Convert 'WeekBeginning' column to datetime format
0   2013-06-30
1   2013-06-30
2   2013-06-30
3   2013-06-30
4   2013-06-30
Name: WeekBeginning, dtype: datetime64[ns]

C:\Users\bavik\AppData\Local\Temp\ipykernel_15464\2765944061.py:1:
UserWarning: Parsing dates in %d-%m-%Y %H:%M format when
dayfirst=False (the default) was specified. Pass `dayfirst=True` or
specify a format to silence this warning.
  data['WeekBeginning'] = pd.to_datetime(data['WeekBeginning'],
errors='coerce')
```

```
data['StopName'] = data['StopName'].str.strip()
print("\nClean 'StopName' column")
print(data['StopName'].head())
```

```
print(data.nunique())
```

```
TripID              3123
RouteID               20
StopID               963
StopName             577
WeekBeginning         54
NumberOfBoardings    156
dtype: int64
```

```
data.shape
data.columns
data.head(3)
```

```
   TripID  RouteID  StopID      StopName WeekBeginning
NumberOfBoardings
0   23631    100.0   14156  181 Cross Rd    2013-06-30
1
1   23631    100.0   14144  177 Cross Rd    2013-06-30
1
2   23632    100.0   14132  175 Cross Rd    2013-06-30
1
```

```
data.isnull().sum()
```

```
TripID                0
RouteID               0
StopID                0
StopName              0
WeekBeginning         0
NumberOfBoardings     0
dtype: int64
```

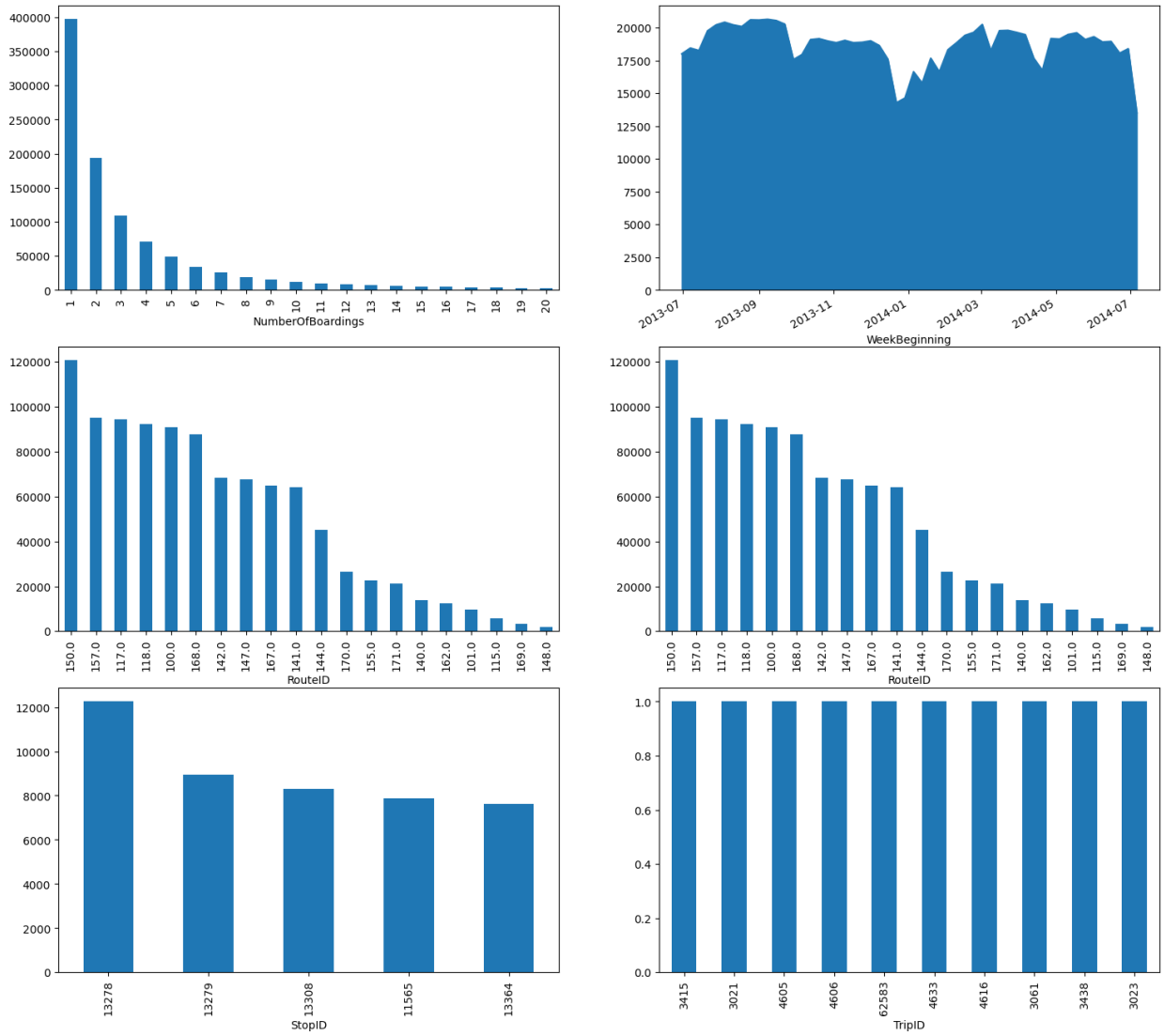
```
data['WeekBeginning'].unique()
```

```
<DatetimeArray>
['2013-06-30 00:00:00', '2013-07-07 00:00:00', '2013-07-14 00:00:00',
 '2013-07-21 00:00:00', '2013-07-28 00:00:00', '2013-08-04 00:00:00',
 '2013-08-11 00:00:00', '2013-08-18 00:00:00', '2013-08-25 00:00:00',
 '2013-09-01 00:00:00', '2013-09-08 00:00:00', '2013-09-15 00:00:00',
 '2013-09-22 00:00:00', '2013-09-29 00:00:00', '2013-10-06 00:00:00',
 '2013-10-13 00:00:00', '2013-10-20 00:00:00', '2013-10-27 00:00:00',
 '2013-11-03 00:00:00', '2013-11-10 00:00:00', '2013-11-17 00:00:00',
 '2013-11-24 00:00:00', '2013-12-01 00:00:00', '2013-12-08 00:00:00',
 '2013-12-15 00:00:00', '2013-12-22 00:00:00', '2013-12-29 00:00:00',
 '2014-01-05 00:00:00', '2014-01-12 00:00:00', '2014-01-19 00:00:00',
 '2014-01-26 00:00:00', '2014-02-02 00:00:00', '2014-02-09 00:00:00',
 '2014-02-16 00:00:00', '2014-02-23 00:00:00', '2014-03-02 00:00:00',
 '2014-03-09 00:00:00', '2014-03-16 00:00:00', '2014-03-23 00:00:00',
 '2014-03-30 00:00:00', '2014-04-06 00:00:00', '2014-04-13 00:00:00',
 '2014-04-20 00:00:00', '2014-04-27 00:00:00', '2014-05-04 00:00:00',
 '2014-05-11 00:00:00', '2014-05-18 00:00:00', '2014-05-25 00:00:00',
 '2014-06-01 00:00:00', '2014-06-08 00:00:00', '2014-06-15 00:00:00',
 '2014-06-22 00:00:00', '2014-06-29 00:00:00', '2014-07-06 00:00:00']
Length: 54, dtype: datetime64[ns]
```

```
import matplotlib.pyplot as plt
fig,axrr=plt.subplots(3,2,figsize=(18,18))
data['NumberOfBoardings'].value_counts().sort_index().head(20).plot.ba
r(ax=axrr[0][0])
data['WeekBeginning'].value_counts().plot.area(ax=axrr[0][1])
data['RouteID'].value_counts().head(20).plot.bar(ax=axrr[1][0])
data['RouteID'].value_counts().tail(20).plot.bar(ax=axrr[1][1])
```

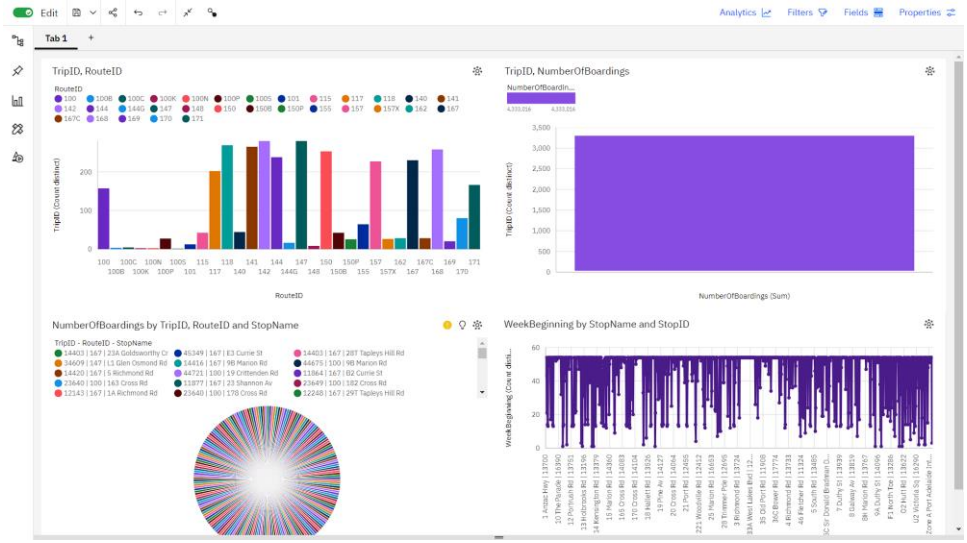
```
data['StopID'].value_counts().head(5).plot.bar(ax=axrr[2][0])
data['TripID'].value_counts().tail(10).plot.bar(ax=axrr[2][1])
```

```
<Axes: xlabel='TripID'>
```




```
data.to_csv('cleaned_data.csv', index=False)
print("\nSave the cleaned dataset to a new CSV file")
print("Cleaned dataset saved successfully.")
```

```
Save the cleaned dataset to a new CSV file
Cleaned dataset saved successfully.
```


Edit
Analytics
Filters
Fields
Properties
Tab 1
TripID, RouteID
RouteID
TripID (Count distinct)
RouteID
TripID, NumberOfBoardings
NumberOfBoardings (Sum)
NumberOfBoardings by TripID, RouteID and StopName
TripID - RouteID - StopName
WeekBeginning by StopName and StopID