PUBLIC TRANSPORTATION EFFICIENCY ANALYSIS DOCUMENTATION

Objective:

The objective of the "Public Transportation Efficiency Analysis " project is to evaluate the effectiveness and quality of public transportation services within a specific region. Through this analysis, we aim to identify areas for improvement, optimize transportation routes, and enhance overall service efficiency. By collecting and analyzing relevant data, our goal is to provide valuable insights and recommendations that can support the enhancement of public transportation infrastructure, making it more reliable, accessible, and convenient for the community. This project seeks to contribute to the improvement of public transportation systems, ultimately benefiting both passengers and the environment by encouraging the use of sustainable and efficient transit options.

Analysis Objective:

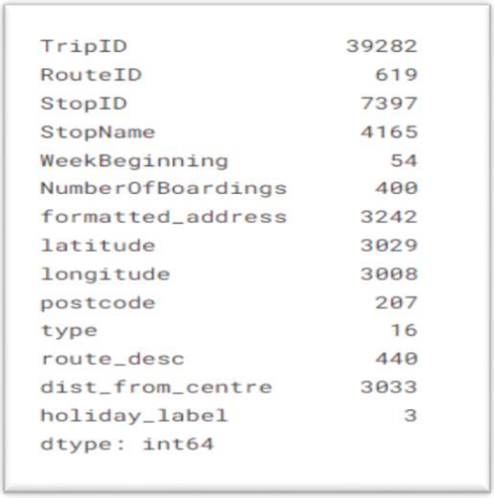
To collect the data of number of trips, what are the route from the user and collecting information from user such as starting point and ending point from the user and finally from the bus management getting the number of passengers from each trip with complete details

Data Collection:

THE DATASET WHICH IS PROVIDED IS → https://www.kaggle.com/datasets/rednivrug/unisys?select=20140711.CSV

The data fields in the given file are:

- **TripID**
- **RouteID**
- **StopID**
- **StopName**
- **WeekBeginning**
- **NumberOfBoarding**
- **Latitude Longitude**
- **Type**
- **PostCode**
- **RouteDesc**



```
TripID                 39282
RouteID                  619
StopID                  7397
StopName                4165
WeekBeginning             54
NumberOfBoardings        400
formatted_address       3242
latitude                3029
longitude               3008
postcode                 207
type                      16
route_desc               440
dist_from_centre        3033
holiday_label              3
dtype: int64
```

Data Visualization & Code Implementation

Tools/package	Purpose
python	Data Analysis
Python	Data Visualization

ransportation-efficiency-analysis

November 1, 2023

[1]:

```
import pandas as pd

data=pd.read_csv("C:/Users/abuba/Downloads/publicTransport.csv")
data
```

```
C:\Users\abuba\AppData\Local\Temp\ipykernel_5396\2646756992.py:3: DtypeWarning:
Columns (1) have mixed types. Specify dtype option on import or set
low_memory=False.
  data=pd.read_csv("C:/Users/abuba/Downloads/publicTransport.csv")
```

[1]:

```
          TripID RouteID  StopID                    StopName  \
0          23631     100   14156                181 Cross Rd
1          23631     100   14144                177 Cross Rd
2          23632     100   14132                175 Cross Rd
3          23633     100   12266  Zone A Arndale Interchange
4          23633     100   14147                178 Cross Rd
…            …     …      …                         …
10857229   13346    W91C   14629                21 Cashel St
10857230   13346    W91C   14708                22 Cashel St
10857231   13346    W91C   13709              2 Greenhill Rd
10857232   13346    W91C   14029                  10 East Av
10857233   13346    W91C   13824                 6 Leader St

                WeekBeginning  NumberOfBoardings
0         2013-06-30 00:00:00                  1
1         2013-06-30 00:00:00                  1
2         2013-06-30 00:00:00                  1
3         2013-06-30 00:00:00                  2
4         2013-06-30 00:00:00                  1
…                         …                …
10857229  2014-07-06 00:00:00                  1
10857230  2014-07-06 00:00:00                  3
10857231  2014-07-06 00:00:00                  1
10857232  2014-07-06 00:00:00                  1
10857233  2014-07-06 00:00:00                  1

[10857234 rows x 6 columns]
```

[2]:

```
data.head(10)
```

[2]:

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

   NumberOfBoardings
0                  1
1                  1
2                  1
3                  2
4                  1
5                  1
6                  1
7                  1
8                  1
9                  1
```

[3]:

```
data.shape
```

[3]:

```
(10857234, 6)
```

[4]:

```
data.columns
```

[4]:

```
Index(['TripID', 'RouteID', 'StopID', 'StopName', 'WeekBeginning',
       'NumberOfBoardings'],
      dtype='object')
```

[5]:

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

[5]:

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

[6]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10857234 entries, 0 to 10857233
Data columns (total 6 columns):
 #   Column             Dtype
---  ------             -----
 0   TripID             int64
 1   RouteID            object
 2   StopID             int64
 3   StopName           object
 4   WeekBeginning      object
 5   NumberOfBoardings  int64
dtypes: int64(3), object(3)
memory usage: 497.0+ MB
```

[7]:

```
df=data
```

[8]:

```
a=df.TripID.value_counts()
a
```

[8]:

```
57020    2819
57018    2741
27478    2733
57041    2718
57029    2691
         …
59297       1
3061        1
3414        1
3415        1
61163       1
Name: TripID, Length: 39282, dtype: int64
```

[9]:

```
b=df.RouteID.value_counts()
b
```

[9]:

```
G10     358005
B10     332694
M44     331442
H30     326004
300     228373
         …
FX1          1
FX10         1
FX8          1
FX3          1
FX2          1
Name: RouteID, Length: 619, dtype: int64
```

```
[10]: c=df.StopID.value_counts()
      c
```

```
[10]: 13354    44089
      13277    43339
      13364    43265
      13330    36992
      13279    33800
               …
      17107        1
      15420        1
      15243        1
      17805        1
      17807        1
      Name: StopID, Length: 7397, dtype: int64
```

```
[11]: d=df.WeekBeginning.value_counts()
      d
```

```
[11]: 2014-03-02 00:00:00    217162
      2014-05-18 00:00:00    215932
      2014-05-11 00:00:00    214947
      2014-06-01 00:00:00    213789
      2014-05-04 00:00:00    212681
      2014-03-23 00:00:00    212552
      2014-03-16 00:00:00    212188
      2014-02-23 00:00:00    212103
      2013-09-08 00:00:00    211914
      2014-04-27 00:00:00    211782
      2014-05-25 00:00:00    211534
      2014-03-30 00:00:00    211460
      2013-09-01 00:00:00    210968
      2014-04-06 00:00:00    210557
      2013-08-25 00:00:00    209497
      2013-11-17 00:00:00    209341
      2013-11-24 00:00:00    208881
      2013-10-20 00:00:00    208655
      2013-12-01 00:00:00    208470
      2014-06-15 00:00:00    208457
      2014-06-08 00:00:00    208417
      2013-09-15 00:00:00    208241
      2014-02-16 00:00:00    208178
      2013-10-27 00:00:00    207971
      2013-09-22 00:00:00    207769
      2013-12-08 00:00:00    207353
      2013-10-13 00:00:00    207351
      2013-08-04 00:00:00    207082
```

```
2013-11-03 00:00:00    206863
2013-11-10 00:00:00    206853
2014-06-29 00:00:00    206138
2013-07-28 00:00:00    205492
2013-08-11 00:00:00    205385
2013-08-18 00:00:00    203852
2013-07-21 00:00:00    201257
2014-06-22 00:00:00    200950
2014-02-09 00:00:00    197978
2014-01-19 00:00:00    196344
2013-10-06 00:00:00    195830
2014-03-09 00:00:00    195200
2013-12-15 00:00:00    194102
2014-02-02 00:00:00    192507
2013-09-29 00:00:00    192023
2013-07-07 00:00:00    190543
2014-04-13 00:00:00    190060
2013-07-14 00:00:00    187192
2014-01-05 00:00:00    186105
2014-04-20 00:00:00    185080
2013-06-30 00:00:00    182229
2014-01-26 00:00:00    180259
2014-01-12 00:00:00    178456
2013-12-29 00:00:00    168771
2013-12-22 00:00:00    163331
2014-07-06 00:00:00    149202
Name: WeekBeginning, dtype: int64
```

[12]:

```
e=df.NumberOfBoardings.value_counts()
e
```

[12]:

```
1      4270812
2      2057245
3      1128820
4       731537
5       502763
         …
547          1
539          1
443          1
474          1
342          1
Name: NumberOfBoardings, Length: 400, dtype: int64
```

[13]:

```
data['WeekBeginning'] = pd.to_datetime(data['WeekBeginning']).dt.date
data['WeekBeginning'][1]
```

```
[13]: datetime.date(2013, 6, 30)
```

```
[14]: grouped = data.groupby(['StopName','WeekBeginning',]).agg({'NumberOfBoardings':␣
       ↪['sum', 'count','max']})
      grouped
```

```
[14]:                                                  NumberOfBoardings
                                                                    sum count max
      StopName                     WeekBeginning
      1 Anzac Hwy                  2013-06-30                      1003   378  51
                                   2013-07-07                       783   360  28
                                   2013-07-14                       843   343  45
                                   2013-07-21                       710   356  28
                                   2013-07-28                       898   379  41
      …                                                               …     …  ..
      Zone I Salisbury Interchange 2014-06-08                       822   117  44
                                   2014-06-15                       965   113  39
                                   2014-06-22                       896   111  58
                                   2014-06-29                      1052   113  39
                                   2014-07-06                       534    90  21

      [207864 rows x 3 columns]
```

```
[15]: st_week_grp = pd.DataFrame(grouped).reset_index()
      st_week_grp1 = pd.DataFrame(st_week_grp.groupby('StopName')["WeekBeginning"].
       ↪count()).reset_index()
      st_week_grp1.head()
```

```
[15]:          StopName  WeekBeginning
      0      1 Anzac Hwy             54
      1     1 Bartels Rd             54
      2     1 Botanic Rd             54
      3       1 Frome Rd             54
      4   1 Fullarton Rd             54
```

```
[16]: stopListName = list(st_week_grp1[st_week_grp1['WeekBeginning'] ==␣
       ↪54]['StopName'])
      stopListName[1:30]
```

```
[16]: ['1 Bartels Rd',
       '1 Botanic Rd',
       '1 Frome Rd',
       '1 Fullarton Rd',
       '1 George St',
       '1 Glen Osmond Rd',
       '1 Goodwood Rd',
       '1 Henley Beach Rd',
       '1 Kensington Rd',
       '1 King William Rd',
       '1 Port Rd',
       '1 Sir Donald Bradman Dr',
       '1 Sir Edwin Smith Av',
       '1 Unley Rd',
       '10  Holbrooks Rd',
       '10  Marion Rd',
       '10  Portrush Rd',
       '10 Airport Rd',
       '10 Anzac Hwy',
       '10 Ashley St',
       '10 Belair Rd',
       '10 Churchill Rd',
       '10 East Av',
       '10 Fullarton Rd',
       '10 Garden Tce',
       '10 Glen Osmond Rd',
       '10 Goodwood Rd',
       '10 Greenhill Rd',
       '10 Harrow Tce']
```

[17]:

```
stopageName_with_boarding = data.groupby(['StopName']).agg({'NumberOfBoardings':
 ↪ ['sum']}).reset_index()
```

[18]:

```
stopageName_with_boarding.columns = ["stopName","Total_No_of_boardings"]
stopageName_with_boarding.head()
```

[18]:

```
          stopName  Total_No_of_boardings
0      1 Anzac Hwy                  39429
1    1 Bartels Rd                    8412
2    1 Botanic Rd                   14868
3      1 Frome Rd                   67458
4  1 Fullarton Rd                     585
```

[19]:

```
stopageName_with_boarding = stopageName_with_boarding.
 ↪sort_values("Total_No_of_boardings", ascending = False)
#stopage with most no of boarding
stopageName_with_boarding.head(10)
```

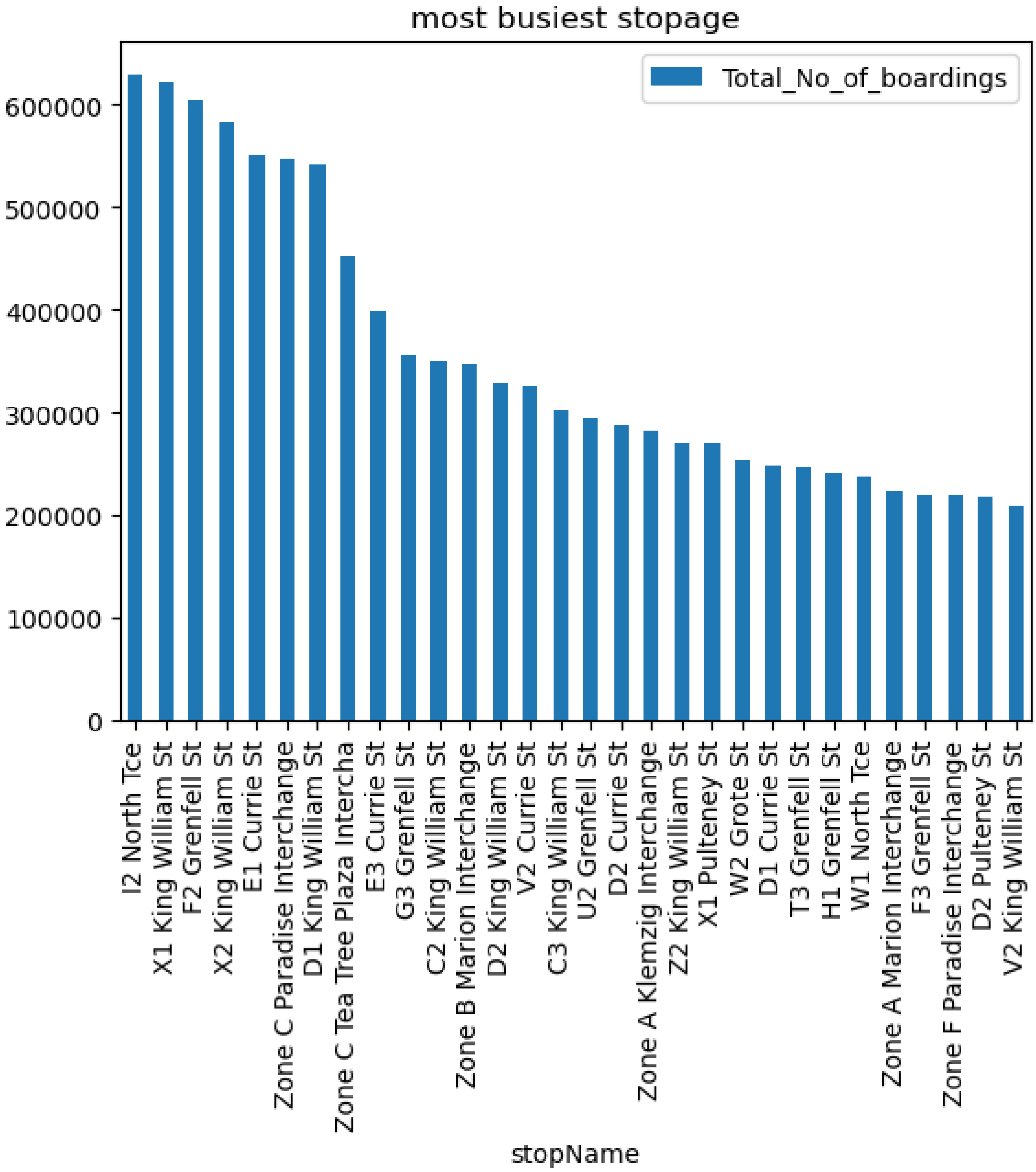
[19]:

```
                            stopName  Total_No_of_boardings
3841                    I2 North Tce                 628859
4023               X1 King William St                 622099
3807                   F2 Grenfell St                 604149
4029               X2 King William St                 583227
3791                    E1 Currie St                  550396
4120      Zone C Paradise Interchange                 547709
```

```
3784              D1 King William St                 541046
4124  Zone C Tea Tree Plaza Intercha                 451960
3796                      E3 Currie St                399351
3819                    G3 Grenfell St                356518
```

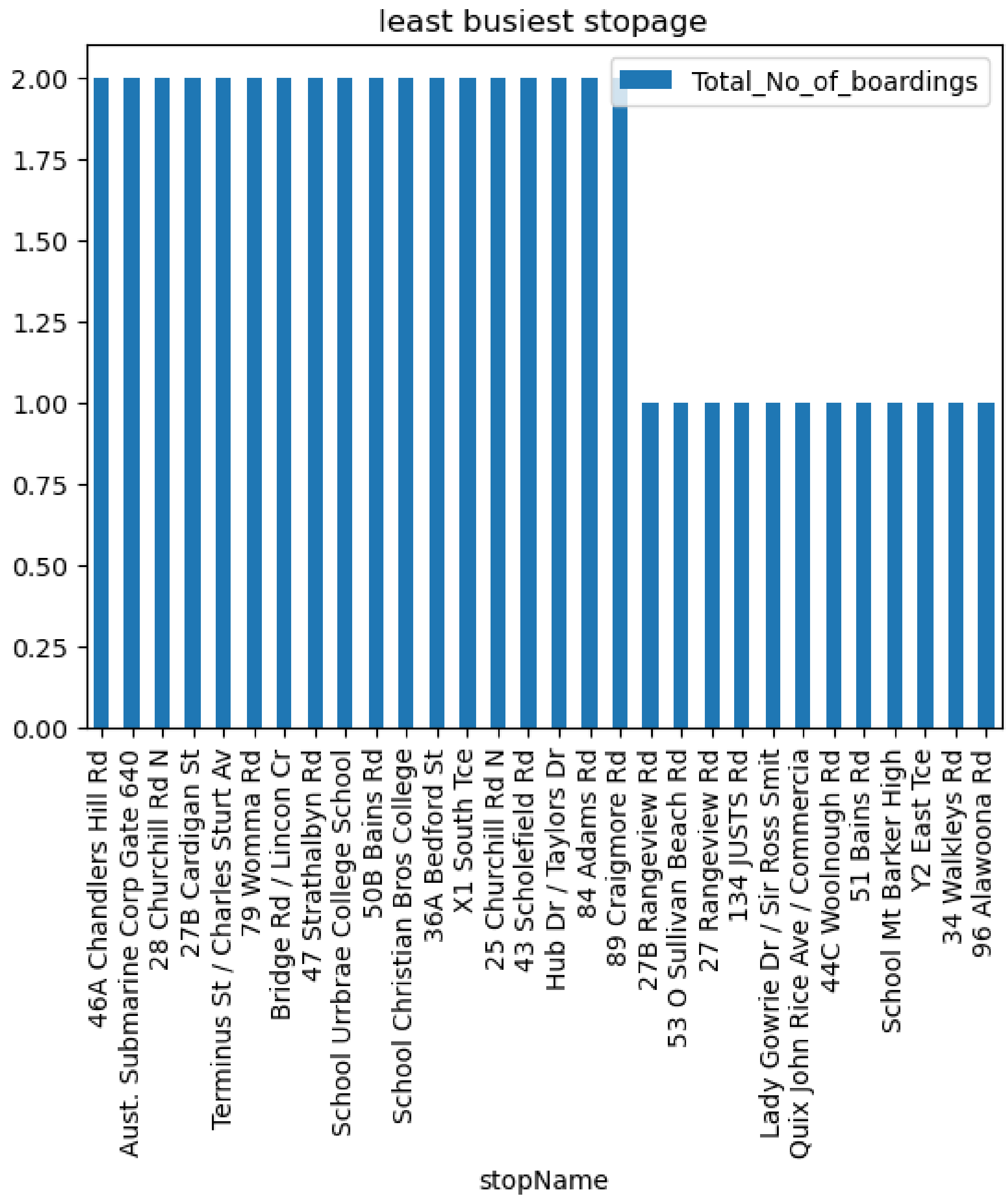
```
[20]: import matplotlib.pyplot as plt
      busiestStop = stopageName_with_boarding.head(30).plot.bar(x="stopName",␣
       ↪y="Total_No_of_boardings", rot=90)
      plt.title("most busiest stopage")
      plt.legend()
```

```
[20]: <matplotlib.legend.Legend at 0x2c0194fe350>
```



```
[21]: leastBusiestStop = stopageName_with_boarding.tail(30).plot.bar(x='stopName',␣
        ↪y='Total_No_of_boardings', rot=90)
      plt.title("least busiest stopage")
      plt.legend()
```

```
[21]: <matplotlib.legend.Legend at 0x2c01944b5d0>
```

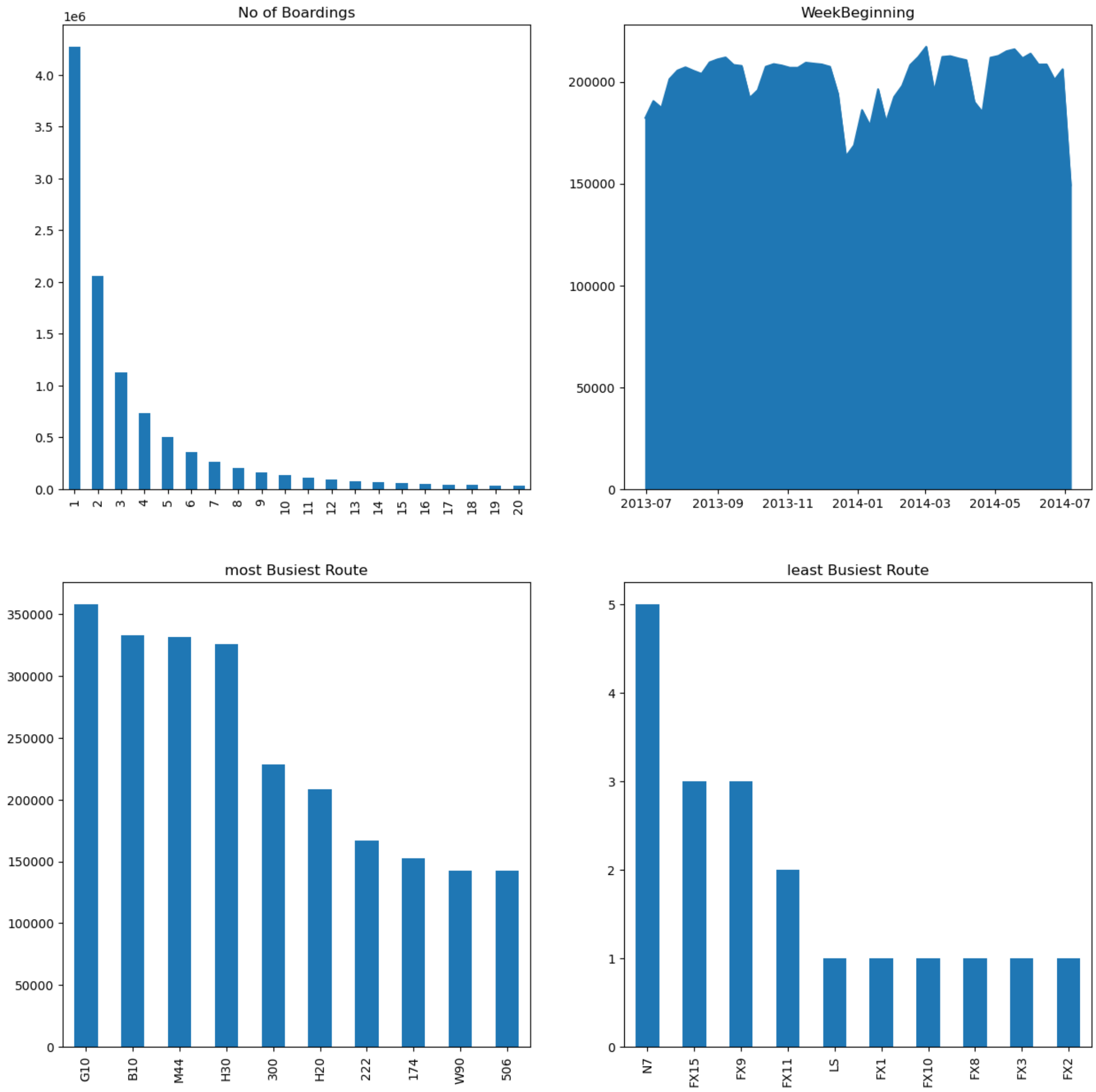


[22]:

```
import matplotlib.pyplot as plt
fig,axrr=plt.subplots(2,2,figsize=(15,15))

ax=axrr[0][0]
ax.set_title("No of Boardings")
data['NumberOfBoardings'].value_counts().sort_index().head(20).plot.
  ↪bar(ax=axrr[0][0])

ax=axrr[0][1]
ax.set_title("WeekBeginning")
data['WeekBeginning'].value_counts().plot.area(ax=axrr[0][1])

ax=axrr[1][0]
ax.set_title("most Busiest Route")
data['RouteID'].value_counts().head(10).plot.bar(ax=axrr[1][0])

ax=axrr[1][1]
ax.set_title("least Busiest Route")
data['RouteID'].value_counts().tail(10).plot.bar(ax=axrr[1][1])
```

[22]:

```
<Axes: title={'center': 'least Busiest Route'}>
```



[23]:
```
from math import sin, cos, sqrt, atan2, radians
def calc_dist(lat1,lon1):
## approximate radius of earth in km
    R = 6373.0
    dlon = radians(138.604801) - radians(lon1)
    dlat = radians(-34.921247) - radians(lat1)
    a = sin(dlat / 2)**2 + cos(radians(lat1)) * cos(radians(-34.921247))␣
 ↪*sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))
    return R * c
```

[25]:
```
out_geo = pd.read_csv('C:/Users/abuba/Downloads/output_geo.csv')
out_geo.shape
out_geo.head()
```

[25]:

```
           accuracy                                  formatted_address  \
0           ROOFTOP   181 Cross Rd, Westbourne Park SA 5041, Australia
1           ROOFTOP   177 Cross Rd, Westbourne Park SA 5041, Australia
2           ROOFTOP   175 Cross Rd, Westbourne Park SA 5041, Australia
3  GEOMETRIC_CENTER  Zone A Arndale Interchange - South side, Kilke…
4           ROOFTOP           178 Cross Rd, Malvern SA 5061, Australia

               google_place_id                input_string   latitude  \
0  ChIJKT7I9rbPsGoRVHMHkIy-Oyk                181 Cross Rd -34.966656
1  ChIJ-VFZ87bPsGoRyfVgC5qbPpE                177 Cross Rd -34.966607
2  ChIJIztlirbPsGoR38KRk76kPFI                175 Cross Rd -34.966758
3  ChIJn0C1hCPGsGoRIWvCdhF1RIg  Zone A Arndale Interchange -34.875160
4  ChIJycNiylv0sGoRdhfq9GKnpq0                178 Cross Rd -34.964960

    longitude  number_of_results postcode status  \
0  138.592148                  1     5041     OK
1  138.592301                  1     5041     OK
2  138.592715                  1     5041     OK
3  138.551628                  1     5009     OK
4  138.611477                  1     5061     OK

                                                type
0                                     street_address
1                                     street_address
2                                     street_address
3  bus_station,establishment,point_of_interest,tr…
4                                     street_address
```

[26]:

```
out_geo['dist_from_centre'] = out_geo[['latitude','longitude']].apply(lambda x:
  ↪calc_dist(*x), axis=1)
out_geo.head()
```

[26]:

```
           accuracy                                  formatted_address  \
0           ROOFTOP   181 Cross Rd, Westbourne Park SA 5041, Australia
1           ROOFTOP   177 Cross Rd, Westbourne Park SA 5041, Australia
2           ROOFTOP   175 Cross Rd, Westbourne Park SA 5041, Australia
3  GEOMETRIC_CENTER  Zone A Arndale Interchange - South side, Kilke…
4           ROOFTOP           178 Cross Rd, Malvern SA 5061, Australia

               google_place_id                input_string   latitude  \
0  ChIJKT7I9rbPsGoRVHMHkIy-Oyk                181 Cross Rd -34.966656
1  ChIJ-VFZ87bPsGoRyfVgC5qbPpE                177 Cross Rd -34.966607
2  ChIJIztlirbPsGoR38KRk76kPFI                175 Cross Rd -34.966758
3  ChIJn0C1hCPGsGoRIWvCdhF1RIg  Zone A Arndale Interchange -34.875160
4  ChIJycNiylv0sGoRdhfq9GKnpq0                178 Cross Rd -34.964960
```

```
    longitude  number_of_results postcode status  \
0  138.592148                  1     5041     OK   
1  138.592301                  1     5041     OK   
2  138.592715                  1     5041     OK   
3  138.551628                  1     5009     OK   
4  138.611477                  1     5061     OK   

                                                type  dist_from_centre  
0                                     street_address          5.180961  
1                                     street_address          5.172525  
2                                     street_address          5.180709  
3  bus_station,establishment,point_of_interest,tr…          7.057549  
4                                     street_address          4.900099  
```

[27]:
```
out_geo['type'].fillna('street_address',inplace=True)
out_geo['type'] = out_geo['type'].apply(lambda x: str(x).split(',')[-1])
```

[28]:
```
out_geo['type'].unique()
```

[28]:
```
array(['street_address', 'transit_station', 'premise', 'political',
       'school', 'route', 'intersection', 'point_of_interest',
       'subpremise', 'real_estate_agency', 'university', 'travel_agency',
       'restaurant', 'supermarket', 'store', 'post_office'], dtype=object)
```

[29]:
```
data= pd.merge(data,out_geo,how='left',left_on = 'StopName',right_on␣
 ↪='input_string')
data.head(5)
data.shape
```

[29]:
```
(10857234, 17)
```

[30]:
```
col = ['TripID', 'RouteID', 'StopID',␣
 ↪'StopName','WeekBeginning','NumberOfBoardings',
       'latitude', 'longitude','postcode','type','dist_from_centre']
data = data[col]
```

[31]:
```
new_data = data[data['StopName'].isin(stopListName)]
new_data.shape
print("data without stopage removing: ", data.shape)
print("data, after removing stoppage not having the data of whole 54 weeks:␣
 ↪",new_data.shape)
```

```
data without stopage removing:  (10857234, 11)
data, after removing stoppage not having the data of whole 54 weeks:  (10567931,
11)
```

```
[32]: new_data.head(2)
```

```
[32]:    TripID RouteID  StopID      StopName WeekBeginning  NumberOfBoardings  \
      0   23631     100   14156  181 Cross Rd    2013-06-30                  1
      1   23631     100   14144  177 Cross Rd    2013-06-30                  1

          latitude   longitude postcode            type  dist_from_centre
      0 -34.966656  138.592148     5041  street_address          5.180961
      1 -34.966607  138.592301     5041  street_address          5.172525
```

```
[33]: grouped = data.groupby(['StopName','WeekBeginning','type'])
      grouped = data.groupby(['StopName','WeekBeginning','type']).
       ↪agg({'NumberOfBoardings': ['sum', 'count','max']})
      grouped.columns = ["_".join(x) for x in grouped.columns.ravel()]
```

```
C:\Users\abuba\AppData\Local\Temp\ipykernel_5396\28787847.py:3: FutureWarning:
Index.ravel returning ndarray is deprecated; in a future version this will
return a view on self.
  grouped.columns = ["_".join(x) for x in grouped.columns.ravel()]
```

```
[34]: grouped.head(10)
```

```
[34]:                                         NumberOfBoardings_sum  \
      StopName    WeekBeginning type
      1 Anzac Hwy 2013-06-30    street_address                  1003
                  2013-07-07    street_address                   783
                  2013-07-14    street_address                   843
                  2013-07-21    street_address                   710
                  2013-07-28    street_address                   898
                  2013-08-04    street_address                   799
                  2013-08-11    street_address                  1012
                  2013-08-18    street_address                   793
                  2013-08-25    street_address                   897
                  2013-09-01    street_address                  1368

                                              NumberOfBoardings_count  \
      StopName    WeekBeginning type
      1 Anzac Hwy 2013-06-30    street_address                      378
                  2013-07-07    street_address                      360
                  2013-07-14    street_address                      343
                  2013-07-21    street_address                      356
                  2013-07-28    street_address                      379
                  2013-08-04    street_address                      378
                  2013-08-11    street_address                      358
                  2013-08-18    street_address                      333
                  2013-08-25    street_address                      354
                  2013-09-01    street_address                      431
```

```
                                          NumberOfBoardings_max
StopName    WeekBeginning type
1 Anzac Hwy 2013-06-30    street_address                     51
            2013-07-07    street_address                     28
            2013-07-14    street_address                     45
            2013-07-21    street_address                     28
            2013-07-28    street_address                     41
            2013-08-04    street_address                     40
            2013-08-11    street_address                     71
            2013-08-18    street_address                     41
            2013-08-25    street_address                     45
            2013-09-01    street_address                     59
```

[35]:

```
grouped.columns
```

[35]:

```
Index(['NumberOfBoardings_sum', 'NumberOfBoardings_count',
       'NumberOfBoardings_max'],
      dtype='object')
```

[36]:

```
new_data.head(2)
```

[36]:

```
   TripID RouteID  StopID      StopName WeekBeginning  NumberOfBoardings  \
0   23631     100   14156  181 Cross Rd    2013-06-30                  1
1   23631     100   14144  177 Cross Rd    2013-06-30                  1

    latitude   longitude postcode            type  dist_from_centre
0 -34.966656  138.592148     5041  street_address          5.180961
1 -34.966607  138.592301     5041  street_address          5.172525
```

[37]:

```
filtered_data = out_geo[out_geo['dist_from_centre'] <= 100]
filtered_data.shape
```

[37]:

```
(3960, 11)
```

[38]:

```
filtered_data.head(2)
```

[38]:

```
  accuracy                                 formatted_address  \
0  ROOFTOP  181 Cross Rd, Westbourne Park SA 5041, Australia
1  ROOFTOP  177 Cross Rd, Westbourne Park SA 5041, Australia

               google_place_id  input_string   latitude   longitude  \
0  ChIJKT7I9rbPsGoRVHMHkIy-Oyk  181 Cross Rd -34.966656  138.592148
1  ChIJ-VFZ87bPsGoRyfVgC5qbPpE  177 Cross Rd -34.966607  138.592301

   number_of_results postcode status            type  dist_from_centre
0                  1     5041     OK  street_address          5.180961
```

```
1                   1     5041     OK  street_address          5.172525
```

[39]:
```
bb_grp = data.groupby(['dist_from_centre']).agg({'NumberOfBoardings': ['sum']}).
 ↪reset_index()
bb_grp.columns = bb_grp.columns.get_level_values(0)
bb_grp.head()
bb_grp.columns
bb_grp.tail()
```

[39]:
```
      dist_from_centre  NumberOfBoardings
3028      16642.619167                 57
3029      16643.170375                 60
3030      16643.501617                203
3031      16790.222352                 31
3032      17237.137089               1992
```

[40]:
```
import plotly.graph_objs as go
from plotly.offline import iplot
trace0 = go.Scatter(x = bb_grp['dist_from_centre'],
                    y = bb_grp['NumberOfBoardings'],mode = 'lines+markers',name␣
 ↪= 'X2 KingWilliam 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)
```

[41]:
```
x = data["dist_from_centre"]
distance_10 = []
distance_10_50 = []
distance_50_100 = []
#distance_100_ = []
distance_100_more = []
total = 0
outlier = []
outlier_ = 0
for i in x:
    if(i<=10):
        distance_10.append(i)
        total += 1
    elif(i<=50):
        distance_10_50.append(i)
        total += 1
    elif(i<=100):
        distance_50_100.append(i)
        total += 1
```

```
print(outlier_)
```

```
0
```

[42]:
```
y = len(distance_10)+len(distance_10_50)+len(distance_50_100)
```

[43]:
```
print(total)
print("passangers, boarding the buses in the radious of 10Km from the city␣
  ↪center = ", (len(distance_10)/total)*100)
print("passanger, boarding the buses from the distance of 10Km to 50Km from the␣
  ↪city center = ", (len(distance_10_50)/total)*100)
print("passanger, boarding the buses from the distance of 50Km to 100 from the␣
  ↪city center = ", (len(distance_50_100)/total)*100)
```

```
10601778
passangers, boarding the buses in the radious of 10Km from the city center =
63.955555379484466
passanger, boarding the buses from the distance of 10Km to 50Km from the city
center =  33.57616995941624
passanger, boarding the buses from the distance of 50Km to 100 from the city
center =  2.468274661099299
```

[44]:
```
grouped_route = data.groupby(['RouteID']).agg({'NumberOfBoardings': ['sum',␣
  ↪'max']})
grouped_route.columns = ["_".join(x) for x in grouped_route.columns.ravel()]
```

```
C:\Users\abuba\AppData\Local\Temp\ipykernel_5396\1308629243.py:2: FutureWarning:

Index.ravel returning ndarray is deprecated; in a future version this will
return a view on self.
```

[45]:
```
grouped_route = grouped_route.head().reset_index()
type(grouped_route)
```

[45]:
```
pandas.core.frame.DataFrame
```

[46]:
```
grouped_route = grouped_route.sort_values("NumberOfBoardings_sum", ascending =␣
  ↪True)
grouped_route.shape
```

[46]:
```
(5, 3)
```

[47]:
```
route_data = grouped_route[grouped_route['RouteID'] == ""]
route_data.head()
```

```
[47]: Empty DataFrame
      Columns: [RouteID, NumberOfBoardings_sum, NumberOfBoardings_max]
      Index: []
```

```
[ ]:
```

Conclusion

- Therefore, it Calculates the number of people commuting within a specific route.
- It shows the accuracy means of commute based upon market information.