Analysis of MRT trips

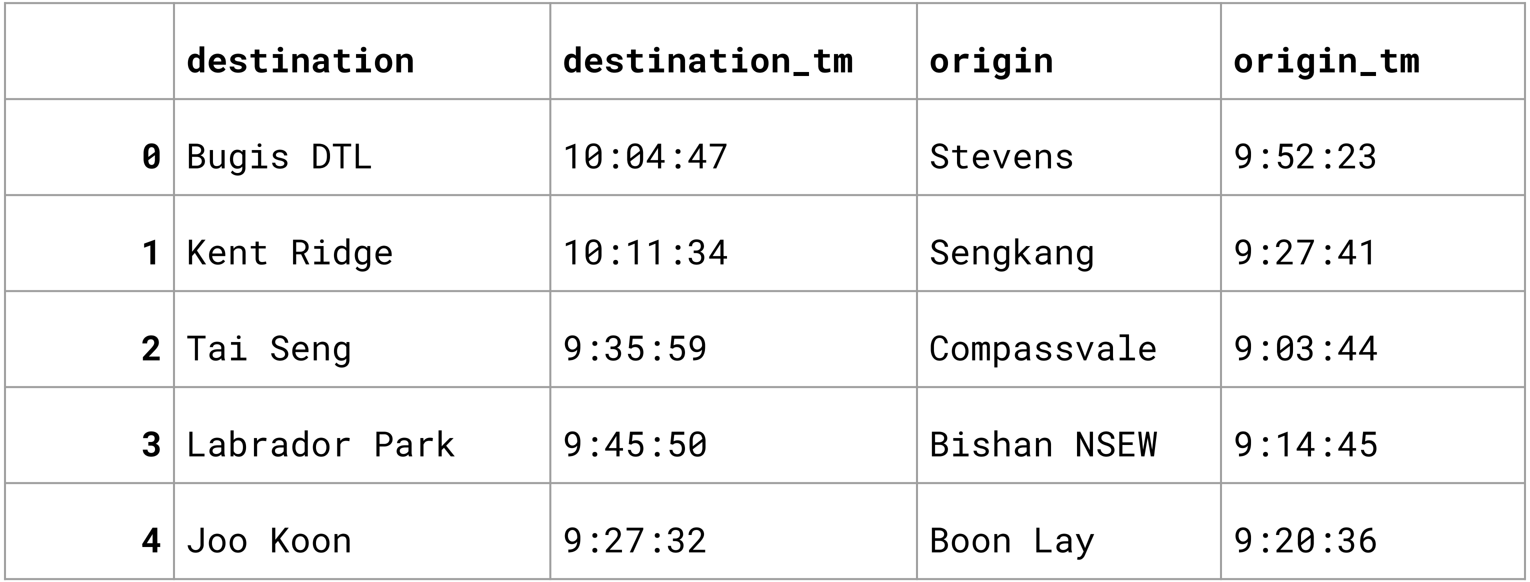
Hu Lingyi

1

Task 1

Derive the underlying MRT network from the dataset

Cleaning and extracting useful information from data



	destination	destination_tm	origin	origin_tm
0	Bugis DTL	10:04:47	Stevens	9:52:23
1	Kent Ridge	10:11:34	Sengkang	9:27:41
2	Tai Seng	9:35:59	Compassvale	9:03:44
3	Labrador Park	9:45:50	Bishan NSEW	9:14:45
4	Joo Koon	9:27:32	Boon Lay	9:20:36

Observations about the data before cleaning

- There are 381249 records
- There are 154 unique stations included,which reduced to 138 after cleaning
- All the trips started between 9 to 10 (peak hour journeys)
- Most frequent destination is Raffles Place and most frequent origin is Ang Mo Kio

Cleaning up the data

- There are trips that begin and end at the same station – remove them.
- Remove outliers (i.e. journeys below 60s or above 2 hours)
- There are stations that are repeated but called different names (e.g. Bugis DTL and Bugis NSEW) – they are taken to be the same station
- Journeys from A to B are equivalent to B to A

Extracting useful information

- Calculate duration of the trip from start and end times
- Subtract 1 minute from each duration to take into account waiting and walking times
- Tally up all the trips that go from a specific A to B, and take the median time
- Journeys with fewer than 4 trips are discarded as they may not be reliable.

Outline of the approach taken

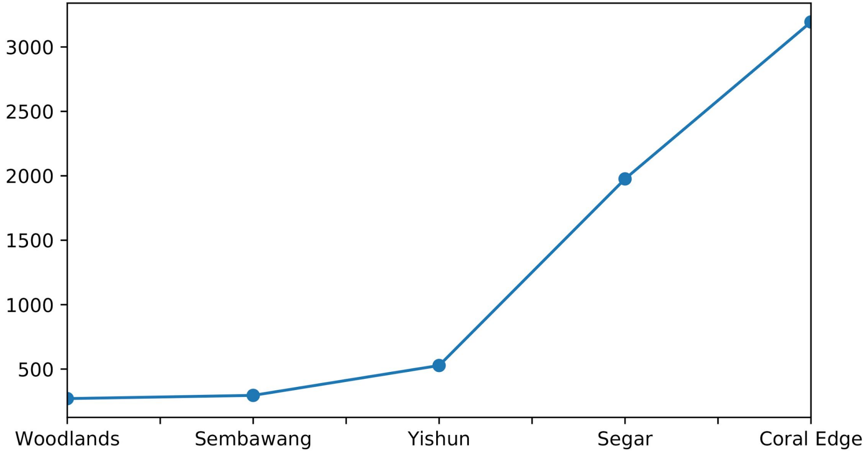
Assume every pair of stations `(X,Y)` with a trip between them is adjacent.

Only mark two stations as not adjacent if there is a station Z such that

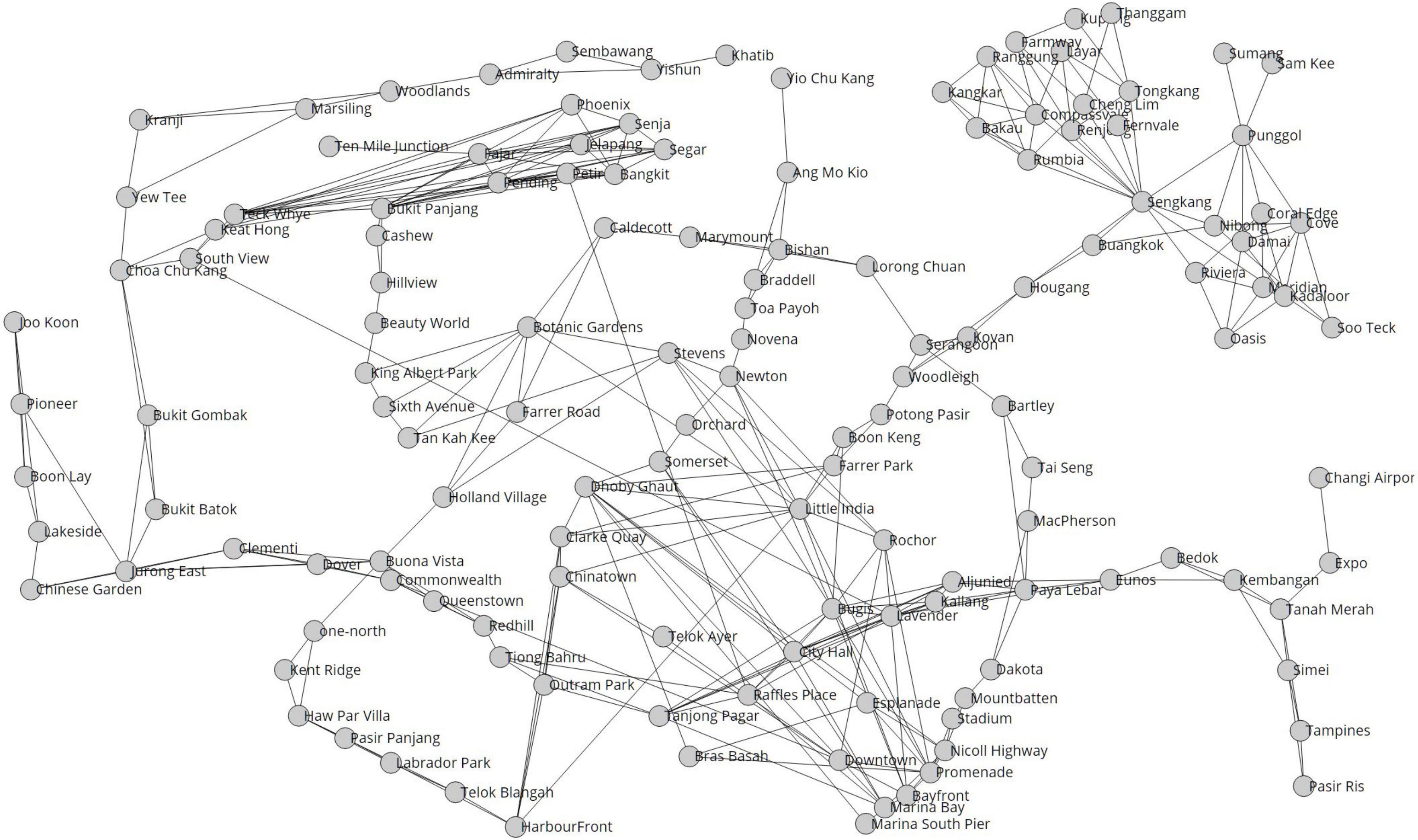
```
time(X to Z) + time(Z to Y) < time(X to Y)
```

Cleaning up the network

The previous approach results in many adjacent stations. For example, here is a plot of adjacent stations of Admiralty and their various trip durations from Aljunied:



Next step: Detect big jump in travel time between adjacent stations and remove them (e.g. Segar and Coral Edge.


Kupang
Thanggam
Farmway
Layar
Ranggung
Sumang
Sam Kee
Sembawang
Khatib
Yishun
Yio Chu Kang
Admiralty
Woodlands
Kangkar
Tongkang
Cheng Lim
Compassvale
Bakau
Renjong
Fernvale
Punggol
Phoenix
Senja
Kranji
Marsiling
Ten Mile Junction
Jelapang
Segar
Fajar
Petir
Bangkit
Pending
Rumbia
Ang Mo Kio
Yew Tee
Bukit Panjang
Teck Whye
Keat Hong
Sengkang
Coral Edge
Cove
Nibong
Caldecott
Marymount
Cashew
Bishan
Damai
Buangkok
South View
Choa Chu Kang
Lorong Chuan
Riviera
Hillview
Braddell
Hougang
Meridian
Kadaloor
Toa Payoh
Joo Koon
Beauty World
Botanic Gardens
Soo Teck
Novena
Kovan
Serangoon
Oasis
Stevens
King Albert Park
Newton
Woodleigh
Pioneer
Sixth Avenue
Bartley
Farrer Road
Potong Pasir
Bukit Gombak
Orchard
Tan Kah Kee
Boon Keng
Somerset
Farrer Park
Tai Seng
Boon Lay
Changi Airpor
Dhoby Ghaut
Holland Village
Bukit Batok
Little India
MacPherson
Lakeside
Clarke Quay
Rochor
Clementi
Dover
Buona Vista
Bedok
Jurong East
Expo
Chinatown
Commonwealth
Kembangan
Aljunied
Chinese Garden
Paya Lebar
Eunos
Queenstown
Kallang
Bugis
Tanah Merah
Lavender
Redhill
one-north
Telok Ayer
City Hall
Tiong Bahru
Kent Ridge
Dakota
Simei
Outram Park
Raffles Place
Mountbatten
Esplanade
Haw Par Villa
Tanjong Pagar
Stadium
Tampines
Pasir Panjang
Nicoll Highway
Bras Basah
Downtown
Labrador Park
Promenade
Pasir Ris
Telok Blangah
Bayfront
Marina Bay
HarbourFront
Marina South Pier

Limitations and improvements

- As there are relatively few trips from some LRT stations, it was not enough to separate them
- Stations two stops apart are often linked together as there may not be enough trips between them to separate them.
- Use a heuristic to clean up the network further (e.g. unlikely for there to be 3 stations in a loop)

2

Task 2

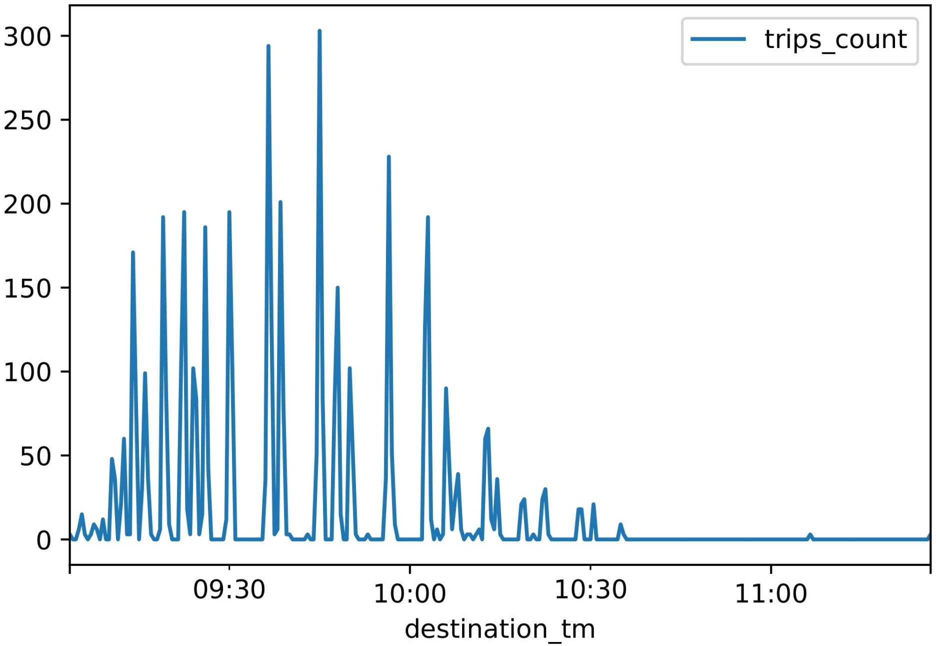
Given an origin and destination station pair (say Station A and Station B), how many trains travelled from Station A to Station B in the given dataset?

Approach outline

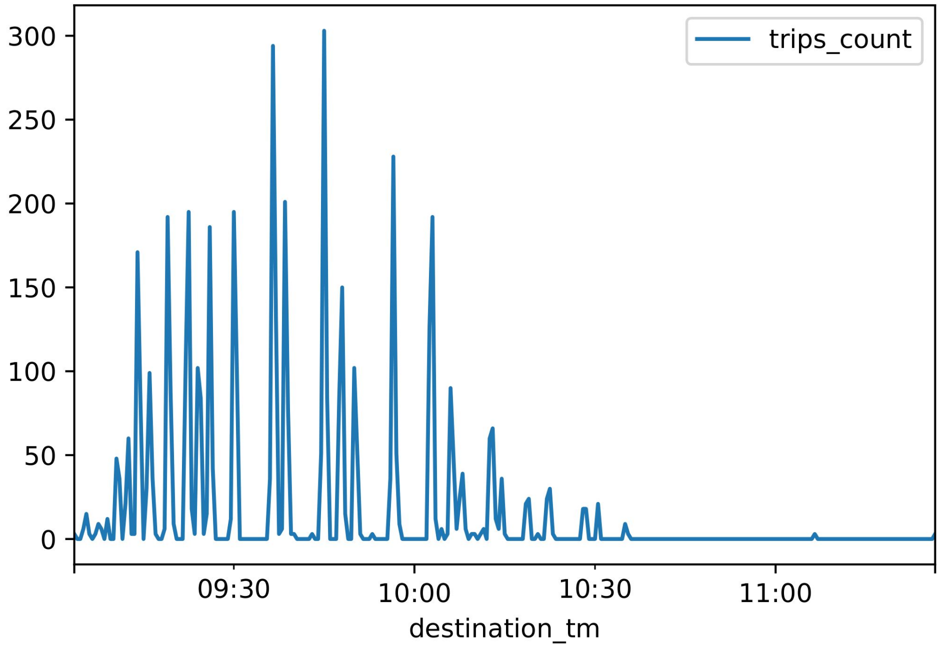
- Determine the line and direction of A to B
- Filter down to a list of trips on this line and in this direction
- Find the most frequent destination station
- Group the time series into 30 second blocks and detect number of spikes using mean absolute deviation

Checking results

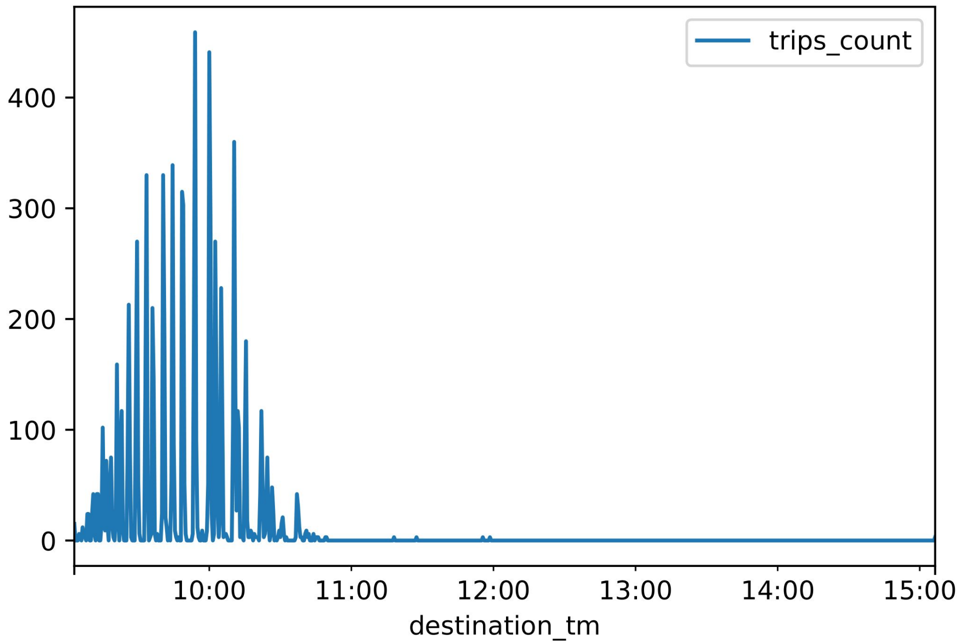
- Two stations that are on the same line and in the same direction, should yield the same result (e.g. Clementi to Kallang should be the same as Boon Lay to Clementi)
- During peak hour, train frequencies are roughly around 2 – 3 minutes each



Boon Lay - Pasir Ris (24)



Pioneer - Boon Lay (24)



Ang Mo Kio - Toa Payoh (33)

Limitations and improvements

- Does not take into account trips that start and end on different lines. This would require us to know the route, which we can know from calling the Google API, if needed.

3

Task 3

Determine the number of people in the train at every station along a commuter’s path.

Approach outline

- Assuming only one train in one direction is at a station at a given time (i.e. there won't be two East-West trains at the station), getting the number of people at Clementi in the same direction at the same time should always give the same result
- Construct a dictionary that takes the line, station, time (rounded off to nearest minute) and returns the number of people
- Given two stations, find the relevant stations between them, and query this dictionary, assuming each train stop takes 2 minutes

Example: Constructing the dictionary

Trip: Bishan to Novena, 9.15 – 9.30

[Bishan – Braddell – Toa Payoh – Novena]

- Add 1 to number of people at Bishan at 9.15
- Add 1 to number of people at Braddell at 9.20
- Add 1 to number of people at Toa Payoh at 9.25

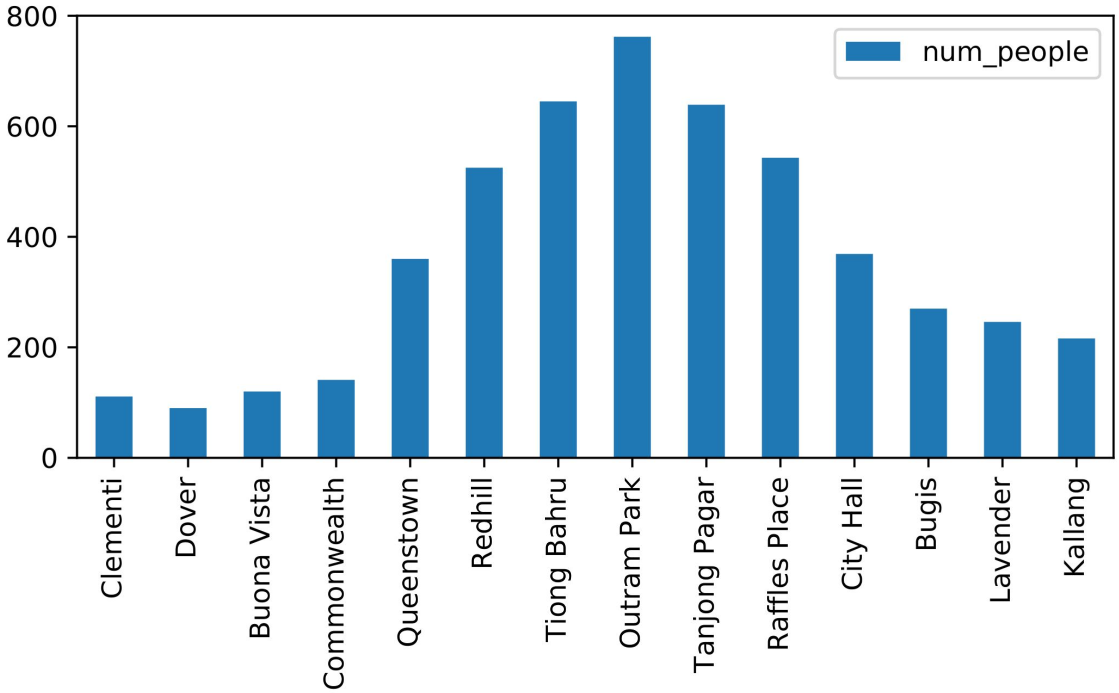
etc. and do this for every trip.

Example: Getting no. of people

To get number of people from Clementi to Kallang at 9am,

- Count at Clementi = Clementi at 9:00 + Clementi at 9.01
- Count at Dover = Dover at 9:02 + Dover at 9:03
- Count at Buona Vista = BV at 9:04 + BV at 9:05

etc.



Clementi - Kallang

Limitations and improvements

- Does not take into account people who change from one line to another
- Does not account for variability of distance between stations, can be improved with information about distances between stations
- Efficiency: Constructing the dictionary is time consuming (no. stations x no. stops in each station), can be possibly improved by using a data structure more suitable for range updates (e.g. a segment tree)

Thanks!

Any questions ?