Visualising Spatial Data in R with Interactive Maps

(or Plotting Public Transport Maps using GTFS and R)

Belinda Maher
Data Analyst

Swinburne University
Ex Public Transport Victoria

about.me/minga

Tableau maps in this presentation available at:
public.tableau.com/profile/belinda.maher/

Confession:
I love maps.

And I love trains.

So, I am quite fond of public transport maps.

Plotting maps in R is not something I ever thought I would get paid to do.

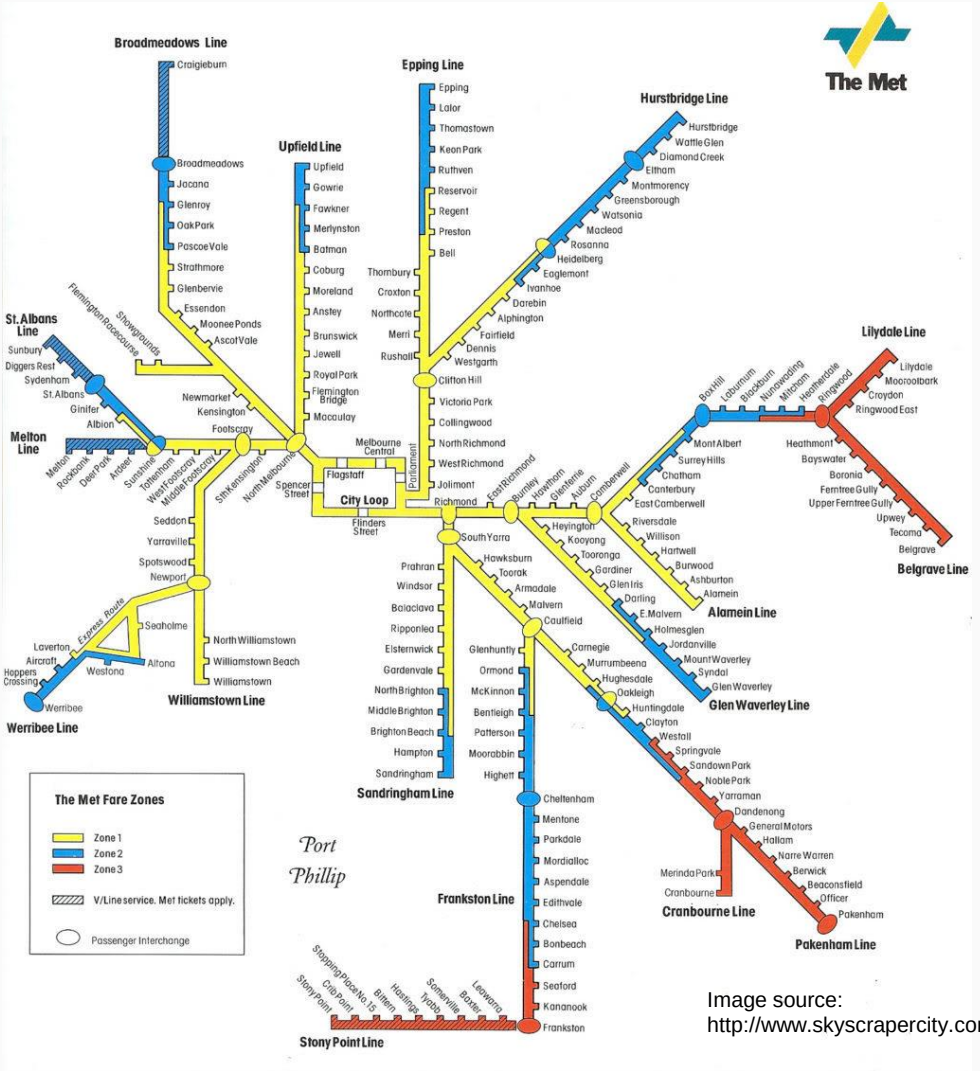


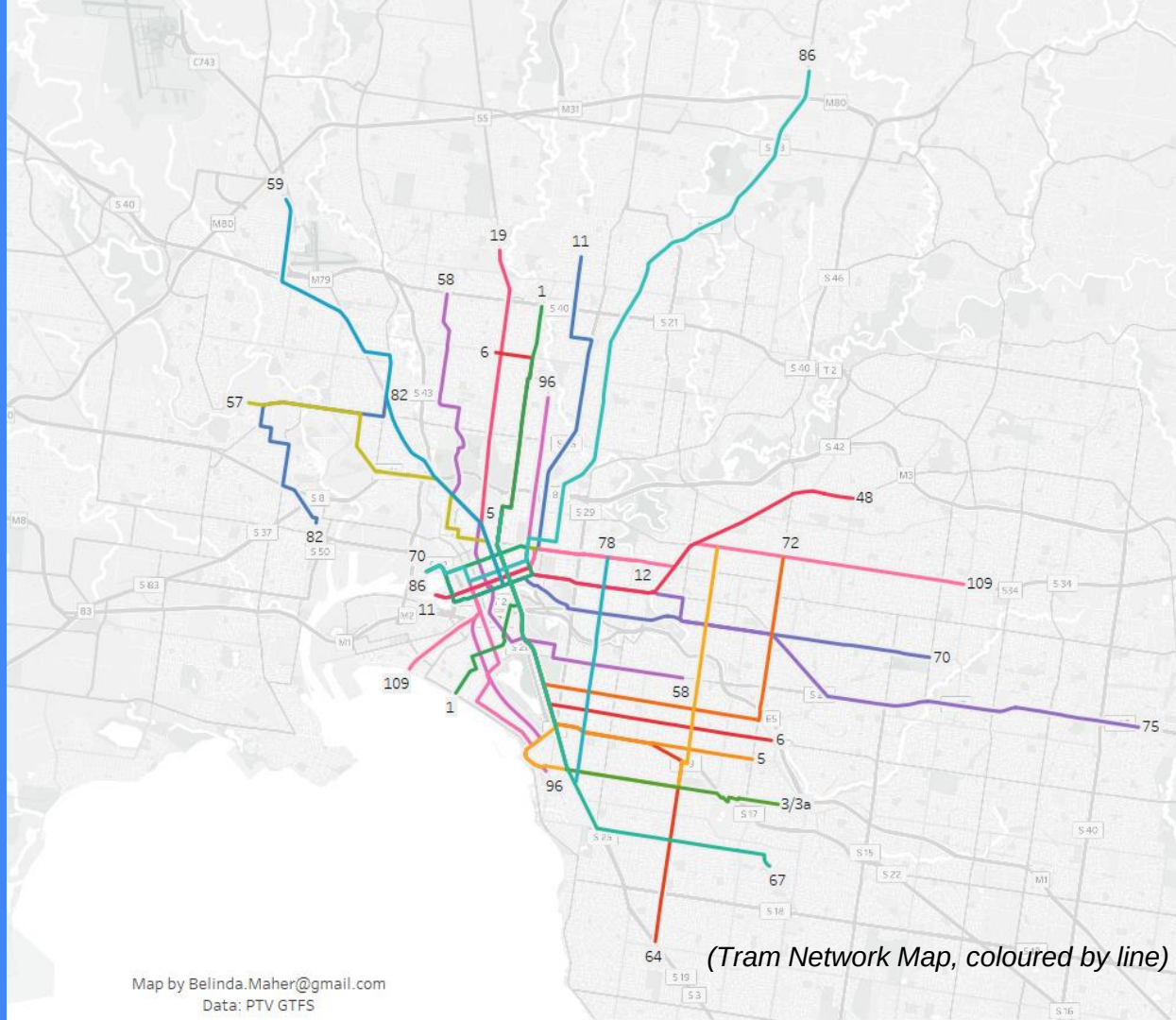
Image source:
http://www.skyscrapercity.com

Working at PTV

In 2016 I started at PTV in the Operational Performance Analysis team

where I drew some public transport maps in R

using geographical data in the publicly available GTFS dataset



(Tram Network Map, coloured by line)

What does this talk cover?

Talk Plan

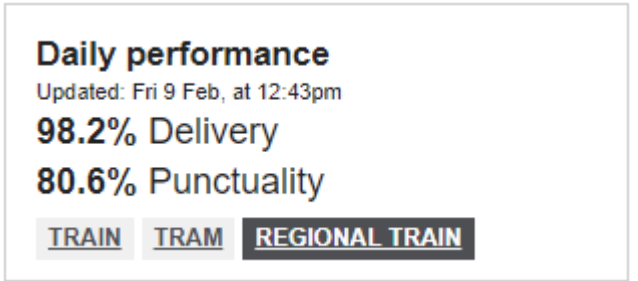
- Background – what was the challenge?
- What is GTFS?
- finding the spatial data in a haystack:
  Avoiding duplication - finding the right shapes to plot in GTFS data
- Other issues with GTFS data – errors and alternate paths
- sp Package objects: SpatialPoints and SpatialLines
- Plotting with the Leaflet package
- See also: googleway package

Background – What was the challenge?

- What is Operational Performance Analysis?
- What sort of data did we have?
- How best to visualise performance data along public transport routes?

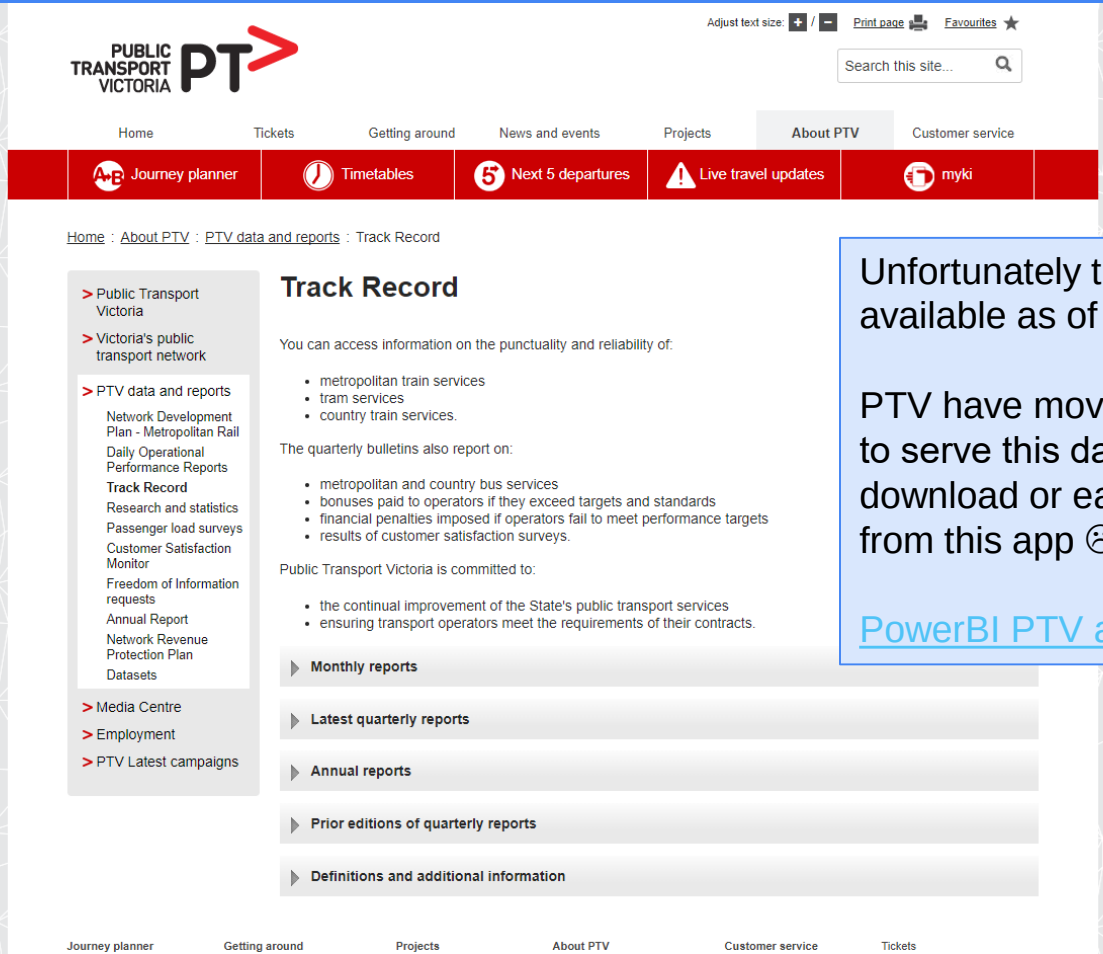
What do the Operational Performance Analysis team do?

+ Daily Operational Performance Reports on the PTV Website (also with further detail):



https://www.ptv.vic.gov.au/about-ptv/data-and-reports/operational-performance/

Track Record Data - Monthly Averages



Unfortunately this page is no longer available as of 14/02/18.

PTV have moved to a PowerBI app to serve this data. You can't download or easily copy the tables from this app ☹

PowerBI PTV app

What sort of data did we have?

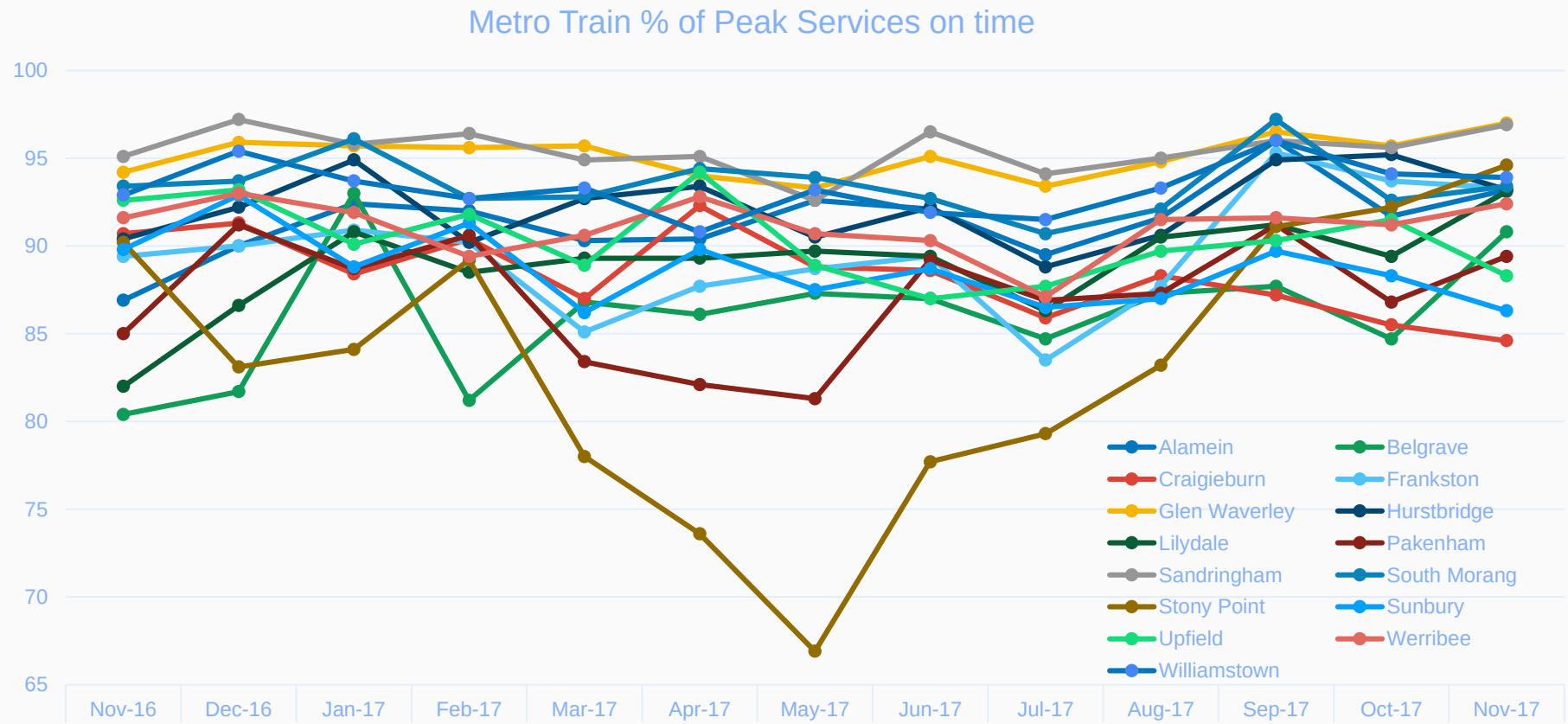
Example: Some track record data from the PTV website

Percentage of peak services on-time

	2016		2017											
Line	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	12 mth avg
Alamein	86.9	90.0	92.4	92.0	90.3	90.4	92.6	92.1	89.5	91.6	96.0	91.7	93.2	91.8
Belgrave	80.4	81.7	93.0	81.2	86.8	86.1	87.3	87.0	84.7	87.3	87.7	84.7	90.8	86.6
Craigieburn	90.7	91.3	88.4	90.4	87.0	92.3	88.8	88.6	85.9	88.3	87.2	85.5	84.6	88.1
Cranbourne	86.4	87.0	87.5	87.4	80.7	80.2	79.1	85.6	85.2	86.9	90.1	87.6	88.2	85.4
Frankston	89.4	90.0	90.9	90.2	85.1	87.7	88.7	89.4	83.5	87.7	95.3	93.7	93.3	89.6
Glen Waverley	94.2	95.9	95.7	95.6	95.7	94.0	93.3	95.1	93.4	94.8	96.5	95.7	97.0	95.2
Hurstbridge	90.4	92.2	94.9	90.2	92.7	93.4	90.5	92.2	88.8	90.6	94.9	95.2	93.2	92.4
Lilydale	82.0	86.6	90.8	88.5	89.3	89.3	89.7	89.4	86.3	90.5	91.2	89.4	93.1	89.6
Pakenham	85.0	91.2	88.7	90.6	83.4	82.1	81.3	89.2	86.9	87.3	91.2	86.8	89.4	87.3
Sandringham	95.1	97.2	95.8	96.4	94.9	95.1	92.6	96.5	94.1	95.0	96.0	95.6	96.9	95.5
South Morang	93.4	93.7	96.1	92.7	92.8	94.4	93.9	92.7	90.7	92.1	97.2	92.6	93.4	93.5
Stony Point	90.2	83.1	84.1	89.2	78.0	73.6	66.9	77.7	79.3	83.2	91.1	92.2	94.6	83.2
Sunbury	89.8	92.9	88.8	91.3	86.2	89.8	87.5	88.7	86.5	87.0	89.7	88.3	86.3	88.5
Upfield	92.6	93.2	90.1	91.8	88.9	94.2	88.9	87.0	87.7	89.7	90.3	91.5	88.3	90.1
Werribee	91.6	93.0	91.9	89.4	90.6	92.8	90.7	90.3	87.1	91.5	91.6	91.2	92.4	91.0
Williamstown	92.9	95.4	93.7	92.7	93.3	90.8	93.2	91.9	91.5	93.3	96.0	94.1	93.9	93.4
Network total	89.6	91.6	91.8	90.8	89.1	90.2	89.1	90.5	88.0	90.2	92.8	91.1	91.8	90.6

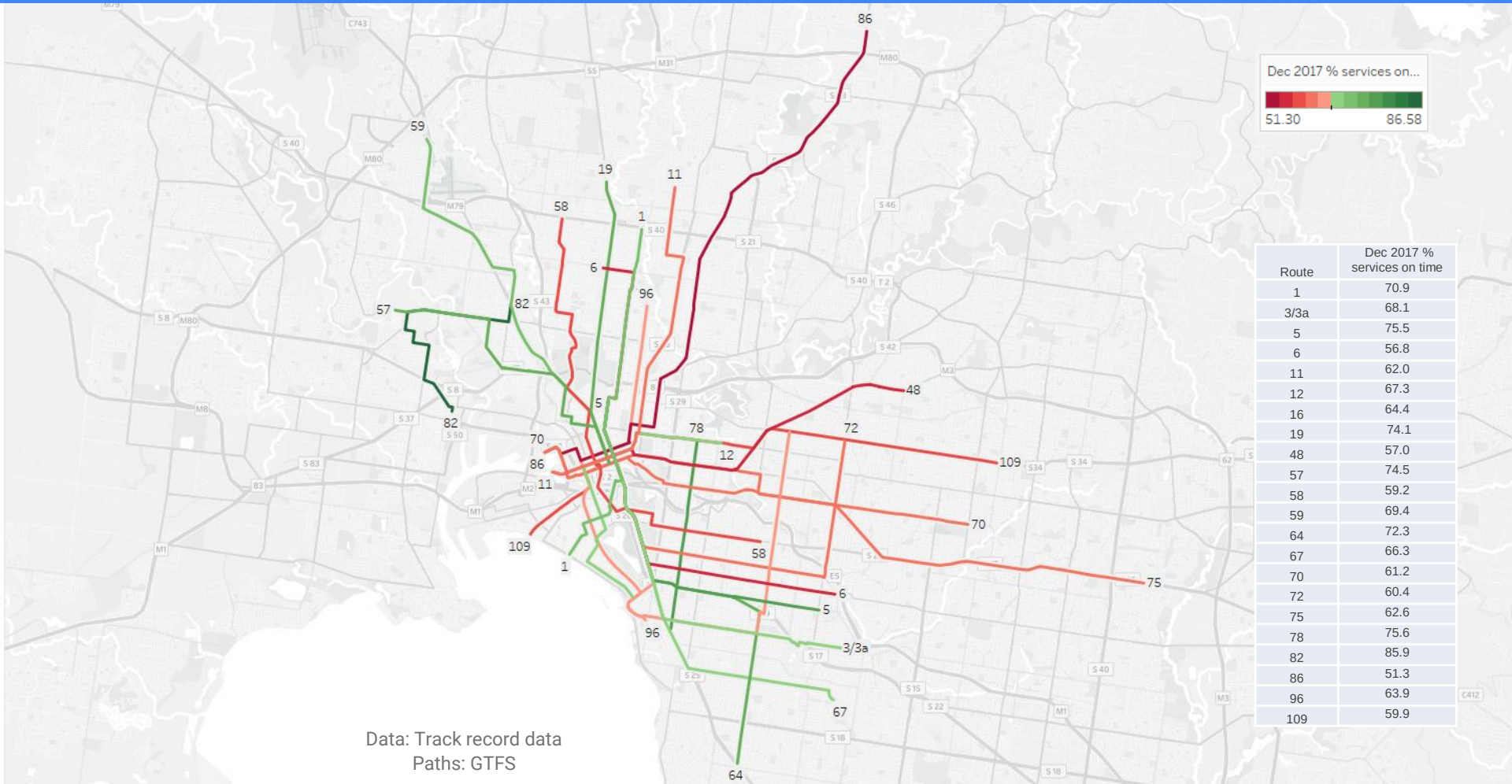
Edit: 14/2 (Unfortunately no longer available to download in this format ☹)

The trouble with line graphs: They're not very spatially clear

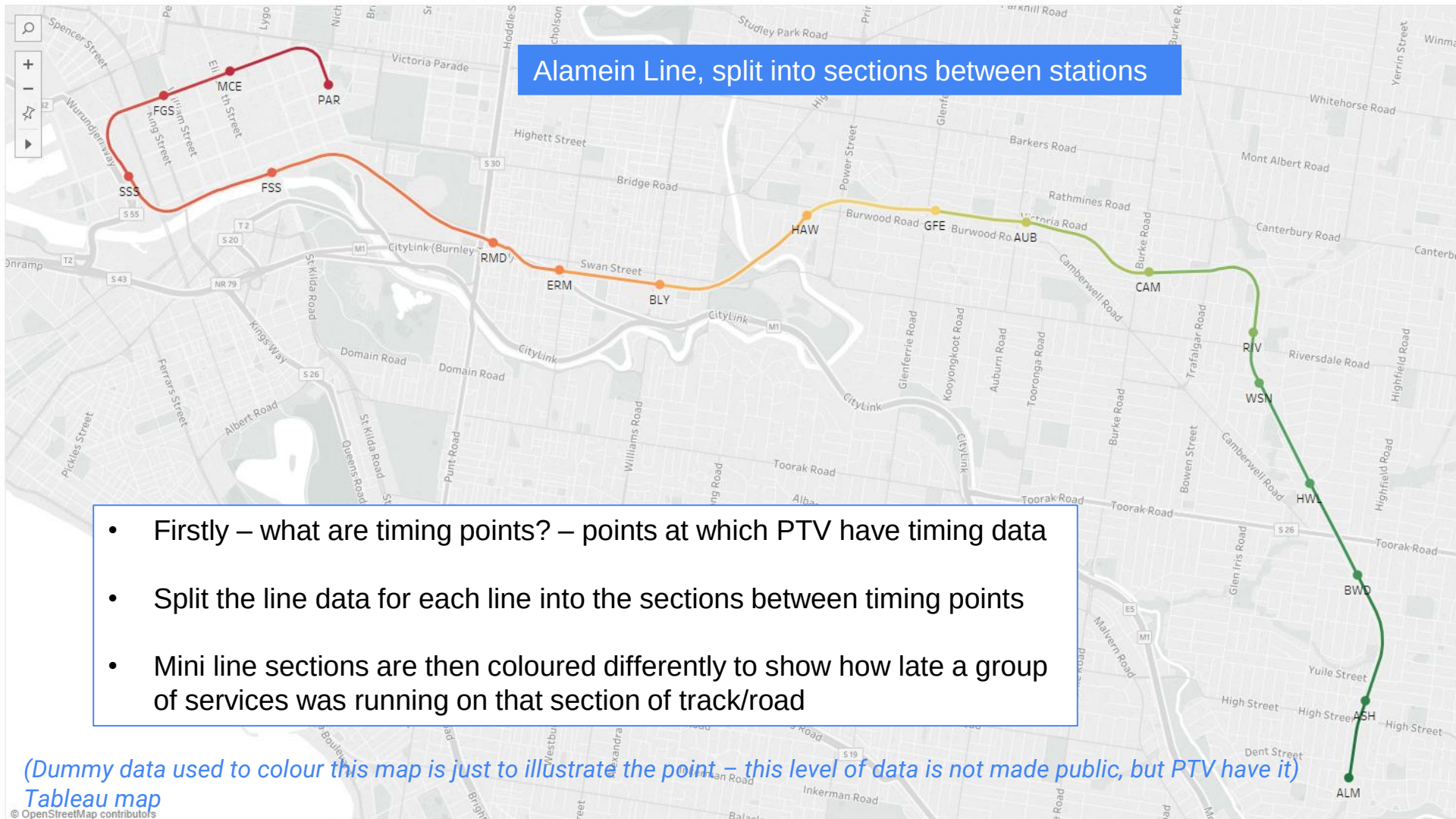


What did we want to do with this data?

December 2017 Tram % On Time Performance



Route	Dec 2017 % services on time
1	70.9
3/3a	68.1
5	75.5
6	56.8
11	62.0
12	67.3
16	64.4
19	74.1
48	57.0
57	74.5
58	59.2
59	69.4
64	72.3
67	66.3
70	61.2
72	60.4
75	62.6
78	75.6
82	85.9
86	51.3
96	63.9
109	59.9

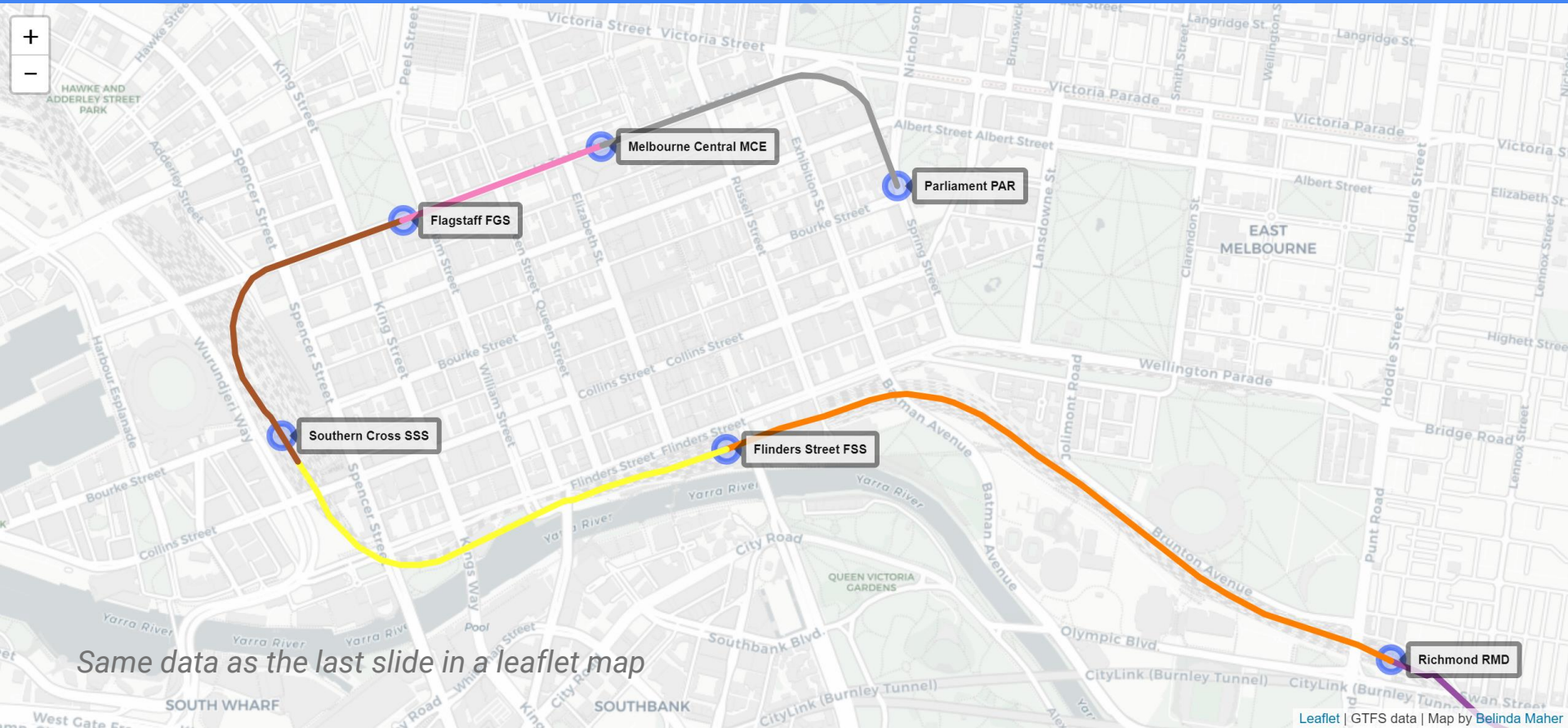


- Firstly – what are timing points? – points at which PTV have timing data
- Split the line data for each line into the sections between timing points
- Mini line sections are then coloured differently to show how late a group of services was running on that section of track/road

(Dummy data used to colour this map is just to illustrate the point – this level of data is not made public, but PTV have it)

Tableau map

# A bit more detail – line sections between timing points
leaflet()



Same data as the last slide in a leaflet map

PTV GTFS Data

…is not really intended to be used for this!

Working with GTFS Data - Plan

- What is GTFS?

What did I need to do with the data?

- Picking out the duplicates
- Checking that you have the full line in each case
- Then splitting the line in to the sections
- And then labelling those sections

So, what is GTFS anyway?

What is GTFS?

- An open standard for publishing public transport timetables, stop locations and transport service path information
- The General Transit Feed Specification was invented in 2005 in Portland, Oregon
- It was originally called the Google Transit Feed Specification
- PTV first published a GTFS feed of their timetable data in 2015

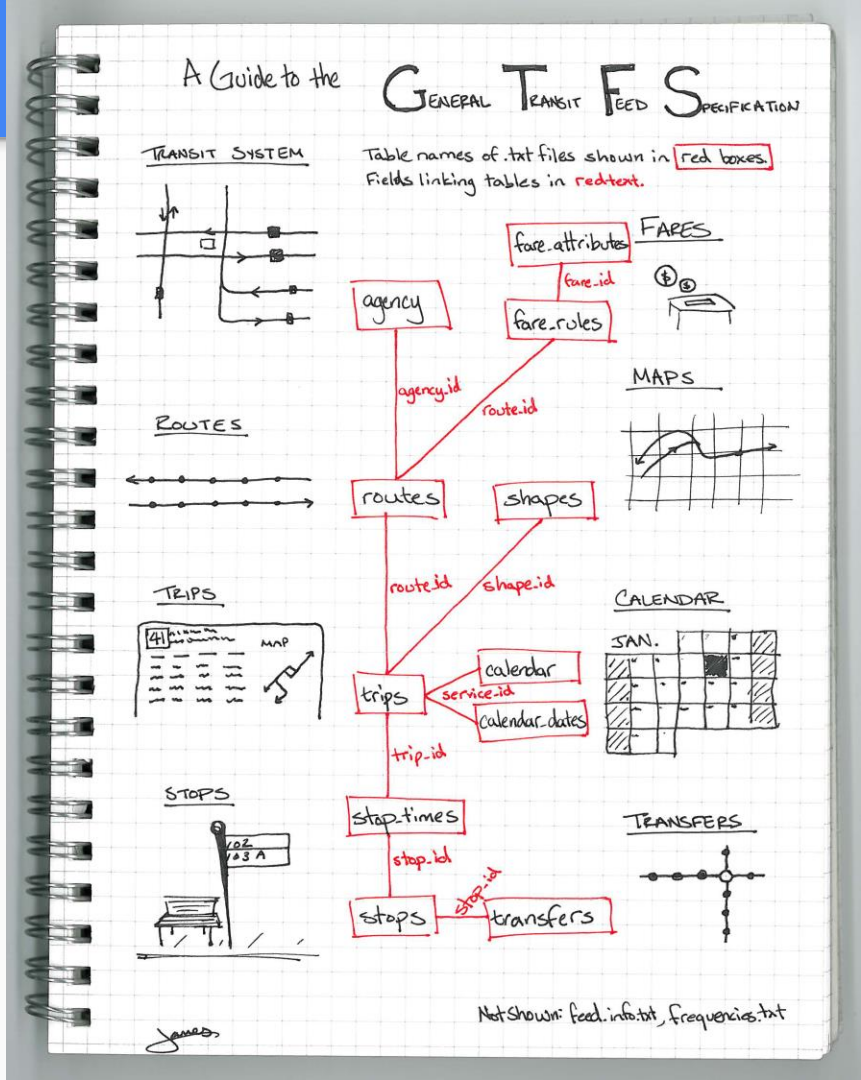
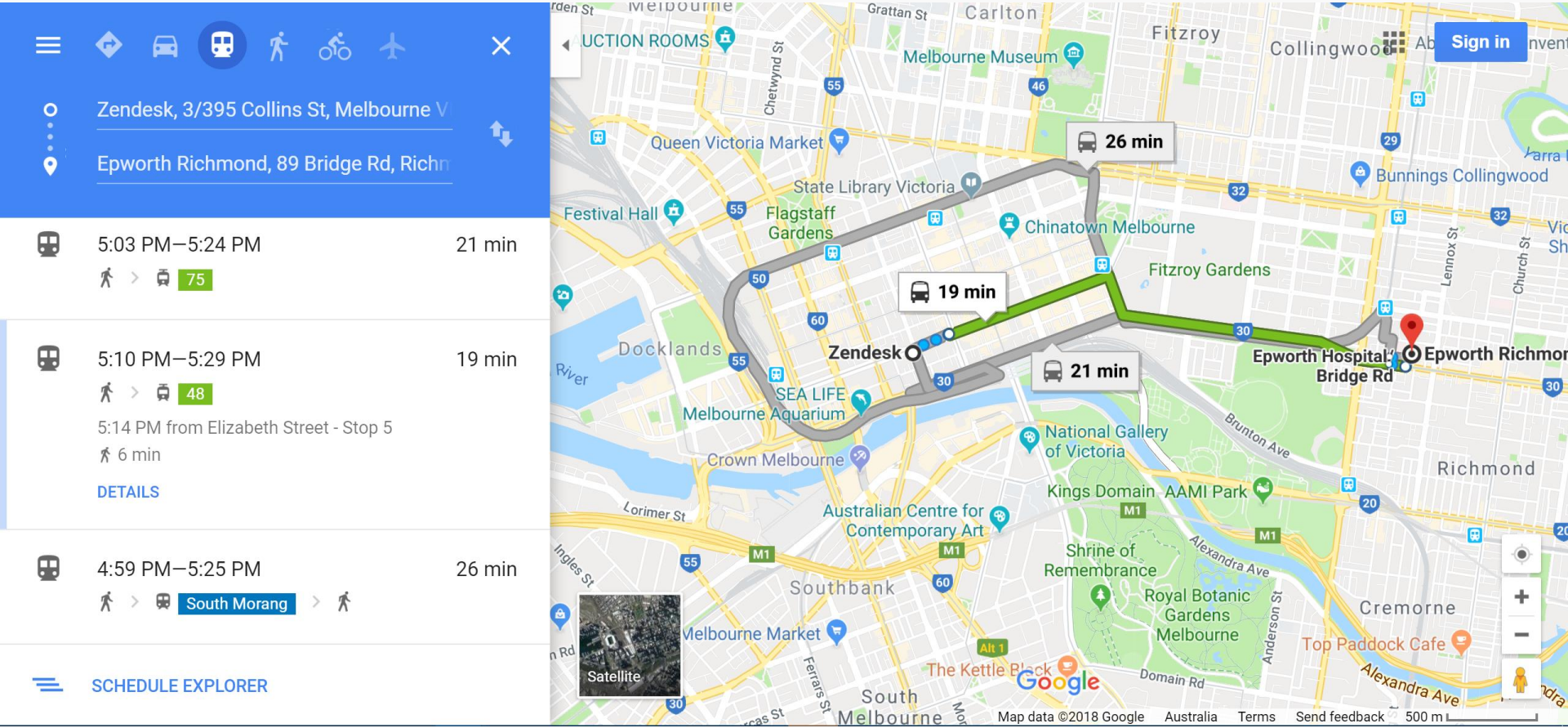


Image source:
http://blog.openplans.org/2012/08/the-openplans-guide-to-gtfs-data/

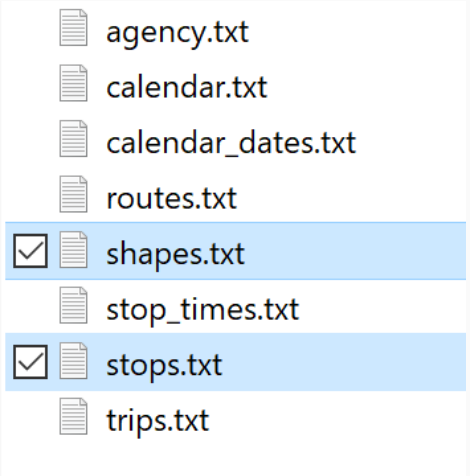
So, what is GTFS data designed for?



Where do I get the data?

Latest PTV GTFS zip file
(unlike the data.vic.gov.au file)
http://data.ptv.vic.gov.au/downloads/gtfs.zip

The data is then in folders by operator,
with another zip file underneath



It’s a bit complicated

Thankfully, we don’t need to understand all of this information to draw a map!

The main files we need are:
Stops.txt for stations

Shapes.txt for paths
(but the name for the shape ‘shape_id’ is not very meaningful in terms of exactly which route you are looking at without further joining to routes.txt via trips.txt or plotting the data)

Additional info in **Routes.txt** for route names and numbers.

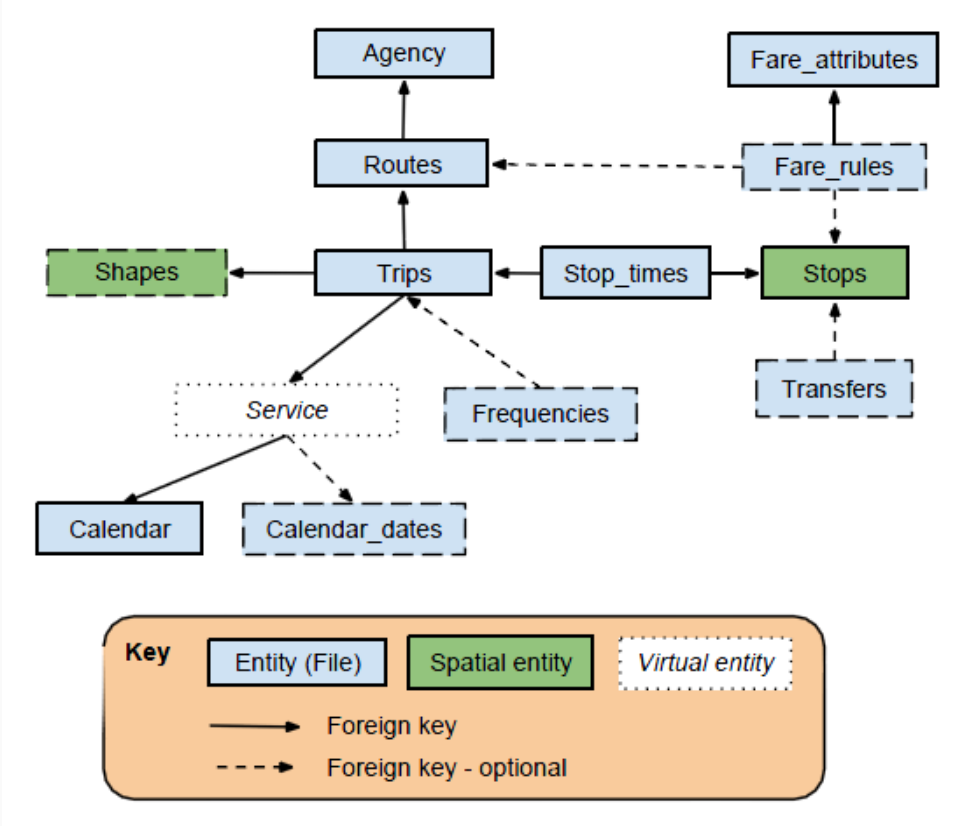
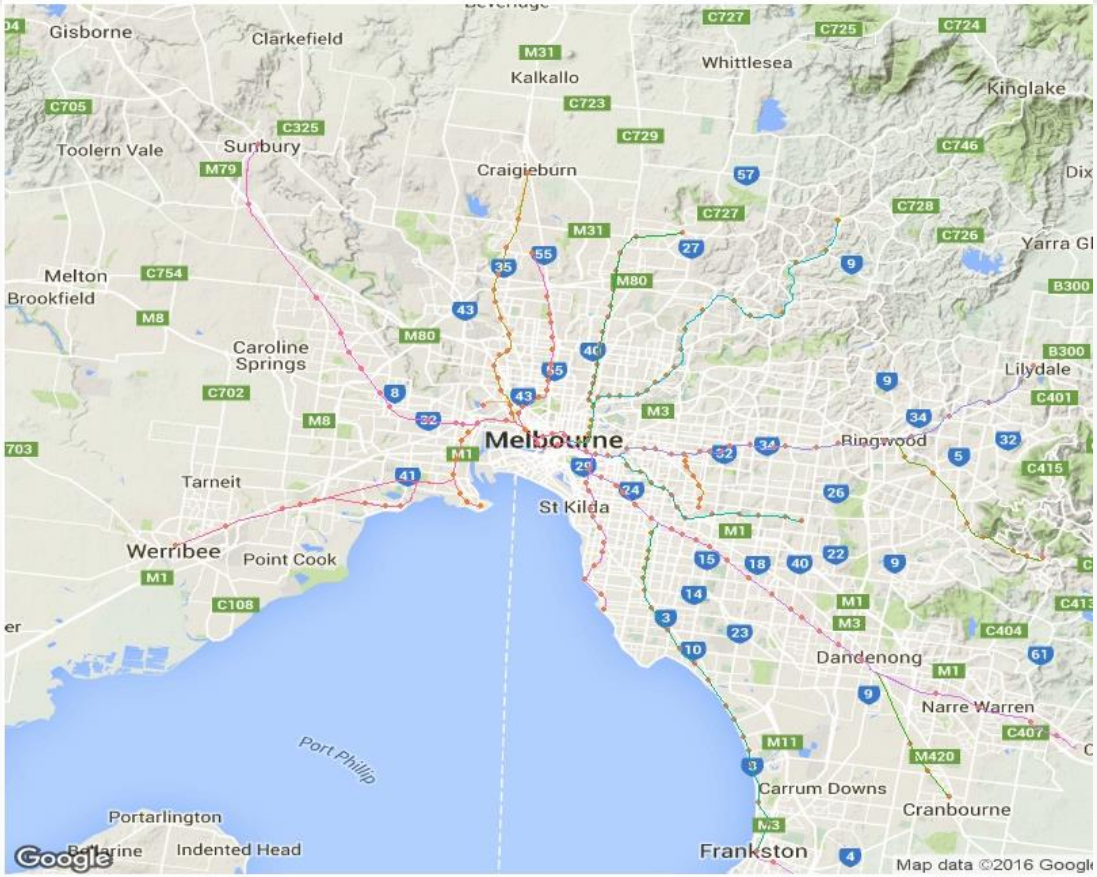


Diagram source: http://lin-ear-th-inking.blogspot.com.au/2011/09/data-model-diagrams-for-gtfs.html

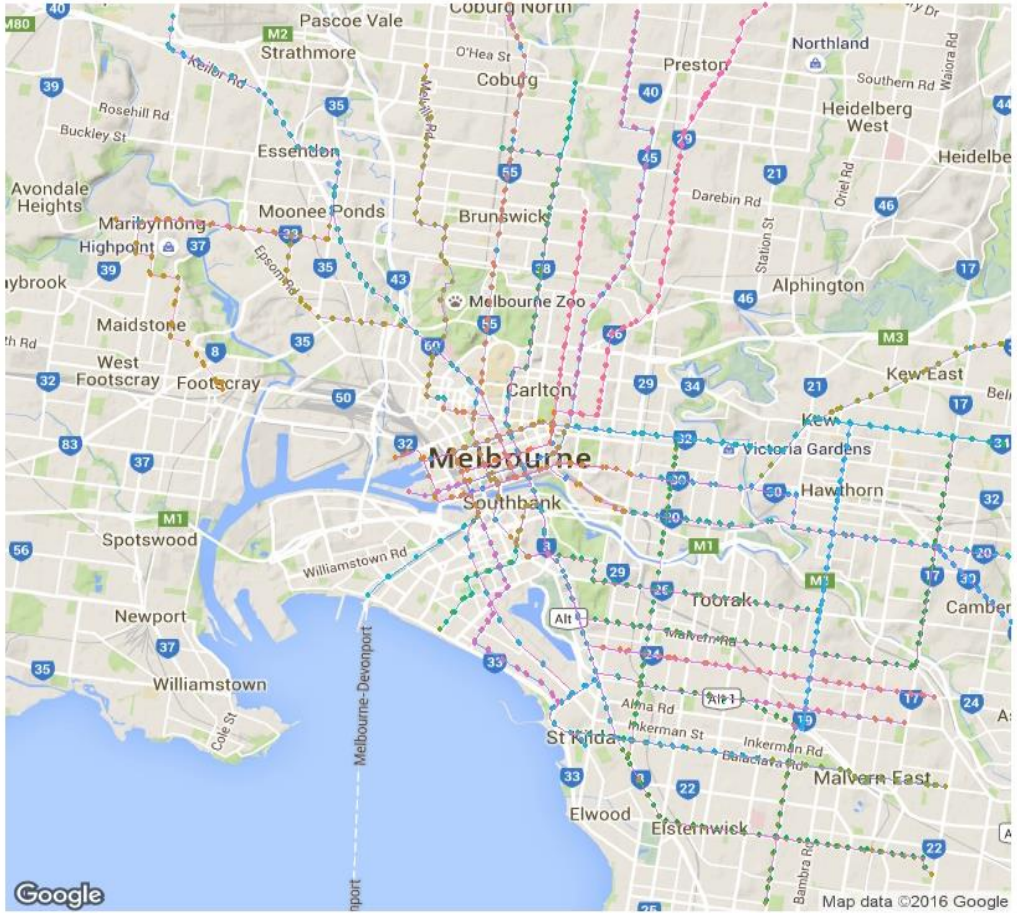
A first look at the data:
The trouble with duplication

One of my first maps in R: ggplot2




```
melb_map <- get_map("Melbourne, Australia",
zoom =10)

(train_map<-ggmap(melb_map) +
geom_point(data=metro_stops,
aes(x=stop_lon, y=stop_lat,
color = factor(stop_id)), size=1)+
theme(legend.position="none")+
geom_path(data=metro_shapes,
aes(shape_pt_lon, shape_pt_lat, group =
shape_id, color = factor(shape_id)),size = .1,
alpha = .1) +
coord_equal() +
theme_map()+
theme(legend.position="none"))
```

Attempt #2: Tram Network



Drawbacks for these maps:

- Not very clear
- A bit of a pain to go back into the code to zoom or centre
- Takes ages to plot – not suitable for running within a dashboard

GTFS Path data (shapes.txt) plus stops.txt for the tram network (using ggplot2)

Shapes.txt

Why were those plots so slow to draw?

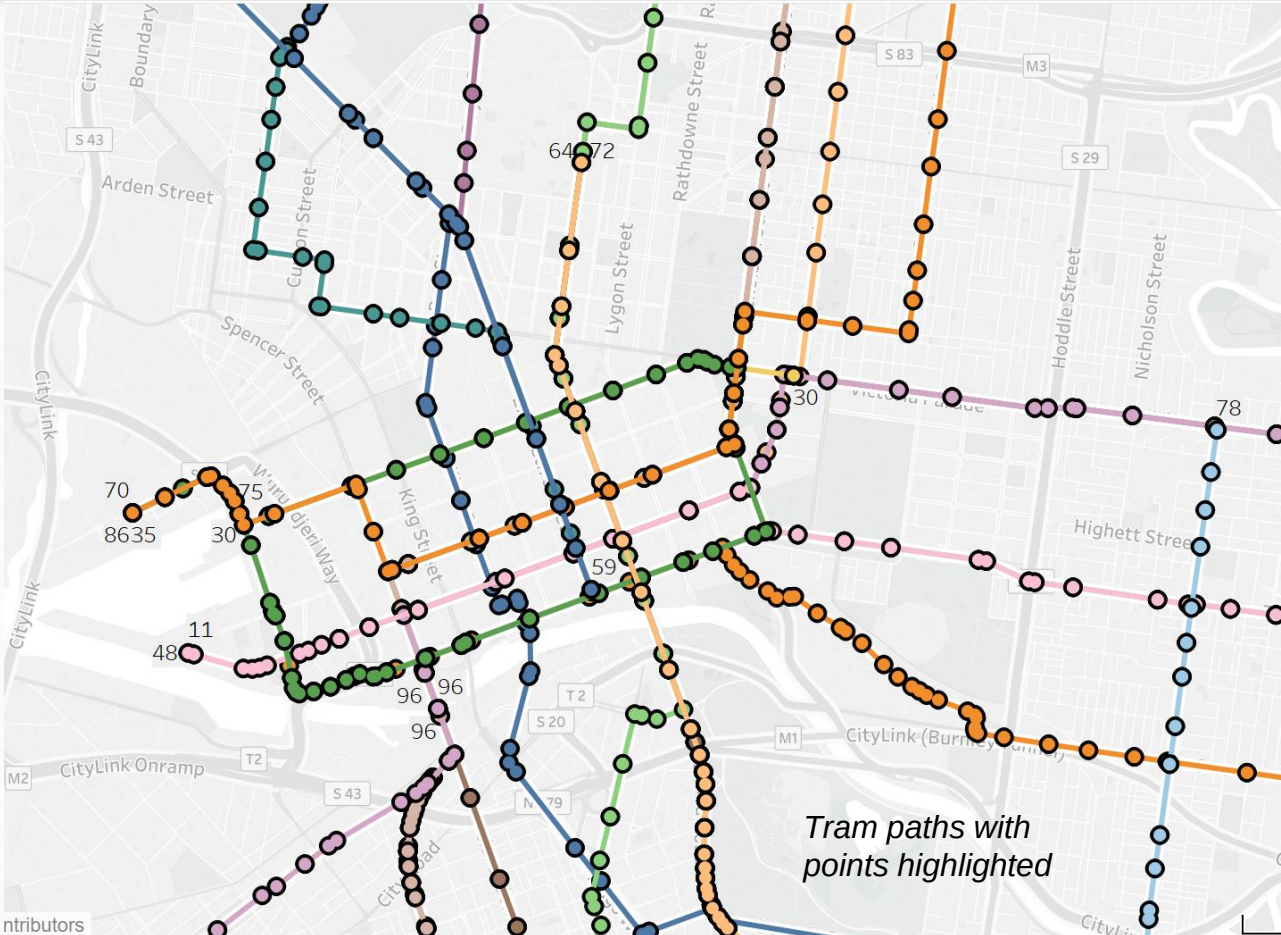
I looked at how big the files are:

10.1 MB for tram, 52,660 points

31.6 MB for metro trains, 393k points!
(Metro train data is only 120 Kb without duplicated routes, ~3400 points)

The data isn't at high resolution, and on inspection contains lots of duplicate shapes.

So, why is there so much duplication?



Why so many shapes?

Just some of the possible paths for an Alamein train:

ALM-PAR	ALM-FSS	RIV-PAR	ALM-FSS	RIV-FSS	ALM-CAM
City Loop	City Loop	City Loop	Direct	Direct	(Direct)

- There are multiple copies of each of these shapes with different shape_ids in the data, including identical lines going in opposite directions.
- PTV don’t seem to have altered the ‘trip_headsign’ field for different termini for train, which would help to differentiate the lines in the data with destinations which aren’t at the track terminus

A quick(ish) way of solving the problem: longest shapes

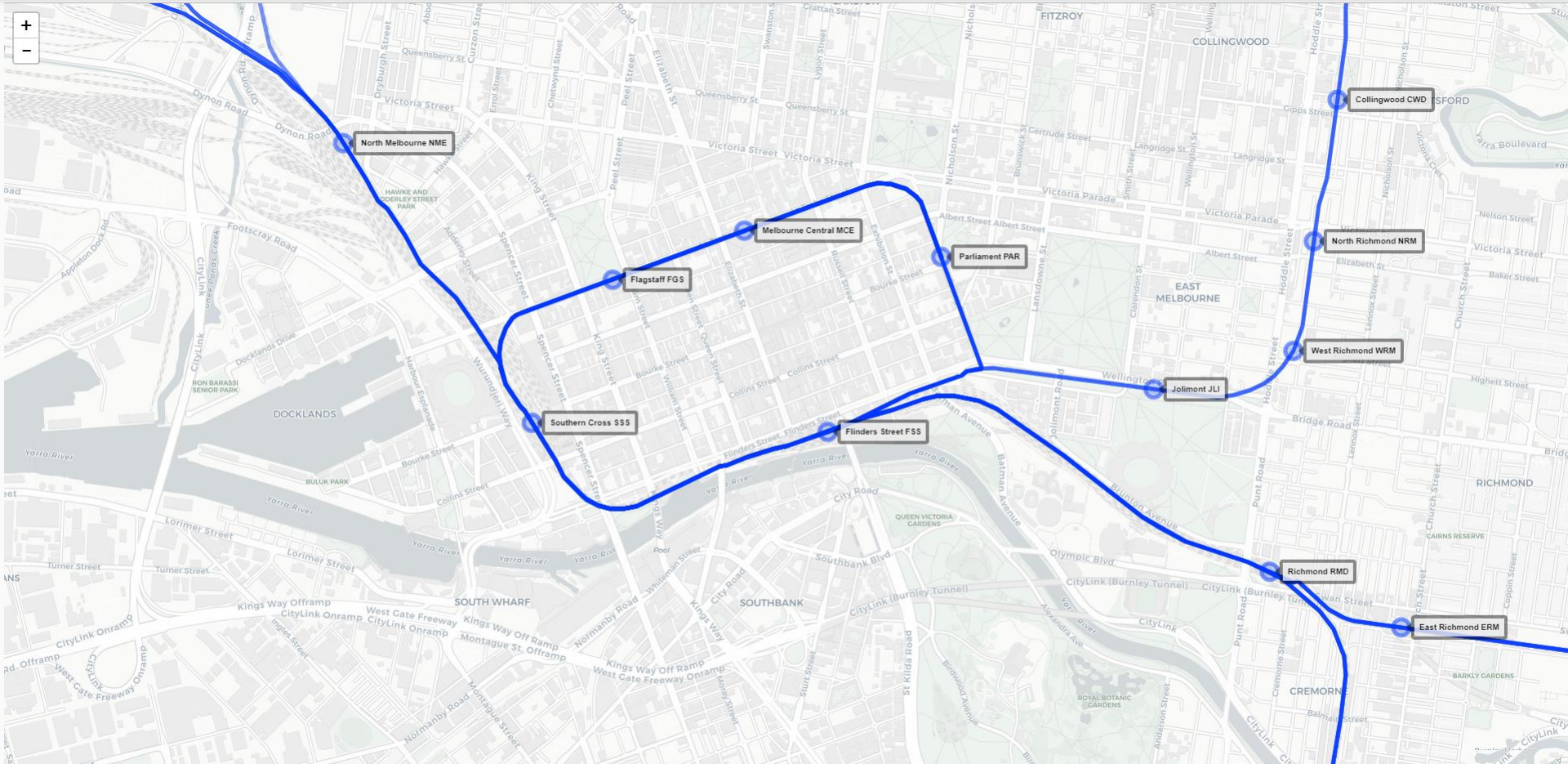
Instead of looking at where each line starts and ends or what points it travels through to find a complete line,

Why not use the field shape_dist_travelled which is in the path data?

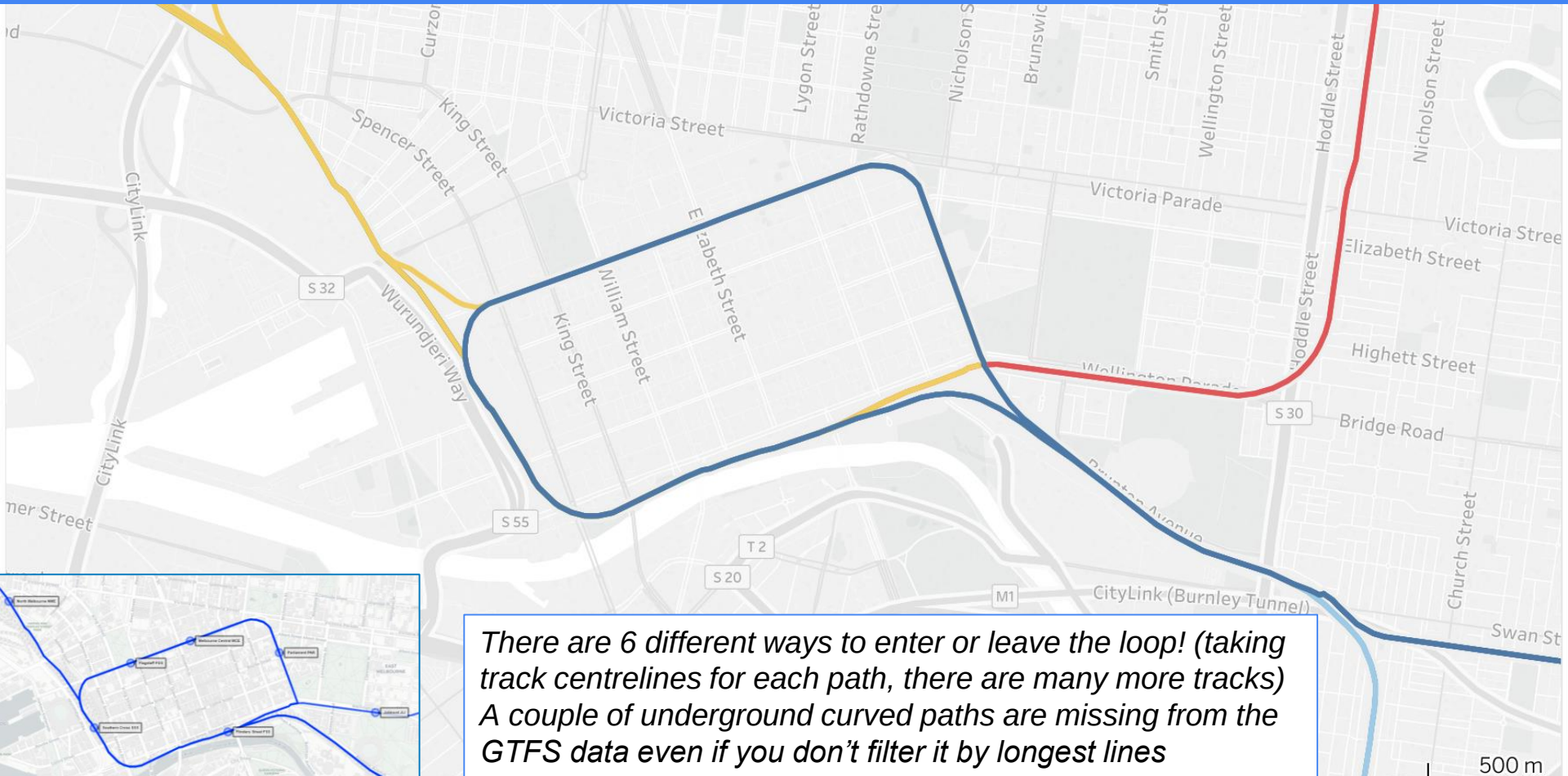
(I also needed to select only lines travelling from the terminus towards the city to assist with labelling the track sections in order, but I won't go in to this)

Metro train map coloured by shape_dist_travelled – Tableau, BM

Longest Paths – Can you tell me what's missing?

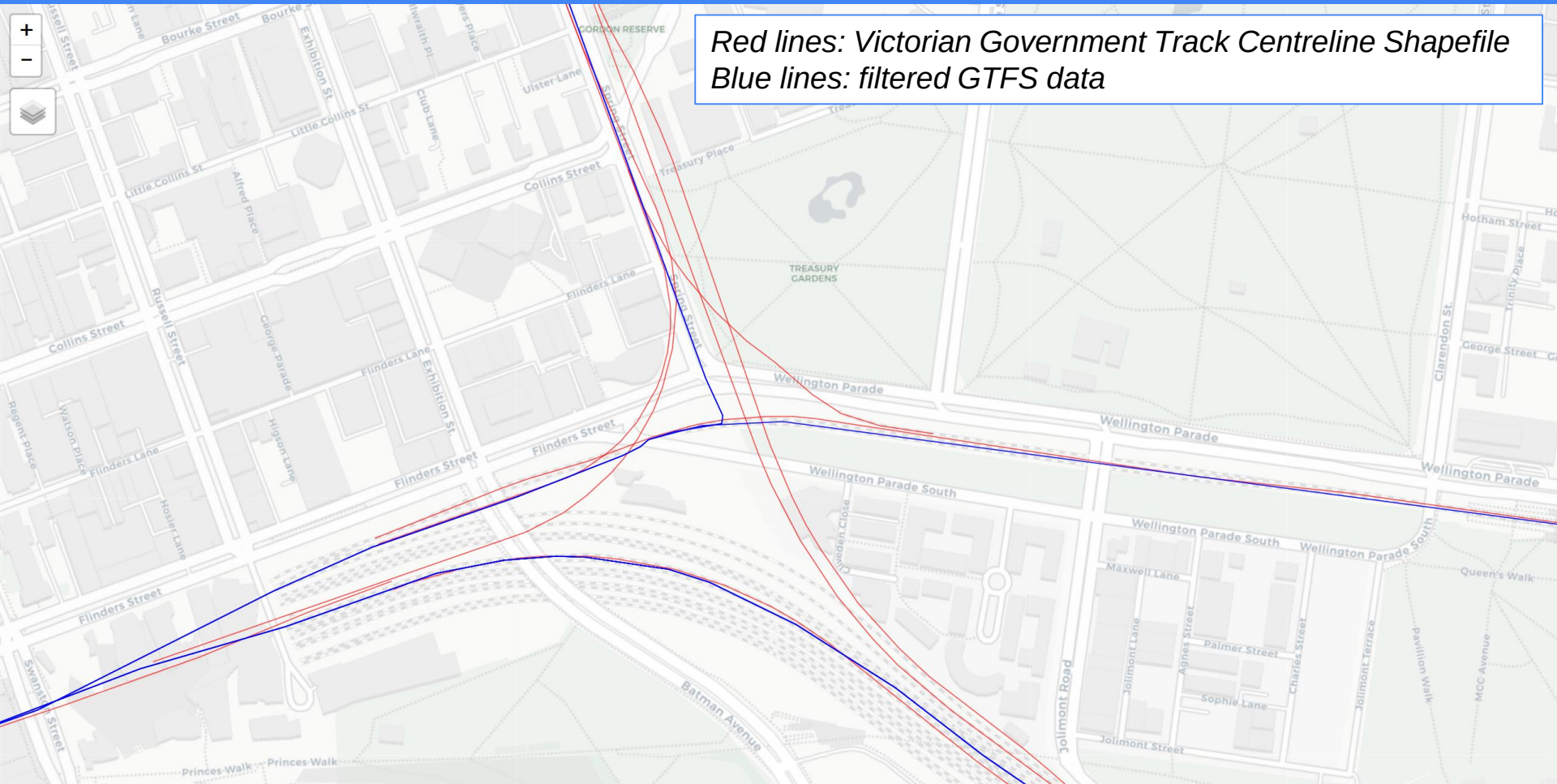


The problem with paths

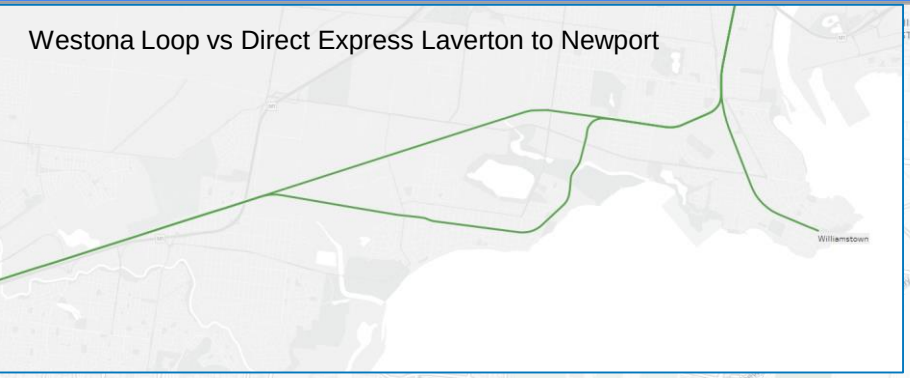


There are 6 different ways to enter or leave the loop! (taking track centrelines for each path, there are many more tracks)
A couple of underground curved paths are missing from the GTFS data even if you don't filter it by longest lines

Missing Curves: Flinders Street to Parliament, and Jolimont to Parliament



The longest shape won’t always give you the whole line



Tottenham TOT
West Footscray WFY
South Kensington SKN
North Melbourne NME
Seddon SEN
Yarraville YVE
Spotswood SPT
Newport NPT
North Williamstown NWN
Williamstown Beach WBH
Williamstown WIL
Laverton LAV
Aircraft ACF
Williams Landing WLD
Westona WTO
Altona ALT
Seaholme SHE
Hoppers Crossing HCG
Werribee WER

Splitting the lines at stations

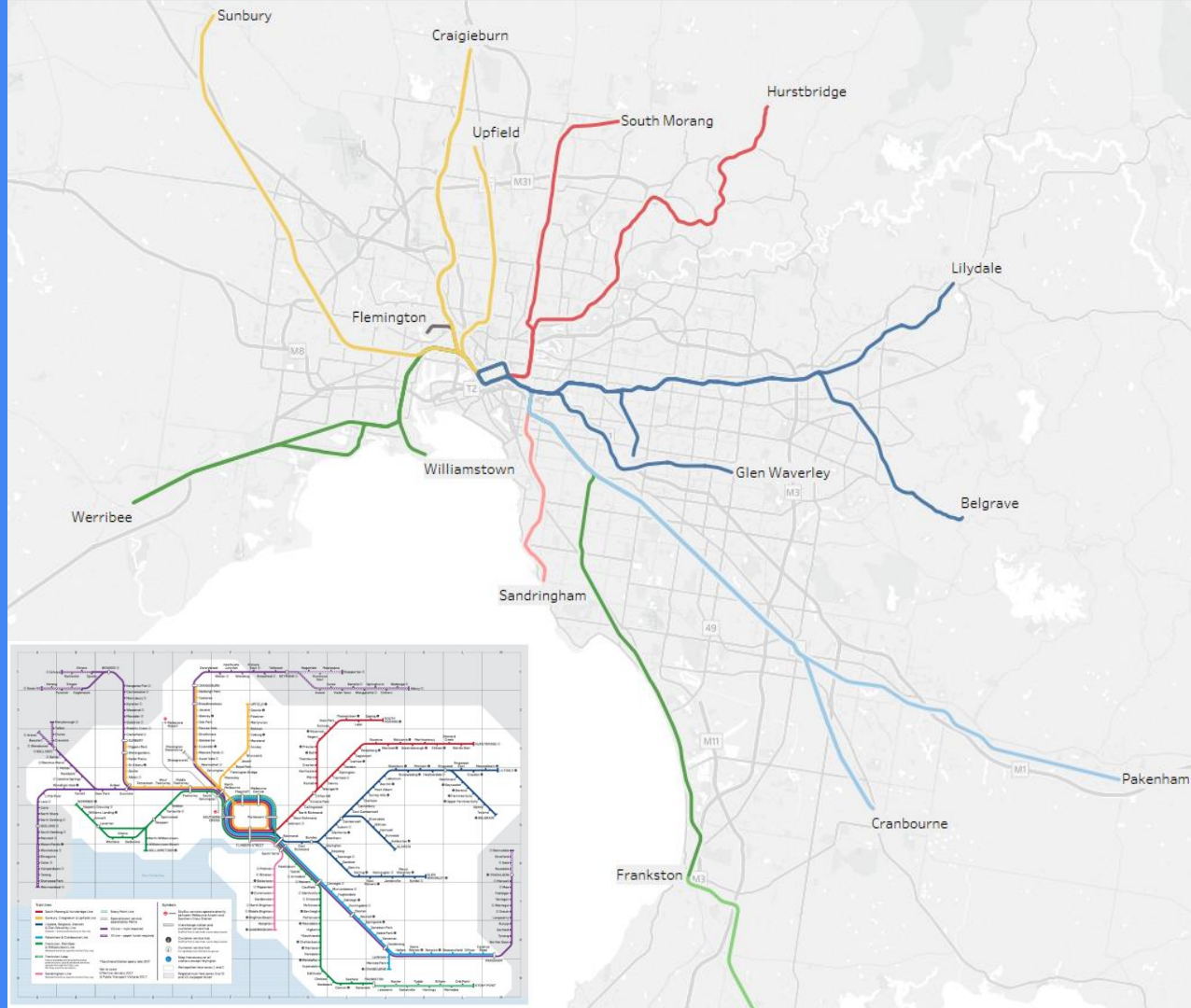
- I looked into using spatial functions to cut the line at station points
- But then I looked at the data and saw that each line is already cut at stations, but it is not labelled. There is a duplicate data point at each station/stop in the data
- So I allocated a different section name at each point where there was a duplicate point

Allocating section names in shape data

shape_id	lat	lon	shape_pt_sequence	shape_dist_traveled	line	section	name	split	ID
2-ALM-A-mjp-1.12.R	-37.8683	145.0797	1	0	ALM	1	ALM-ASH	Start	1 ALM-ASH
2-ALM-A-mjp-1.12.R	-37.8644	145.0804	2	443.5493	ALM	1	ALM-ASH	NA	1 ALM-ASH
2-ALM-A-mjp-1.12.R	-37.8638	145.0806	3	510.8174	ALM	1	ALM-ASH	NA	1 ALM-ASH
2-ALM-A-mjp-1.12.R	-37.862	145.0813	4	723.0178	ALM	1	ALM-ASH	End	1 ALM-ASH
2-ALM-A-mjp-1.12.R	-37.862	145.0813	5	723.0178	ALM	2	ASH-BWD	Start	2 ASH-BWD
2-ALM-A-mjp-1.12.R	-37.8583	145.0829	6	1153.437	ALM	2	ASH-BWD	NA	2 ASH-BWD
2-ALM-A-mjp-1.12.R	-37.858	145.083	7	1188.948	ALM	2	ASH-BWD	NA	2 ASH-BWD
2-ALM-A-mjp-1.12.R	-37.8573	145.0831	8	1270.25	ALM	2	ASH-BWD	NA	2 ASH-BWD
2-ALM-A-mjp-1.12.R	-37.8563	145.0831	9	1380.363	ALM	2	ASH-BWD	NA	2 ASH-BWD
2-ALM-A-mjp-1.12.R	-37.8555	145.0829	10	1471.369	ALM	2	ASH-BWD	NA	2 ASH-BWD
2-ALM-A-mjp-1.12.R	-37.8546	145.0824	11	1578.144	ALM	2	ASH-BWD	NA	2 ASH-BWD
2-ALM-A-mjp-1.12.R	-37.8516	145.0805	12	1951.735	ALM	2	ASH-BWD	End	2 ASH-BWD
2-ALM-A-mjp-1.12.R	-37.8516	145.0805	13	1951.735	ALM	3	BWD-HWL	Start	3 BWD-HWL
2-ALM-A-mjp-1.12.R	-37.8484	145.0785	14	2342.326	ALM	3	BWD-HWL	NA	3 BWD-HWL
2-ALM-A-mjp-1.12.R	-37.844	145.0756	15	2899.073	ALM	3	BWD-HWL	End	3 BWD-HWL

Spatial Data in R

- sp
- Spatial Points
- Spatial Lines
- leaflet
- googleway



library(sp)
for spatial objects

Three main types of spatial data

Points, Lines and Polygons

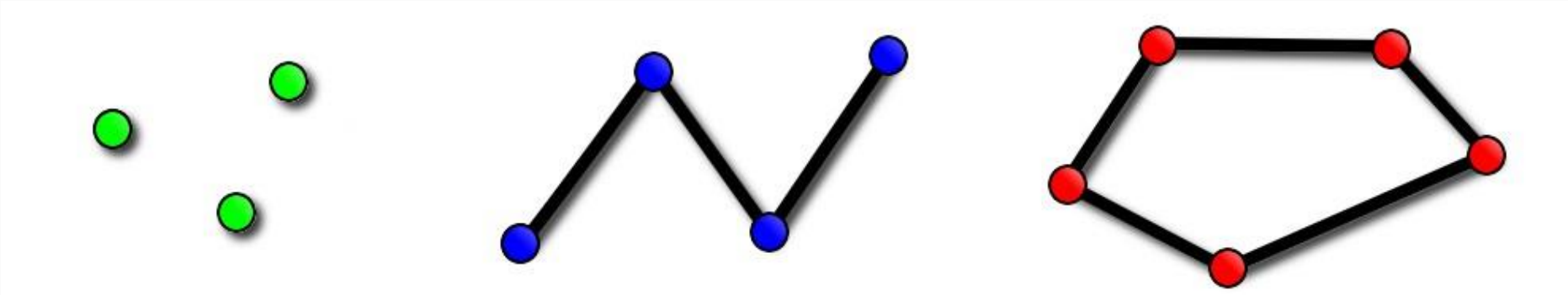
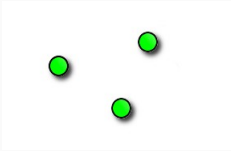


Image source: https://geographicprimitive.wordpress.com/

Spatial Objects for GTFS Data in R (helpful for plotting in leaflet)

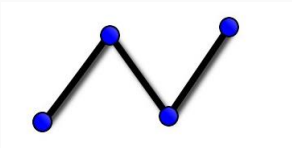
```
library(sp)
```

Locations for stations – Spatial Points



```
#first read in the data
metro_stops <- read.csv("~/stops.txt")
> names(metro_stops)
[1] "stop_id"   "stop_name" "stop_lat"  "stop_lon"

> stations<- SpatialPoints(metro_stops[3:4], proj4string = CRS("+proj=longlat
+ellps=WGS84"))
```

Paths – like a collection of points joined in an order (by sequence id)



```
metro_shapes <- read.csv("~/shapes.txt")
> names(metro_shapes)
[1] "shape_id"    "shape_pt_lat"     "shape_pt_lon"   "shape_pt_sequence"
"shape_dist_traveled"
```

We need to read the points and sequence info in shapes.txt into a SpatialLines object

- This is a bit trickier!

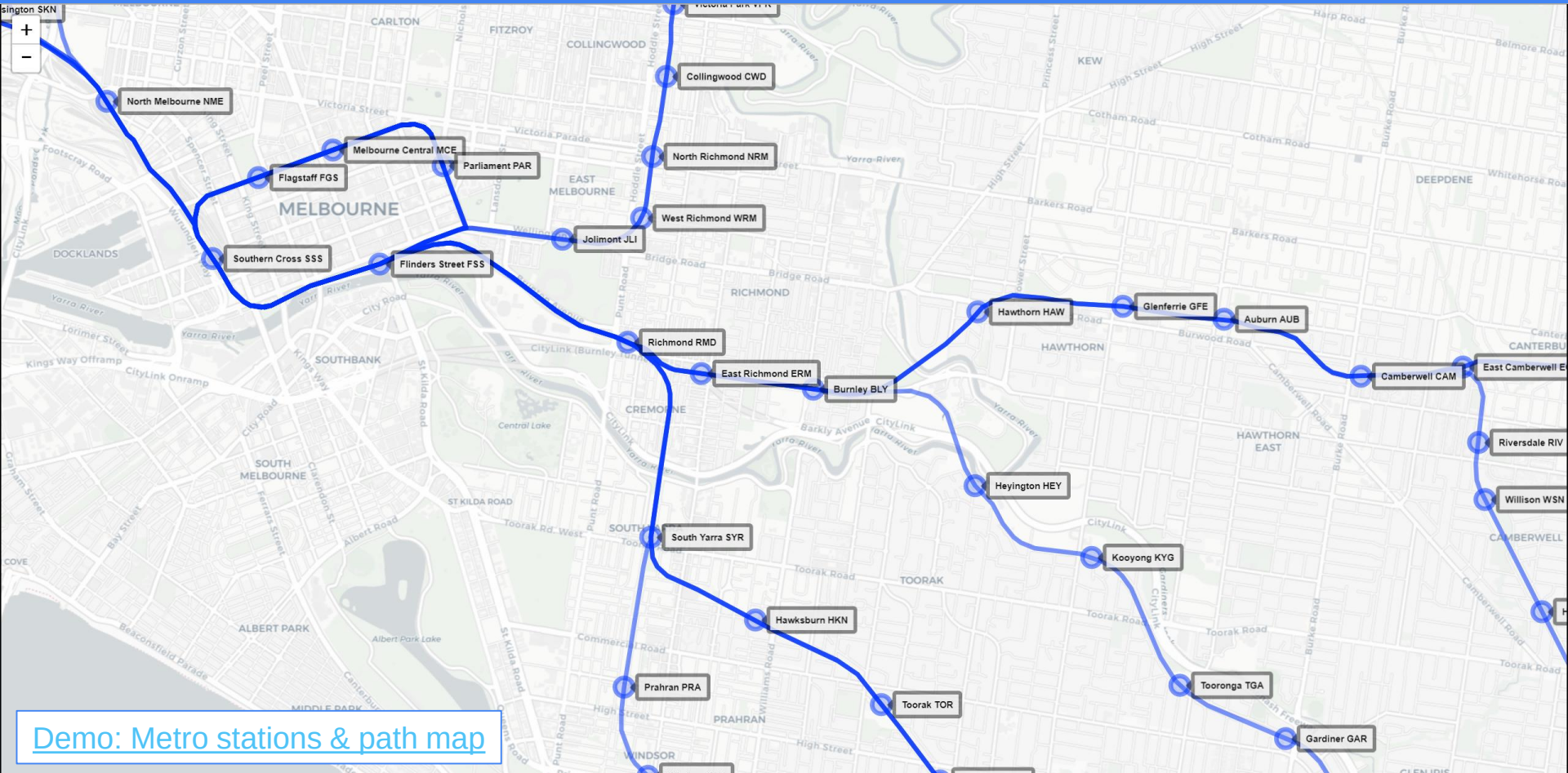
Spatial lines: points_to_line

```
library(sp)
library(maptools)
library(rgeos)

metro_lines <- points_to_line(data = metro_shapes_long,
                              long = "shape_pt_lon",
                              lat = "shape_pt_lat",
                              id_field = "shape_id",
                              sort_field = "shape_pt_sequence")

> class(metro_lines)
[1] "SpatialLines"
attr(,"package")
[1] "sp"
```

Points to line function:
https://rpubs.com/walkerke/points_to_line

Plotting with leaflet

The leaflet package



[Demo: Metro stations & path map]

leaflet syntax

```
leaflet() %>%

addTiles() %>%

addCircleMarkers(

data,lng,lat,label)%>%

addPolylines(SpatialLinesObject)
```

```
library(leaflet)
library(magrittr)

leaflet() %>%
  addTiles() %>%
  addCircleMarkers(data=metro_stops,
    lng=~stop_lon,
    lat=~stop_lat,
    label=~stop_name,
    labelOptions=labelOptions(noHide=TRUE))%>%
  addPolylines(data=metro_lines)
```

Create a handy empty map object

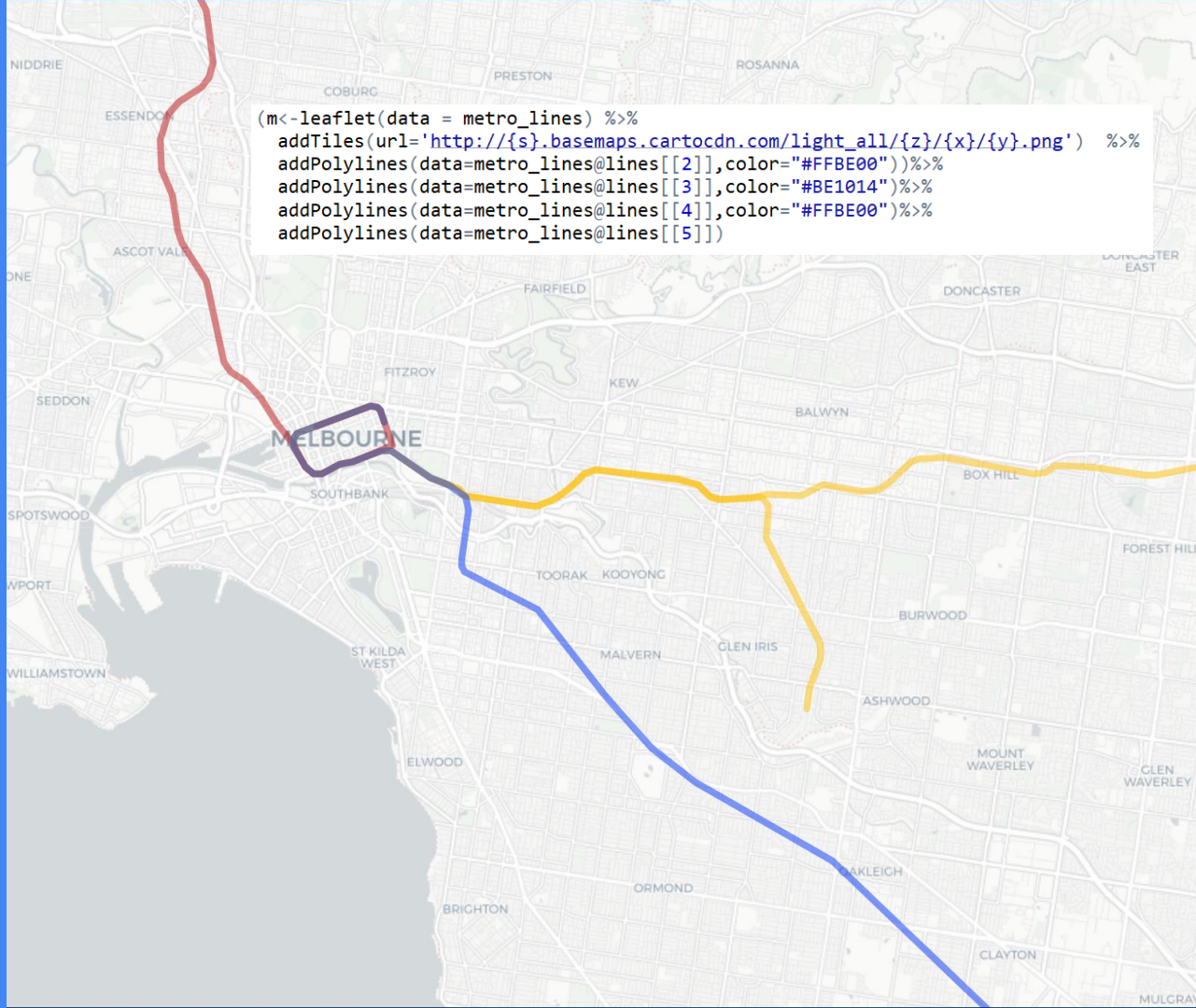
```
e.map<-(leaflet() %>%
addTiles(url='http://{s}.basemaps.cartocdn.com
/light_all/{z}/{x}/{y}.png’)


#which you can then add a marker layer to
(ALM.map<-
e.map%>%addCircleMarkers(data=ALM_stops,
                 lng=~stop_lon,
                 lat=~stop_lat,
                 label=~stop_name,
labelOptions =labelOptions(noHide=TRUE))

#And then add more layers to ALM.map
```

Multicoloured lines - first try

leaflet ()%>%

addTiles()%>%

addPolylines(data, color)



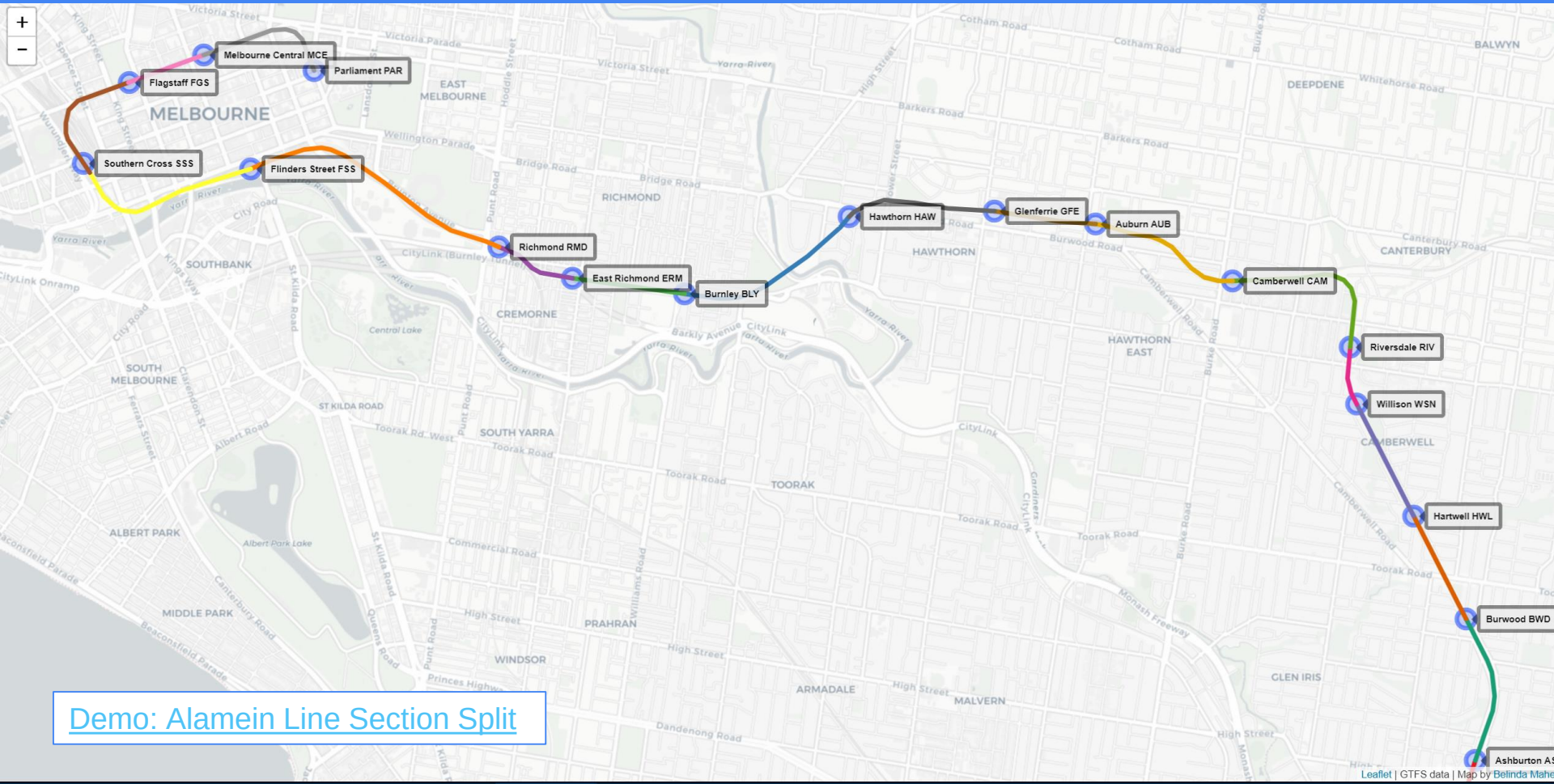
Adding each line section with a different colour

Accessing individual lines using indexing on a SpatialLines object:

SpatialLinesObject@lines[[*line number*]]

```
#can use RColorBrewer to get a list of colours
#Add coloured lines

for(i in 1:length(ALM_lines)){
ALM.map<-(
ALM.map%>%addPolylines(data=ALM_lines@lines[[i]],
color=colours[i],opacity=1)
          )
}
```

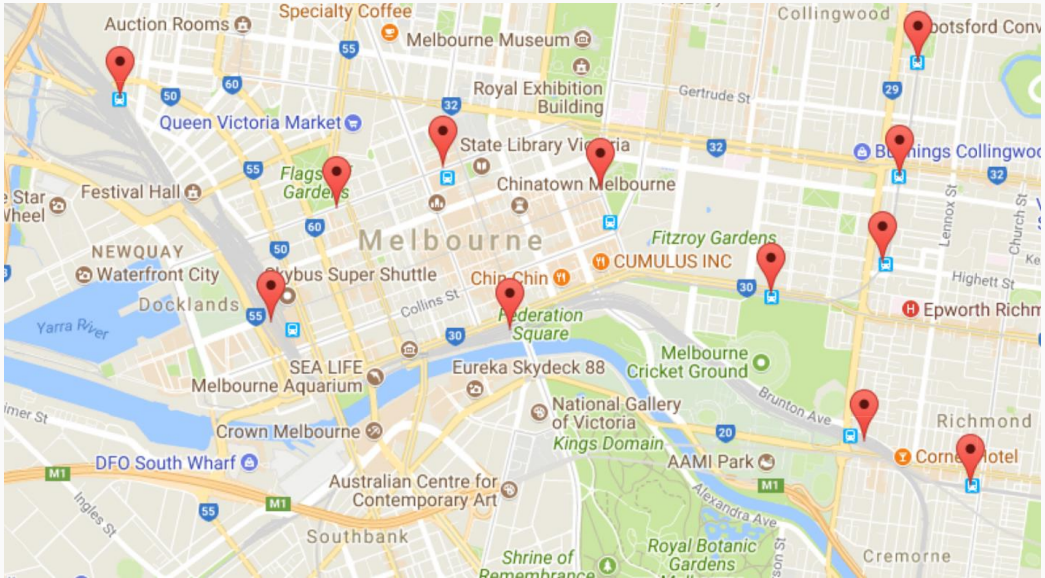
Alamein Line Section Split



[Demo: Alamein Line Section Split]

library(googleway)


```
library(googleway)
map_key <- "your api key"

google_map(key = map_key, data = metro_stops) %>%
 add_markers(lat = "stop_lat", lon = "stop_lon",
info_window = "stop_name")
```



Further Reading

Other packages to look in to:
Googleway – developed by Symbolix in Melbourne
http://demos.symbolix.com.au/GooglewayRMeetupPresentation.html#/

https://cran.r-project.org/web/packages/googleway/vignettes/googleway-vignette.html

Points to line function which I used in my code
https://rpubs.com/walkerke/points_to_line

Intro to Spatial Data in R
http://spatial.ly/wp-content/uploads/2013/12/intro-spatial-rl-3.pdf

Thanks for coming!

Feel free to ask me questions.

belinda.maher@gmail.com

Twitter: @mingabelle

about.me/minga

Photo credit: twitter: @ptv_official

Data files in GTFS

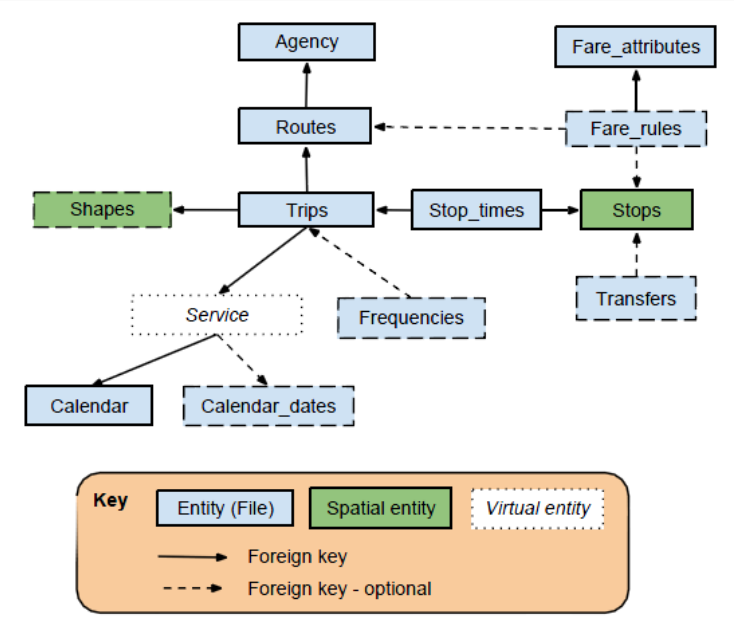
stops.txt *(points)*
stop_id, stop_name, stop_lat, stop_lon

shapes.txt *(polylines)*
shape_id, shape_pt_lat,shape_pt_lon, shape_pt_sequence, shape_dist_traveled

routes.txt *(route names)*
route_id, agency_id, route_short_name, route_long_name, route_type

trips.txt
route_id, service_id, trip_id, shape_id, trip_headsign, direction_id

stop_times.txt
trip_id, arrival_time, departure_time, stop_id, stop_sequence, stop_headsign, pickup_type, drop_off_type, shape_dist_travelled



Folders & Files in GTFS

http://data.ptv.vic.gov.au/downloads/PTVGTFSReleaseNotes.pdf

1 - Regional Train	V/Line
2 - Metropolitan Train	Metro
3 - Metropolitan Tram	Yarra
4 – Metropolitan Bus	
5 - Regional Coach	
6 - Regional Bus	
7 - TeleBus	
8 – Night Bus	
10 - Interstate	
11 - SkyBus