Public Transport Efficiency Analysis

Objective:

In this notebook,We have explored how people are travelling from different stops in Adelaide Metropolitan area and managing the buses on

each route according to the no of passenger commuting through the buses.

%matplotlib inline

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import datetime

import os

from math import sqrt

import warnings

## For Multiple Output in single cell

from IPython.core.interactiveshell import InteractiveShell

InteractiveShell.ast\_node\_interactivity = "all"

warnings.filterwarnings('ignore')

data = pd.read\_csv('../input/unisys/ptsboardingsummary/20140711.CSV')

data.shape

data.head(10)



(10857234, 6)

**TripID RouteID StopID**

**StopName**

**WeekBeginning NumberOfBoardings**

**0**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

23631

23631

23632

23633

23633

23634

23634

23634

23634

23634

100

100

100

100

100

100

100

100

100

100

14156

14144

14132

181 Cross Rd 2013-06-30 00:00:00

177 Cross Rd 2013-06-30 00:00:00

175 Cross Rd 2013-06-30 00:00:00

1

1

1

2

1

1

1

1

1

1

12266 Zone A Arndale Interchange 2013-06-30 00:00:00

14147

13907

14132

13335

13875

13045

178 Cross Rd 2013-06-30 00:00:00

9A Marion Rd 2013-06-30 00:00:00

175 Cross Rd 2013-06-30 00:00:00

9A Holbrooks Rd 2013-06-30 00:00:00

9 Marion Rd 2013-06-30 00:00:00

206 Holbrooks Rd 2013-06-30 00:00:00

out\_geo = pd.read\_csv('../input/outgeo/output\_geo.csv')

out\_geo.shape

out\_geo.head()

(4165, 10)

**accuracy**

**formatted\_address**

**google\_place\_id**

ChIJKT7I9rbPsGoRVHMHkIy-Oyk

ChIJ-VFZ87bPsGoRyfVgC5qbPpE

ChIJIztlirbPsGoR38KRk76kPFI

**input\_string**

**latitud**

**0**

**1**

**2**

**3**

**4**

ROOFTOP 181 Cross Rd, Westbourne Park SA 5041, Australia

ROOFTOP 177 Cross Rd, Westbourne Park SA 5041, Australia

ROOFTOP 175 Cross Rd, Westbourne Park SA 5041, Australia

181 Cross Rd -34.966656

177 Cross Rd -34.96660

175 Cross Rd -34.96675

GEOMETRIC\_CENTER

ROOFTOP

Zone A Arndale Interchange - South side, Kilke... ChIJn0C1hCPGsGoRIWvCdhF1RIg Zone A Arndale Interchange -34.87516

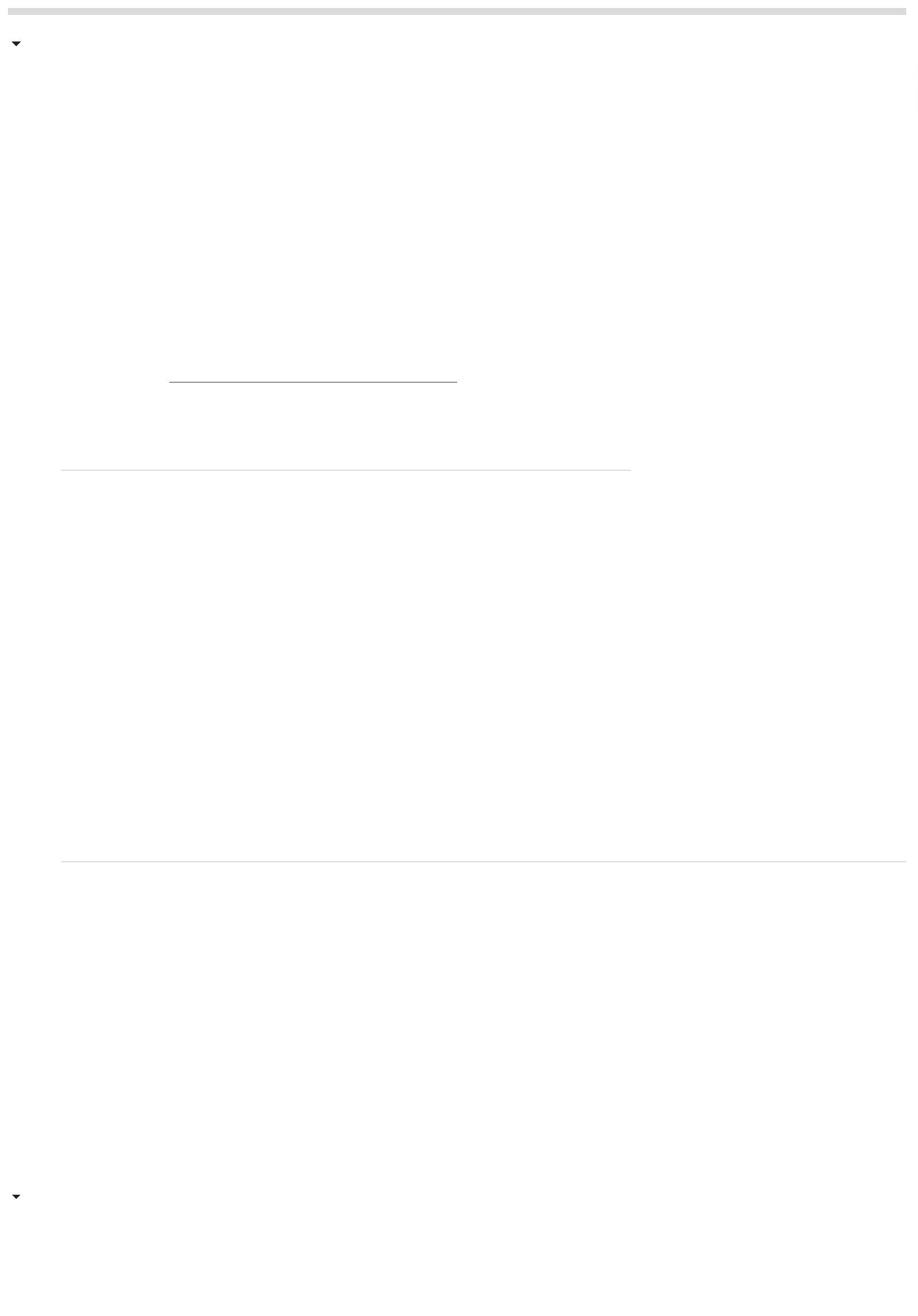
178 Cross Rd, Malvern SA 5061, Australia ChIJycNiylvOsGoRdhfq9GKnpq0 178 Cross Rd -34.96496

External Features

#DistanceFromCentre: Distance measure from the city centre

#For Calculating Distance between centre with other bus stops by using Longitude and Latitude

#we have used the Haversine formula



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

out\_geo['dist\_from\_centre'] = out\_geo[['latitude','longitude']].apply(lambda x: calc\_dist(\*x), axis=1)

out\_geo.head()

**accuracy**

**formatted\_address**

**google\_place\_id**

**input\_string**

**latitude longitud**

181 Cross Rd, Westbourne Park SA 5041,

Australia

**0**

**1**

**2**

ROOFTOP

ChIJKT7I9rbPsGoRVHMHkIy-Oyk

181 Cross Rd -34.966656 138.59214

177 Cross Rd -34.966607 138.59230

177 Cross Rd, Westbourne Park SA 5041,

Australia

ROOFTOP

ROOFTOP

ChIJ-VFZ87bPsGoRyfVgC5qbPpE

ChIJIztlirbPsGoR38KRk76kPFI

175 Cross Rd, Westbourne Park SA 5041,

Australia

175 Cross Rd -34.966758 138.59271

Zone A Arndale

Zone A Arndale Interchange - South side,

Kilke...

**3**

**4**

GEOMETRIC\_CENTER

ChIJn0C1hCPGsGoRIWvCdhF1RIg

ChIJycNiylvOsGoRdhfq9GKnpq0

-34.875160 138.55162

Interchange

ROOFTOP 178 Cross Rd, Malvern SA 5061, Australia

178 Cross Rd -34.964960 138.61147

#exp\_data = out\_geo.head(10)

##Fill the missing values with mode

out\_geo['type'].fillna('street\_address',inplace=True)

out\_geo['type'] = out\_geo['type'].apply(lambda x: str(x).split(',')[-1])

out\_geo['type'].unique()

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)

data['WeekBeginning'] = pd.to\_datetime(data['WeekBeginning']).dt.date

data['WeekBeginning'][1]

datetime.date(2013, 6, 30)

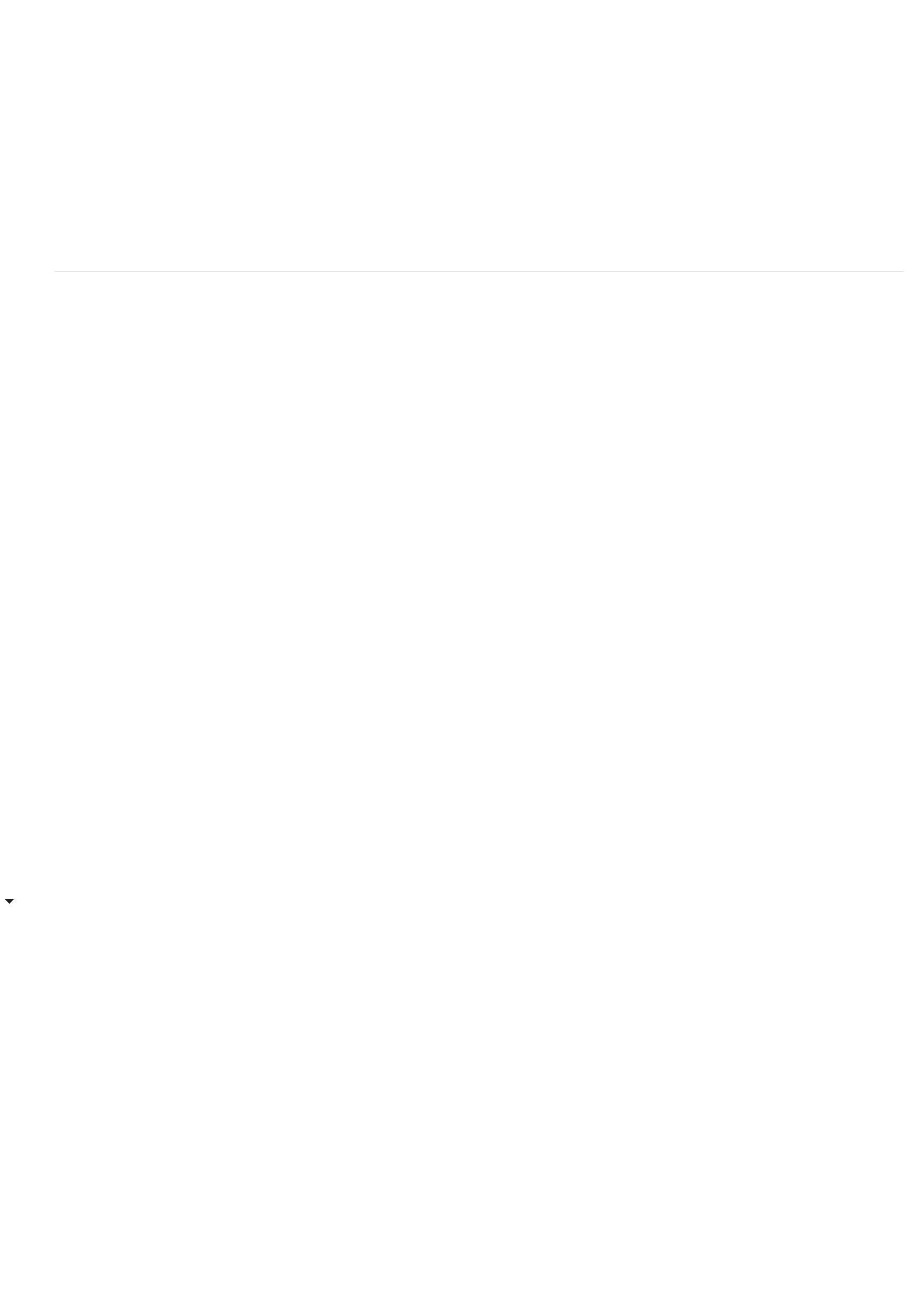
Data Aggregation

#Combine the Geolocation and main input file to get final Output File.

data= pd.merge(data,out\_geo,how='left',left\_on = 'StopName',right\_on = 'input\_string')

data.head(5)

data.shape



**TripID RouteID StopID**

**StopName WeekBeginning NumberOfBoardings**

**accuracy formatted\_address**

**googl**

181 Cross Rd,

ROOFTOP Westbourne Park SA

181 Cross

**0**

23631

100

14156

2013-06-30

1

ChIJKT7I9rbPsGoRV

Rd

#Columns to keep for further analysis

col = ['TripID', 'RouteID', 'StopID', 'StopName', 'WeekBeginning','NumberOfBoardings',

'latitude', 'longitude','postcode','type','dist\_from\_centre']

data = data[col]

y

175 Cross Rd,

Zone A Arndale

##saving the final dataset

#data.to\_csv('Weekly\_Boarding.csv',index=False)

3

Zone A

IW

Aggregate the Data According to Weeks and Stop names

**NumberOfBoardings\_sum** Number of Boardings within particular week for each Bus stop

**NumberOfBoardings\_count** Number of times data is recorded within week

d

**NumberOfBoardings\_max** Maximum number of boarding done at single time within week

grouped = data.groupby(['StopName','WeekBeginning','type'])

#grouped.head()

# st\_week\_grp1 = pd.DataFrame(data.groupby(['StopName','WeekBeginning','type']).agg({'NumberOfBoardings': ['sum', 'count']})).reset\_index

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

grouped.columns = ["\_".join(x) for x in grouped.columns.ravel()]

grouped.head(10)

grouped.columns

**NumberOfBoardings\_sum NumberOfBoardings\_count NumberOfBoardings\_max**

**StopName WeekBeginning**

**type**

**street\_address**

**street\_address**

**street\_address**

**street\_address**

**street\_address**

**street\_address**

**street\_address**

**street\_address**

**street\_address**

**street\_address**

**1 Anzac Hwy**

**2013-06-30**

**2013-07-07**

**2013-07-14**

**2013-07-21**

**2013-07-28**

**2013-08-04**

**2013-08-11**

**2013-08-18**

**2013-08-25**

**2013-09-01**

1003

783

843

710

898

799

1012

793

897

1368

378

360

343

356

379

378

358

333

354

431

51

28

45

28

41

40

71

41

45

59

Index(['NumberOfBoardings\_sum', 'NumberOfBoardings\_count',

'NumberOfBoardings\_max'],

dtype='object')

st\_week\_grp = pd.DataFrame(grouped).reset\_index()

st\_week\_grp.shape

st\_week\_grp.head()

(207864, 6)

**StopName WeekBeginning**

**type NumberOfBoardings\_sum NumberOfBoardings\_count NumberOfBoardings\_max**

**0**

**1**

**2**

**3**

**4**

1 Anzac Hwy

1 Anzac Hwy

1 Anzac Hwy

1 Anzac Hwy

1 Anzac Hwy

2013-06-30 street\_address

1003

783

843

710

898

378

360

343

356

379

51

28

45

28

41

2013-07-07 street\_address

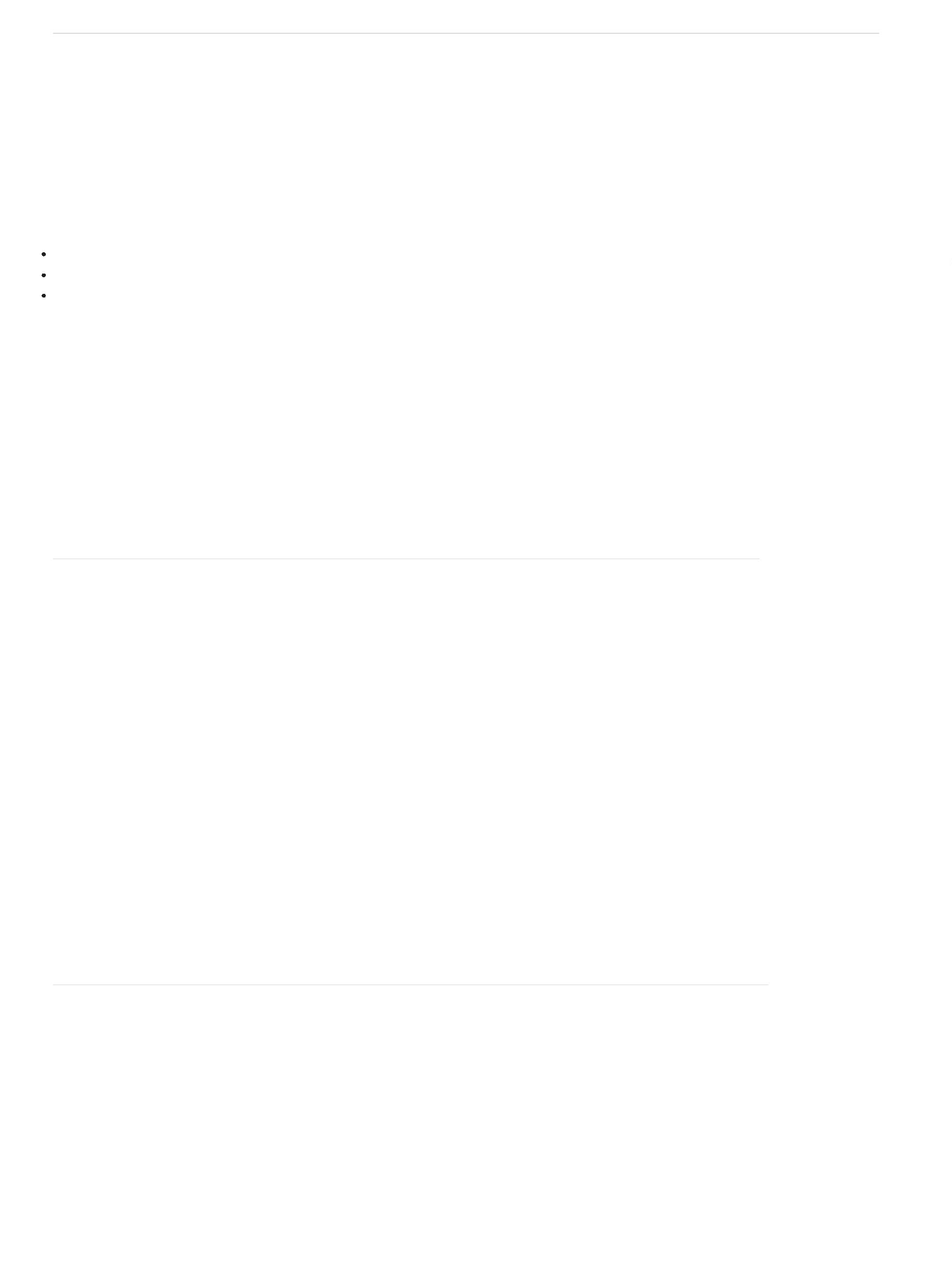
2013-07-14 street\_address

2013-07-21 street\_address

2013-07-28 street\_address

st\_week\_grp1 = pd.DataFrame(st\_week\_grp.groupby('StopName')["WeekBeginning"].count()).reset\_index()

st\_week\_grp1.head()



**StopName WeekBeginning**

**0**

**1**

1 Anzac Hwy

1 Bartels Rd

54

54

#Gathering only the Stop Name which having all 54 weeks of Dat

aa = list(st\_week\_grp1[st\_week\_grp1['WeekBeginning'] == 54]['StopName'])

aa[1:10]

['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']

bb = st\_week\_grp[st\_week\_grp['StopName'].isin(aa)]

bb.head()

bb.shape

type(bb)

**StopName WeekBeginning**

**type NumberOfBoardings\_sum NumberOfBoardings\_count NumberOfBoardings\_max**

**0**

**1**

**2**

**3**

**4**

1 Anzac Hwy

1 Anzac Hwy

1 Anzac Hwy

1 Anzac Hwy

1 Anzac Hwy

2013-06-30 street\_address

1003

783

843

710

898

378

360

343

356

379

51

28

45

28

41

2013-07-07 street\_address

2013-07-14 street\_address

2013-07-21 street\_address

2013-07-28 street\_address

(175446, 6)pandas.core.frame.DataFrame

#removing the stoppage which are not having the data of whole 54 weeks

new\_data = data[data['StopName'].isin(aa)]

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)

(10567931, 11)data without stopage removing: (10857234, 11)

data, after removing stoppage not having the data of whole 54 weeks: (10567931, 11)

new\_data.head(2)

filtered\_data = new\_data[new\_data['dist\_from\_centre'] <= 100]

filtered\_data.shape

**TripID RouteID StopID**

**StopName WeekBeginning NumberOfBoardings**

**latitude longitude postcode**

**type dist\_from\_c**

**0**

**1**

23631

100

100

14156 181 Cross Rd

14144 177 Cross Rd

2013-06-30

2013-06-30

1

1

-34.966656 138.592148

-34.966607 138.592301

5041 street\_address

5041 street\_address

5.1

5.1

23631

(10341468, 11)

data = filtered\_data.copy()

data.shape

(10341468, 11)

#No of boarding for each stopage in all weeks

#bb["StopName"].groupby(NumberOfBoardings\_sum)

stopageName\_with\_boarding = bb.groupby(['StopName']).agg({'NumberOfBoardings\_sum': ['sum']})

#stopageName\_with\_boarding.columns = ["\_".join(x) for x in stopageName\_with\_boarding.columns.ravel()]

#stopageName\_with\_boarding.head()

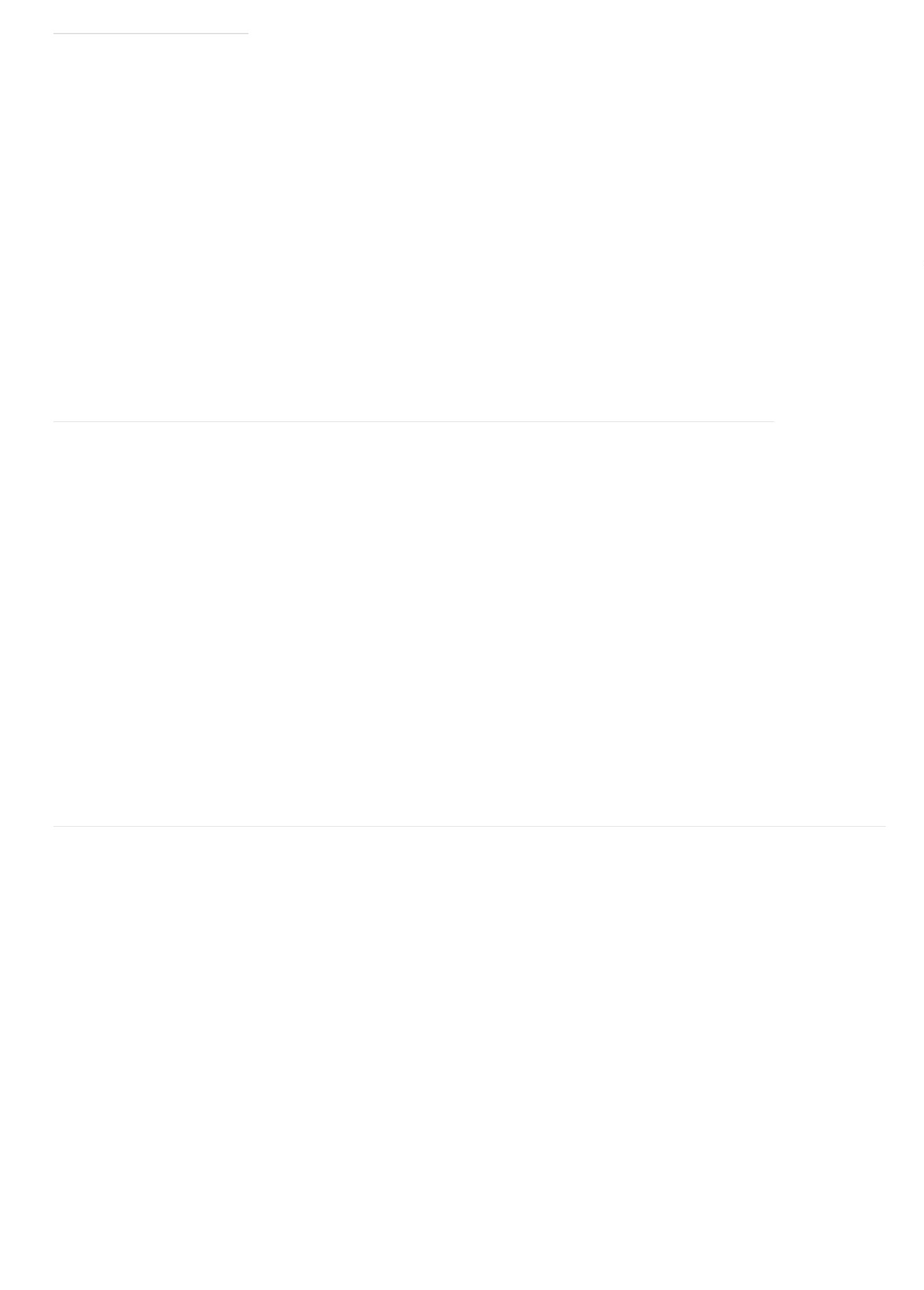
stopageName\_with\_boarding = pd.DataFrame(stopageName\_with\_boarding.reset\_index())

#type(stopageName\_with\_boarding)

stopageName\_with\_boarding.columns = ["StopName", "Total\_boarding\_on\_the\_stopage"]

#stopageName\_with\_boarding.shape

stopageName\_with\_boarding.head()



**StopName Total\_boarding\_on\_the\_stopage**

**0**

**1**

**2**

1 Anzac Hwy

1 Bartels Rd

1 Botanic Rd

39429

8412

14868

## save the aggregate data

#bb.to\_csv('st\_week\_grp.csv', index=False)

**4**

1 Fullarton Rd

585

Data Exploration

data.nunique()

#data.isnull().sum()

#data['WeekBeginning'].unique()

TripID

RouteID

39211

616

StopID

StopName

5838

3127

54

WeekBeginning

NumberOfBoardings

latitude

longitude

postcode

359

2393

2379

138

type

8

dist\_from\_centre

dtype: int64

2397

Data Visualization

##can assign the each chart to one axes at a time

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]

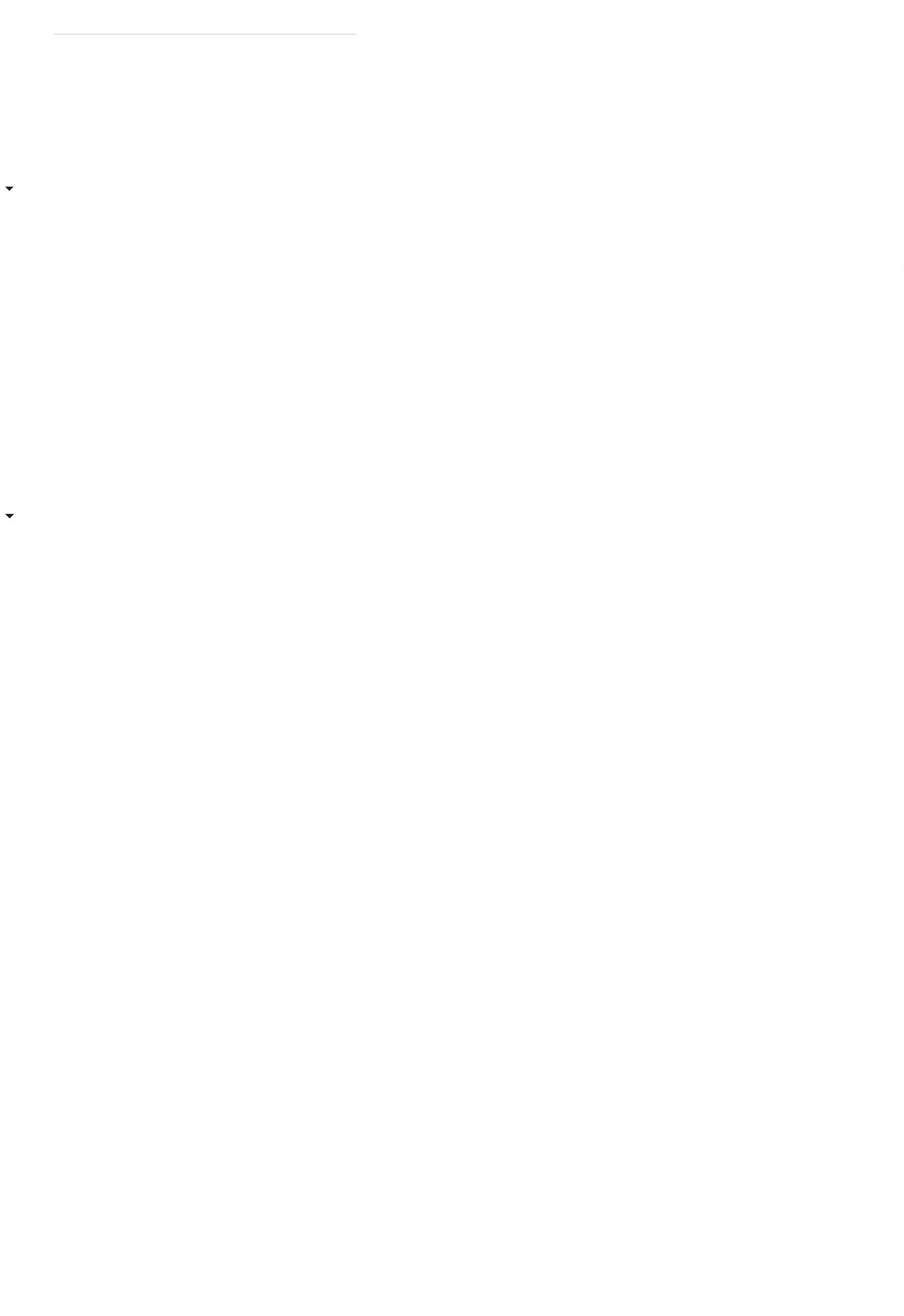
ax.set\_title("most Busiest Route")

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])



Text(0.5,1,'No of Boardings')<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff880af0940>Text(0.5,1,'WeekBeginning')<matplotlib.axes.\_s

<matplotlib.axes.\_subplots.AxesSubplot at 0x7ff709a48e10>Text(0.5,1,'least Busiest Route')<matplotlib.axes.\_subplots.AxesSubplot at

stopageName\_with\_boarding = stopageName\_with\_boarding.sort\_values('Total\_boarding\_on\_the\_stopage', ascending = False)

#stopage with most no of boarding

stopageName\_with\_boarding.head(10)

**StopName Total\_boarding\_on\_the\_stopage**

**3054**

**3125**

**3032**

**3130**

**3021**

**3207**

**3015**

I2 North Tce

X1 King William St

F2 Grenfell St

628859

622099

604149

583227

550396

547709

541046

451960

399351

356518

X2 King William St

E1 Currie St

Zone C Paradise Interchange

D1 King William St

**3211** Zone C Tea Tree Plaza Intercha

**3025**

**3039**

E3 Currie St

G3 Grenfell St

#stopage with least no of boarding

stopageName\_with\_boarding.tail(10)

**StopName Total\_boarding\_on\_the\_stopage**

**1845**

**2318**

**2732**

**58**

45 Mcintyre Rd

57 Philip Hwy

292

281

275

274

266

264

227

221

215

175

75B Frick St

109 Regency Rd

39D Glenloth Dr

127 Lyndoch Rd

**1633**

**170**

**3086** Strathalbyn South Tce

**1231**

**558**

31 Glenroy St

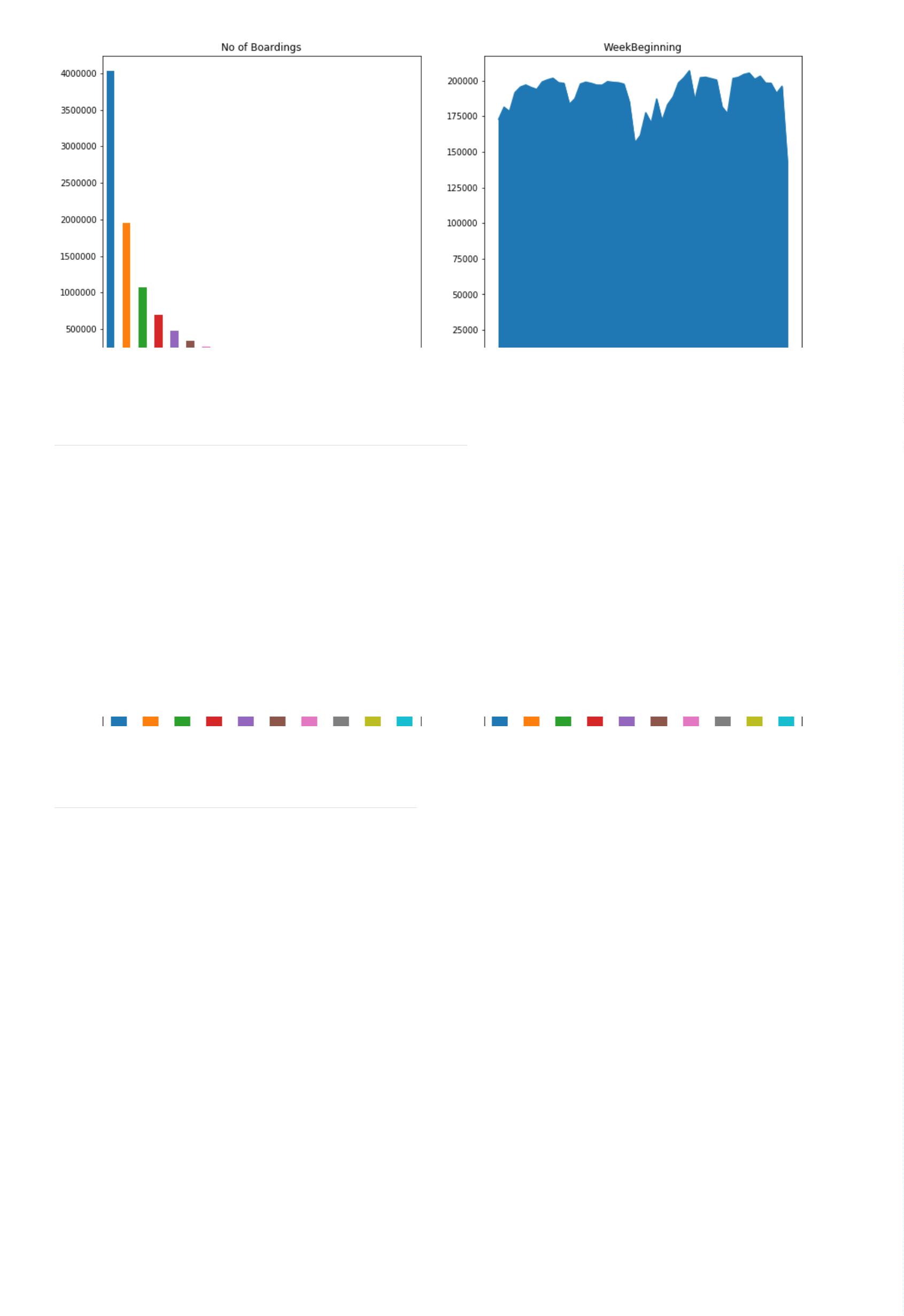
19 Gilles Rd

**294**

145 The Esplanade

ax = stopageName\_with\_boarding.head(10).plot.bar(x='StopName', y='Total\_boarding\_on\_the\_stopage', rot=90)

ax.set\_title("most busiest stopage")



Text(0.5,1,'most busiest stopage')

ax = stopageName\_with\_boarding.tail(10).plot.bar(x='StopName', y='Total\_boarding\_on\_the\_stopage', rot=90)

ax.set\_title("least busiest stopage")

Text(0.5,1,'least busiest stopage')

data['WeekBeginning'].value\_counts().mean()

191508.66666666666

# data['dist\_from\_centre'].nunique()

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()

**dist\_from\_centre NumberOfBoardings**

**0**

**1**

**2**

**3**

**4**

0.000018

0.131368

0.309089

0.314937

0.326005

1892435

167535

356518

1484824

120061

Index(['dist\_from\_centre', 'NumberOfBoardings'], dtype='object')

**dist\_from\_centre NumberOfBoardings**

**2392**

**2393**

**2394**

**2395**

**2396**

86.471064

94.826409

99.625655

99.665190

99.748995

18905

321

1101

4373

21216

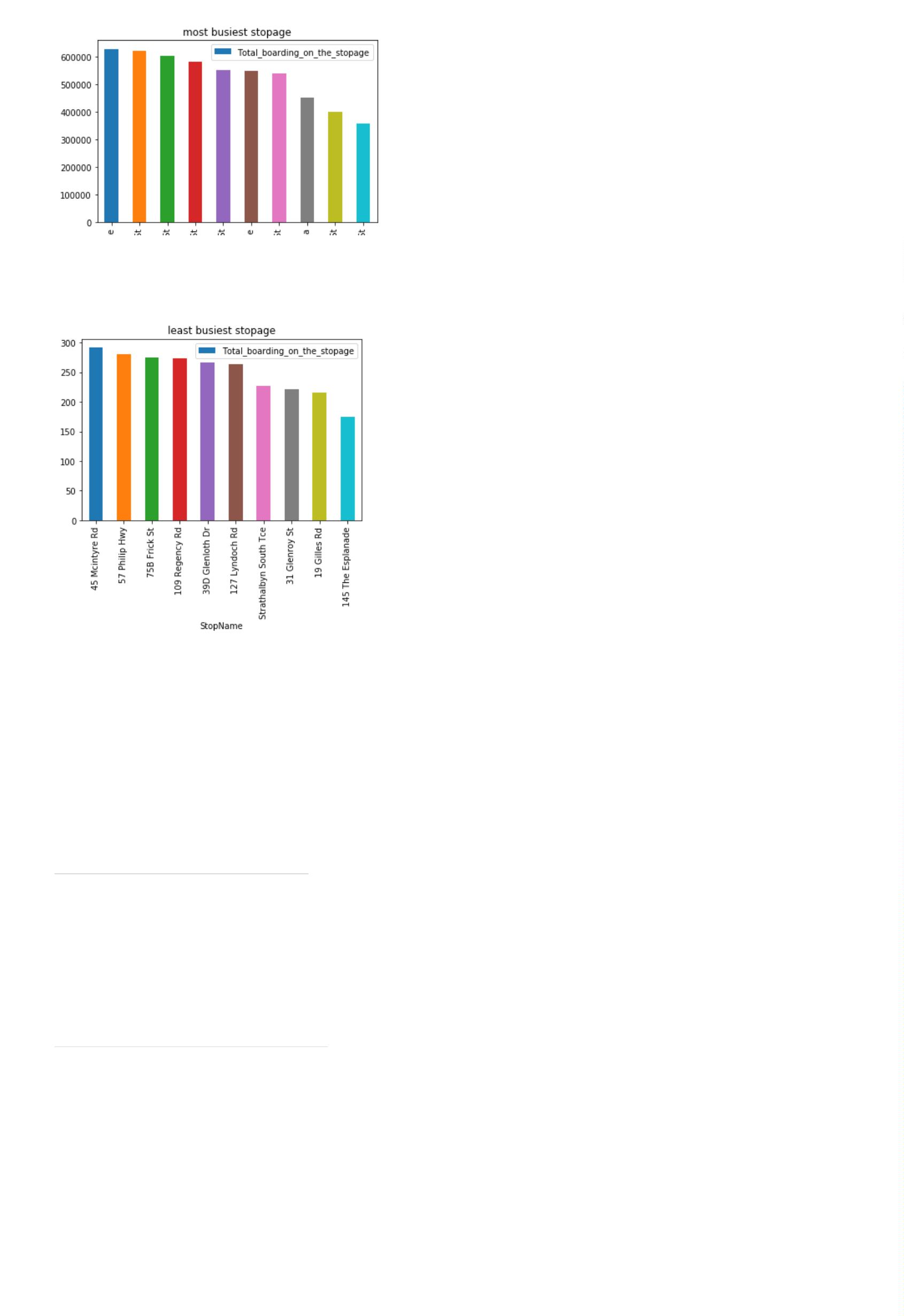
import plotly.graph\_objs as go

from plotly.offline import iplot

trace0 = go.Scatter(

x = bb\_grp['dist\_from\_centre'],

y = bb\_grp['NumberOfBoardings'],mode = 'lines+markers',name = 'X2 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)

#clustering Technique// based on the distance from city centre

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

#elif(i>100 and i< 2000):

#distance\_100\_more.append(i)

#total += 1

#elif(i>2000):

#outlier.append(i)

#outlier\_ += 1

print(outlier\_)

0

y = len(distance\_10)+len(distance\_10\_50)+len(distance\_50\_100)

#+len(distance\_100\_more)

#print(y)

#print(total)

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)

#print("passanger, boarding the buses from the distance of 100Km and more from the city center = ", (len(distance\_100\_more)/total)\*100)

10341468

passangers, boarding the buses in the radious of 10Km from the city center = 64.31275521038212

passanger, boarding the buses from the distance of 10Km to 50Km from the city center = 33.16731241638035

passanger, boarding the buses from the distance of 50Km to 100 from the city center = 2.5199323732375327

#busiest route on weekly basis

#data.head(10)

# st\_week\_grp1 = pd.DataFrame(data.groupby(['StopName','WeekBeginning','type']).agg({'NumberOfBoardings': ['sum', 'count']})).reset\_index

grouped\_route = data.groupby(['RouteID']).agg({'NumberOfBoardings': ['sum', 'max']})

grouped\_route.columns = ["\_".join(x) for x in grouped\_route.columns.ravel()]

"""grouped\_route = grouped\_route.head().reset\_index()

type(grouped\_route)

grouped\_route = grouped\_route.sort\_values("NumberOfBoardings\_sum", ascending = True)

#stopageName\_with\_boarding = stopageName\_with\_boarding.sort\_values('Total\_boarding\_on\_the\_stopage', ascending = False)

#stopage with most no of boarding

#stopageName\_with\_boarding.head(10)

#grouped\_route["NumberOfBoardings\_sum"] = grouped\_route["NumberOfBoardings\_sum"] / 365

grouped\_route.head(10)

grouped\_route.shape"""

'grouped\_route = grouped\_route.head().reset\_index()\ntype(grouped\_route)\ngrouped\_route =

grouped\_route.sort\_values("NumberOfBoardings\_sum", ascending = True)\n#stopageName\_with\_boarding =

stopageName\_with\_boarding.sort\_values(\'Total\_boarding\_on\_the\_stopage\', ascending = False)\n#stopage with most no of



boarding\n#stopageName\_with\_boarding.head(10)\n#grouped\_route["NumberOfBoardings\_sum"] = grouped\_route["NumberOfBoardings\_sum"] /

365\ngrouped\_route.head(10)\ngrouped\_route.shape'

....

"""route\_data = grouped\_route[grouped\_route['RouteID'] == "G10"]

route\_data.head()"""

'route\_data = grouped\_route[grouped\_route[\'RouteID\'] == "G10"]\nroute\_data.head()'

