PUBLIC TRANSPOTATION AND ANALYSIS

PHASE 4:DEVELOPMENT PART 2

About dataset:

Content:

The public bus transportation boarding summary.csv file contains route,trip,stop and week of year from 20140711.

Data source

The data fields in the given file are

- **TripID** Unique identity of trip
- **RouteID** Value representing public transport route
- **StopID** Unique identity of stop
- **StopName** Name of given stop
- **WeekBeginning** Date representing first day of any week
- **NumberOfBoarding** Count of all boarding's occurred at this stop for the named trip over the previous week

External Features

Some Important external data fields calculation

- **IsHoliday** Number of public holidays within that week
- **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

In [8]:

```
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
```

In [9]:

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

In [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])
```

In [11]:

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

Out[11]:

```
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)
```

Adding the details regarding the Public holidays from June 2013 to June 2014

In [12]:

```
'''Holidays--
```

Out[12]:

```
"Holidays--\n2013-09-01,Father's Day\n2013-10-07,Labour day\n2013-12-25,Christmas day\n2013-12-26,Proclamation Day\n2014-01-01,New Year\n2014-01-27,Australia Day\n2014-03-10,March Public Holiday\n2014-04-18,Good Friday\n2014-04-19,Easter Saturday\n2014-04-21,Easter Monday\n2014-04-25,Anzac Day\n2014-06-09,Queen's Birthday"
```

In [13]:

```
def holiday_label (row):
    if row == datetime.date(2013, 9, 1) :
          return '1'
    if row == datetime.date(2013, 10, 6) :
          return '1'
    if row == datetime.date(2013, 12, 22) :
          return '2'
    if row == datetime.date(2013, 12, 29):
          return '1'
    if row  == datetime.date(2014, 1, 26):
          return '1'
    if row == datetime.date(2014, 3, 9):
          return '1'
    if row == datetime.date(2014, 4, 13) :
          return '2'
    if row == datetime.date(2014, 4, 20):
        return '2'
    if row == datetime.date(2014, 6, 8):
        return '1'
    return '0'
```

In [14]:

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

In [15]:

```
data['holiday_label'] = data['WeekBeginning'].apply (lambda row: holiday_label(row))
```

Data Aggregation

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

In [16]:

```
data= pd.merge(data,out_geo,how='left',left_on = 'StopName',right_on =
'input_string')
```

In [17]:

```
data = pd.merge(data, route, how='left', left_on = 'RouteID', right_on = 'route_id')
```

Columns to keep for further analysis

In [18]:

```
col = ['TripID', 'RouteID', 'StopID', 'StopName',
'WeekBeginning','NumberOfBoardings','formatted_address',
      'latitude',
'longitude','postcode','type','route_desc','dist_from_centre','holiday_label']
```

In [19]:

```
data = data[col]
```

In [20]:

```
##saving the final dataset
data.to_csv('Weekly_Boarding.csv',index=False)
```

In [21]:

```
## getting the addresses for geolocation api.
# Address data['StopName'].unique()
# sub = pd.DataFrame({'Address': Address})
# sub=sub.reindex(columns=["Address"])
# sub.to_csv('addr.csv')
```

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

- **NumberOfBoardings_max** Maximum number of boarding done at single time within week

In [22]:

```
# 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()]
```

In [23]:

```
st_week_grp = pd.DataFrame(grouped).reset_index()
st_week_grp.shape
st_week_grp.head()
```

Out[23]:

```
(207864, 6)
```

Out[23]:

	StopName	WeekBeginning	type	NumberOfBoardings_sum	NumberOfBoardings_count	NumberOfBoardings_max
0	1 Anzac Hwy	2013-06-30	street_address	1003	378	51
1	1 Anzac Hwy	2013-07-07	street_address	783	360	28
2	1 Anzac Hwy	2013-07-14	street_address	843	343	45
3	1 Anzac Hwy	2013-07-21	street_address	710	356	28

	Stop Name	WeekBeginning	type	NumberOfBoardings_sum	NumberOfBoardings_count	NumberOfBoardings_max
4	1 Anzac Hwy	2013-07-28	street_address	898	379	41

Gathering only the Stop Name which having all 54 weeks of Data

In [24]:

```
st_week_grp1 = pd.DataFrame(st_week_grp.groupby('StopName')['WeekBeginning'].count()).reset_index()
```

In [25]:

```
aa=list(st_week_grp1[st_week_grp1['WeekBeginning'] == 54]['StopName'])
```

In [26]:

```
bb = st_week_grp[st_week_grp['StopName'].isin(aa)]
```

In [27]:

```
## save the aggregate data
bb.to_csv('st_week_grp.csv', index=False)
```

Data Exploration

Having Total of 4165 Stops in South Australian Metropolitan Area.

In [28]:

```
data.nunique()
```

Out[28]:

```
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
```

In [29]:

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

Out[29]:

```
(10857234, 14)
```

Out[29]:

```
Index(['TripID', 'RouteID', 'StopID', 'StopName', 'WeekBeginning',
       'NumberOfBoardings', 'formatted_address', 'latitude', 'longitude',
       'postcode', 'type', 'route_desc', 'dist_from_centre', 'holiday_label'],
      dtype='object')
```

Out[29]:

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings	formatted_address	latitude	longitude	postcode	type	route_desc	dist_from_centre	holiday_label
0	23631	100	14156	181 Cross Rd	2013-06-30	1	181 Cross Rd, Westbourne Park SA 5041, Australia	-34.966656	138.592148	5041	street_address	via Woodville Road, Holbrooks Road, Marion Roa...	5.180961	0

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings	formatted_address	latitude	longitude	postcode	type	route_desc	dist_from_centre	holiday_label
1	23631	100	14144	177 Cross Rd	2013-06-30	1	177 Cross Rd, Westbourne Park SA 5041, Australia	-34.966607	138.592301	5041	street_address	via Woodville Road, Holbrooks Road, Marion Roa...	5.172525	0
2	23632	100	14132	175 Cross Rd	2013-06-30	1	175 Cross Rd, Westbourne Park SA 5041, Australia	-34.966758	138.592715	5041	street_address	via Woodville Road, Holbrooks Road, Ma	5.180709	0

	TripID	RouteID	StopID	StopName	WeekBeginning	NumberOfBoardings	formatted_address	latitude	longitude	postcode	type	route_desc	dist_from_centre	holiday_label
												rion Roa...		

In [30]:

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

Out[30]:

```
TripID                    0
RouteID                   0
StopID                    0
StopName                  0
WeekBeginning             0
NumberOfBoardings         0
formatted_address      3506
latitude                  0
longitude                 0
postcode             425081
type                      0
route_desc          2106618
dist_from_centre          0
holiday_label             0
dtype: int64
```