



# Phase -3

OBJECTIVE:

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

# TOOLS:

The Public Transit toolbox contains tools for converting, displaying, editing, and analyzing public transit data. Several tools convert between General Transit Feed Specification (GTFS) datasets and feature classes and tables. Other tools perform analysi using public transit schedule data.

**DATA SET:**

TripID RouteID StopID StopName WeekBeginning

NumberOfBoardings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 23631 | 100 | 14156 | 181 Cross Rd 2013-06-30 00:00:00 | |
|  | 1 |  |  |  | |
| 1 | 23631 | 100 | 14144 | 177 Cross Rd 2013-06-30 00:00:00 | |
|  | 1 |  |  |  | |
| 2 | 23632 | 100 | 14132 | 175 Cross Rd 2013-06-30 00:00:00 | |
|  | 1 |  |  |  | |
| 3 | 23633 | 100 | 12266 | Zone A Arndale Interchange 2013- | |
| 06-30 00:00:00 | | 2 | | | |
| 4 | 23633 | 100 | 14147 | 178 Cross Rd | 2013-06-30 00:00:00 |
|  | 1 |  |  |  |  |
| 5 | 23634 | 100 | 13907 | 9A Marion Rd | 2013-06-30 00:00:00 |
|  | 1 |  |  |  |  |
| 6 | 23634 | 100 | 14132 | 175 Cross Rd | 2013-06-30 00:00:00 |
|  | 1 |  |  |  |  |
| 7 | 23634 | 100 | 13335 | 9A Holbrooks Rd | 2013-06-30 00:00:00 |
|  | 1 |  |  |  |  |
| 8 | 23634 | 100 | 13875 | 9 Marion Rd | 2013-06-30 00:00:00 |
|  | 1 |  |  |  |  |
| 9 | 23634 | 100 | 13045 | 206 Holbrooks Rd2013-06-30 00:00:00 | |

1

out\_geo = pd.rea

Program:

%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

Features:

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

**INPUT**:

out\_geo['dist\_from\_centre'] = out\_geo[['latitude','longitude']].apply(lambda x:

calc\_dist(\*x), axis=1) out\_geo.head() accuracy form