A Data Analysis of Bus Journey Variability in Dublin

Final Year Project – Owen Duffy

Introduction

- My FYP is an analysis of GPS data points from Dublin Buses over a three-week period in November 2012.
- Aims: The aim of this data analysis is to transform the raw data into some form of metrics from which we can draw any conclusions regarding the variability and performance of Buses in the Dublin Area
- This project appealed to me as I have an interest in Urban planning and traffic management. Additionally the unreliability of Dublin Buses had an affect in me electing to travel by DART to College for the last 4 years.

Tools and Technologies used

- Python
  - Pandas: Pandas is a ubiquitous tool for data processing tasks with Python. It facilitates easy reading and writing of CSV files and its DataFrame object type is perfect for manipulating data in these formats.
  - Plotly: Plotly is a JavaScript library for creating graphs and plotting data. It also has a Python library which I made use of. All the plots and maps I created were made using this library.
  - Jupyter Notebooks: Jupyter notebooks are perfect for trying to visualise data when acclimatizing to it. By using the cells system I could see the data I was working on and make adjustments to it without having to run the whole script again.
  - JavaScript: I used a JS API called ‘dublinbus-client’ to obtain accurate GPS coordinates of bus stops for plotting purposes.

```
data = pd.read_csv("../Datasets/siri.20121106.csv", names = ['Timestamp','LineID','Direction','JourneyPatternID','TimeFrame','VehicleJourneyID',
```

```
display(HTML(data.head(10).to_html()))
```

	Timestamp	LineID	Direction	JourneyPatternID	TimeFrame	VehicleJourneyID	Operator	Congestion	Long	Lat	Delay	BlockID	VehicleID	StopID	AtStop
0	1352160000000000	15	0	00150001	2012-11-05	5826	RD	0	-6.258584	53.340099	-361	15013	33210	4870.0	0
1	1352160000000000	46	0	046A1002	2012-11-05	7267	D2	0	-6.259093	53.345425	-1101	46004	36024	794.0	0
2	1352160000000000	14	0	00140001	2012-11-05	6206	D2	0	-6.257329	53.287521	-126	14003	33325	1047.0	0
3	1352160002000000	41	0	041B0002	2012-11-05	61	SL	0	-6.264167	53.453217	-623	41008	33631	3874.0	1
4	1352160002000000	63	0	NaN	2012-11-05	1116	D2	0	-6.171050	53.259201	292	63003	33137	3283.0	0
5	1352160002000000	39	0	039A1002	2012-11-05	3795	PO	0	-6.262447	53.346767	-532	39026	36060	1479.0	0
6	1352160002000000	65	0	00650001	2012-11-05	4004	RD	0	-6.594641	53.129776	-287	65003	38004	7283.0	0
7	1352160002000000	40	0	040D1001	2012-11-05	2466	HN	0	-6.258850	53.362499	-488	40207	33274	52.0	0
8	1352160002000000	4	0	NaN	2012-11-05	5076	HN	0	-6.261073	53.352112	0	4003	43035	4725.0	0
9	1352160002000000	11	0	00111002	2012-11-05	5241	D1	0	-6.230217	53.323002	-536	11001	33462	320.0	0

Dataset

Dataset columns

- Timestamp micro since 1970 01 01 00:00:00 GMT,
- Line ID,
- Direction,
- Journey Pattern ID,
- Time Frame (The start date of the production time table - in Dublin the production time table starts at 6am and ends at 3am),
- Vehicle Journey ID (A given run on the journey pattern)
- Operator (Bus operator, not the driver),
- Congestion [0=no,1=yes],
- Lon WGS84,
- Lat WGS84,
- Delay (seconds, negative if bus is ahead of schedule),
- Block ID (a section ID of the journey pattern),
- Vehicle ID,
- Stop ID,
- At Stop [0=no,1=yes]

Thanks to Michael Cullen for providing me with this key.

Danish Paper

- Travel time variability: Definition and valuation
  - Fosgerau, Mogens and Hjorth, Katrine and Brems, Camilla and Fukuda, Daisuke
  - 01/2008
- This paper argues that travel time variability needs to play a bigger part in discussions regarding Transport Policies as data gathering is currently focused on Journey time alone while the variability of journeys has economic effects

Points of Note in Danish Paper

- “the standard deviation is comparatively simple to measure and predict. It is hard to conceive of a simpler and more straight-forward measure of travel time variability. It is hence the easiest measure to compute from traffic models.”
- “travel times also become more variable and unpredictable as congestion increases. From the point of view of the traveller, it becomes hard to predict for instance how long the commute to work will take. This uncertainty entails additional costs to travellers and hence to society. It is relevant and necessary to include these costs in the economic evaluations of transport policies, especially those policies that are directed against reduction of travel time variability ”

Initial attempts and Issues with dataset

- Initially I was hopeful that I could use the VehicleJourneyID column to segregate the data into separate journeys, unfortunately it turns out this Value is not unique to a journey as I found out that multiple vehicles on multiple routes could use the same ID simultaneously. As such I had to isolate each vehicle in the dataset and then extract every journey it did.
- Additionally the dataset was not perfect. The data wasn't always recorded perfectly, some journeys weren't recorded in their entirety, some were truncated and only partially recorded.
- As such I decided that I'd be better served by analysing stretches of routes (i.e. between stop pairs) rather than looking at journeys as a whole.

Data Processing Methodology

- Step 1: For each vehicle, for each journey get times between each pair of stops on route and add to output DataFrame creating a new Dataset of Interstop journeys
- Step 2: Using this new dataset aggregate all journeys between stop pairs (and later triples etc) finding the Mean journey time and then normalise all data points to become in relation to the mean rather than as pure timestamps.
- Step 3: Once Journey times had been normalised in this way get the Standard Deviation for each Interstop journey and use this figure to compare.
- Step 4: Visualise and Plot

Data Volume Issue and Processing Time

- The entire dataset for this project is 24 days worth of data. On one single day (6/11/12) there are 1,765,912 data points.
- For each vehicle, for each journey, for each stop pair.
- In a single threaded script it took approximately 90 minutes to process a single day through just one step.
- Not viable to take this long to process, volume of data too large.
- However because the datapoints were independent a prime candidate for multi-threading!

Multithreading

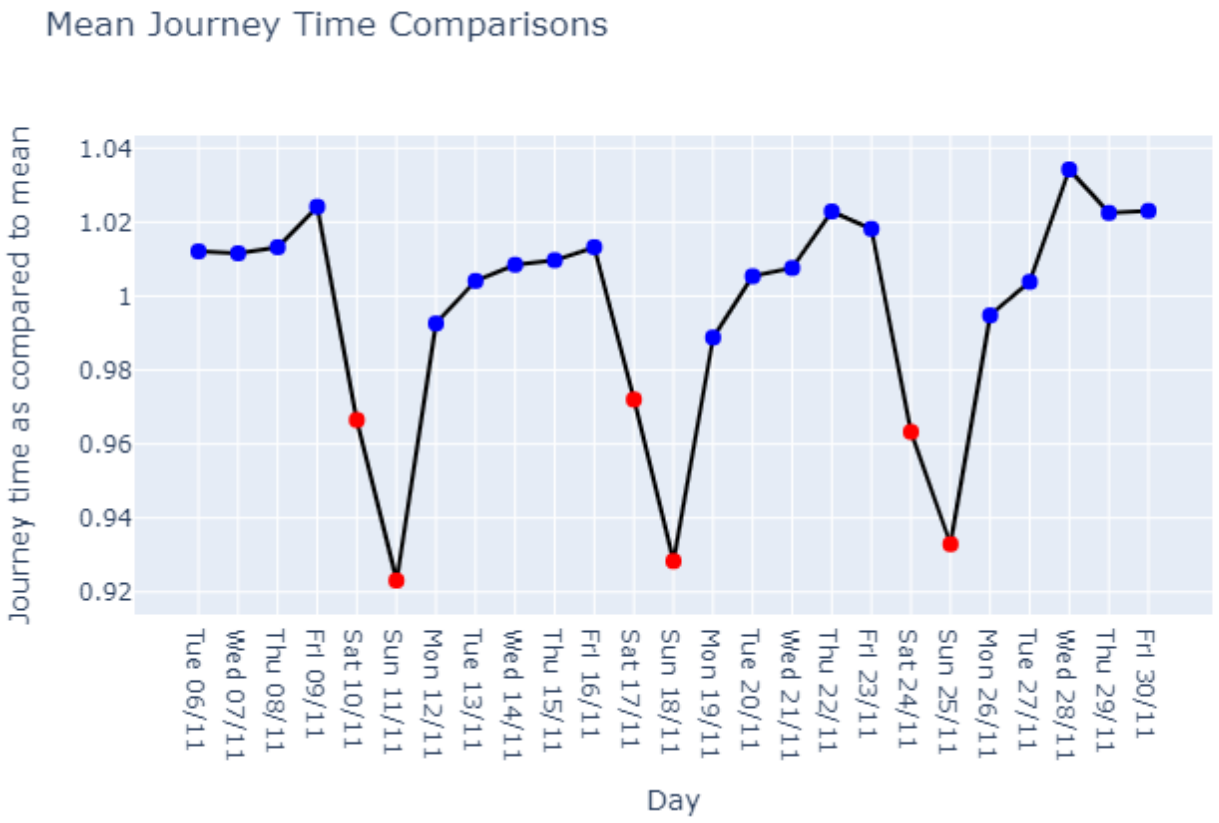
- On 6/11/12
  - 1,765,912 GPS pings
  - 835 unique vehicles
  - An example vehicle made 28 journeys
  - Approximately 30 stops per journey
- Breakdown the dataset by assigning subsets of Vehicles to different threads and then allowing them to proceed as normal and then amalgamating their outputs into one common output of all Interstop journeys.
- Modify scripts to be parallelisable and to accept command line arguments then call script on each file in dataset. With 50 threads running allowed me to process the entire dataset in the kind of time it took me to process one file before

GPU Acceleration and Cloud Computing

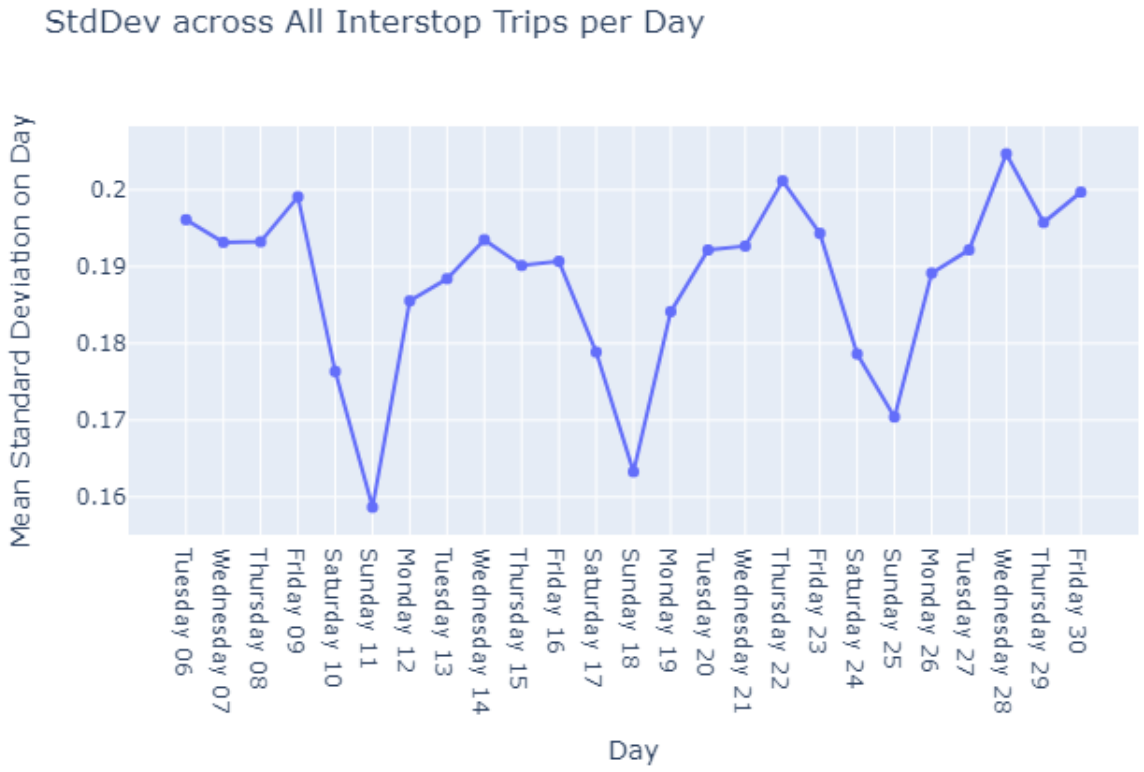
- GPU Acceleration:
  - Seeking extra speed I investigated into GPU acceleration.
  - This is very possible with Python as it's often used in ML done through special Libraries (RAPIDS cuDF is Pandas equivalent). For my purposes when I only needed to process the data once the time investment to use GPU was not worth it.
- Cloud Computing:
  - Possible to use AWS, Google Cloud, Azure etc to process data
  - Given the fact data was processed in ~90mins unnecessary hassle.

Means Compared across days

- Gather all data points for all days together
- Compare all examples of Interstop journeys and express each in reference to the mean for that Interstop Journey across sample.
- Compare the means for all Journeys on each day by getting the mean of the means
- Plot using line Graph to show changes across days

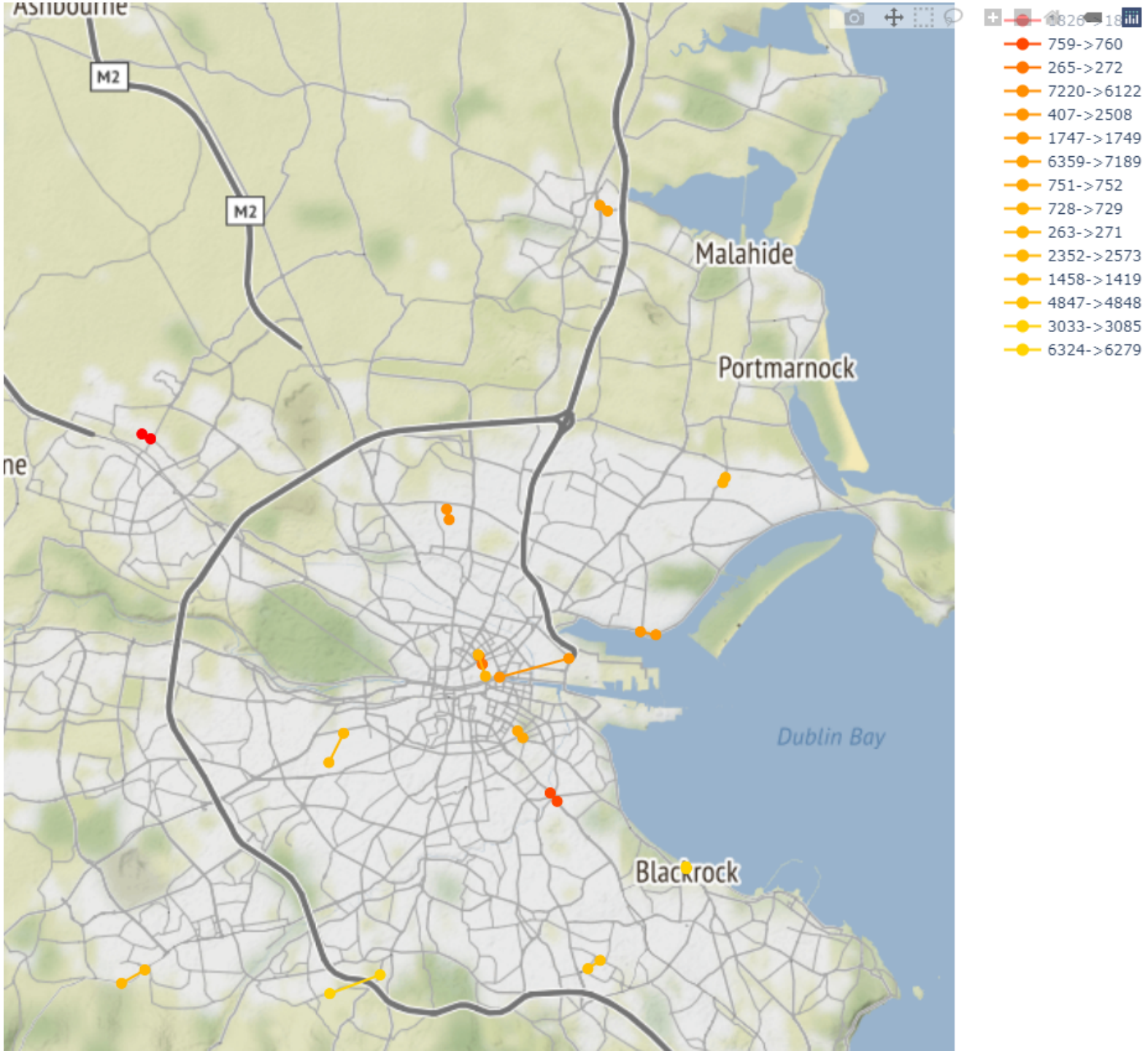


Std Dev across days



- I already have the variability of each Interstop Journey on each day
- Read the Variabilities for each day, plot the Mean for the day.
- Caveats, The data contains a lot of examples where a particular journey occurred very few times so the Std. Deviation for that stretch might not be an accurate representation, this has brought down the mean significantly skewing the data.

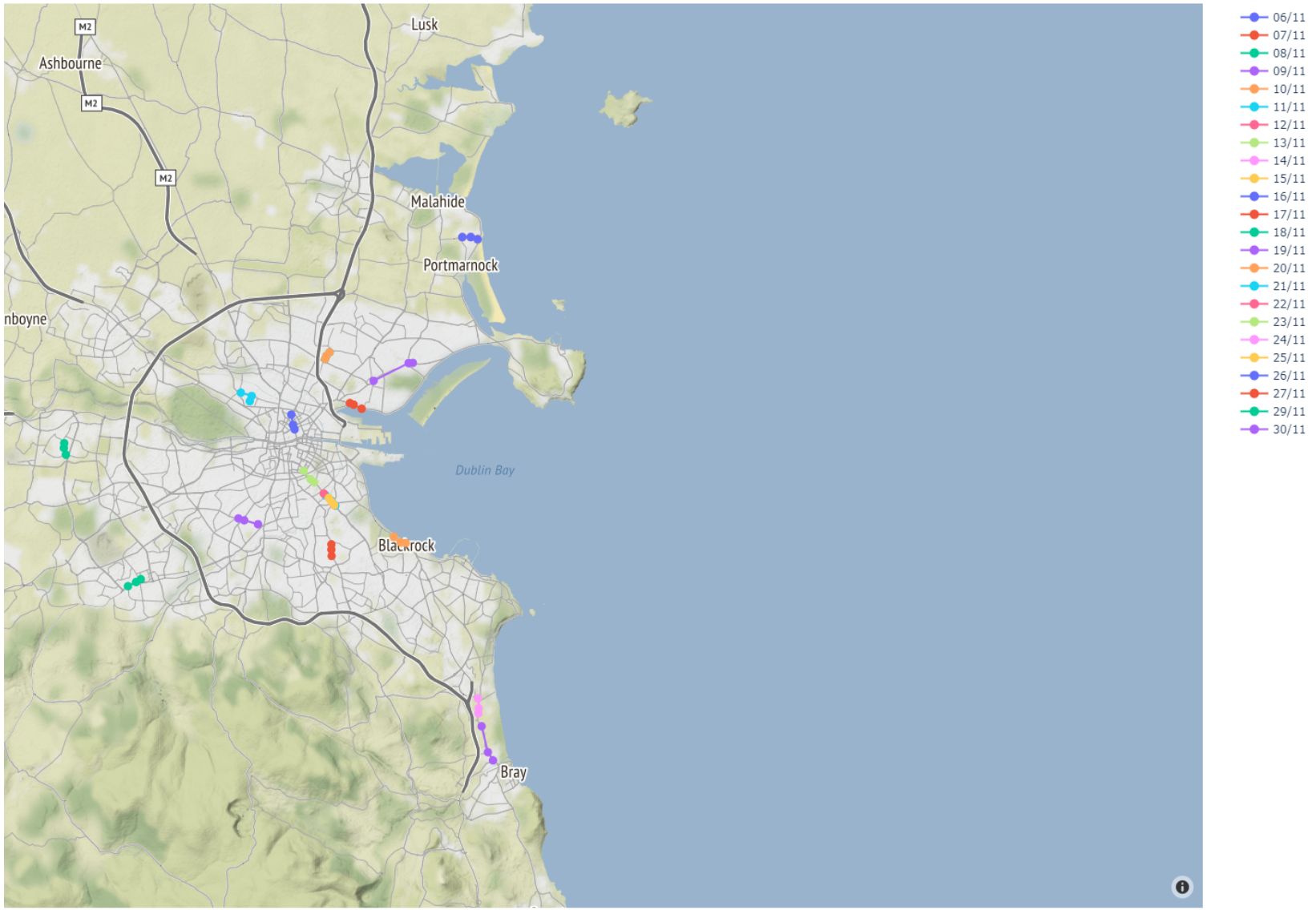
Most Variable Routes by Day 11/11/2012



Most Variable Routes by Day

- In this figure I plot the Most Variable stretches on a particular day.
- In this case I've elected to plot only the top 10, however, I can choose any number of stretches with the limiting factor only being how well the viewer can distinguish them.
- The lines are coloured according to a gradient based on how variable they are.
- The gradient can be over the whole dataset for the day or just the lines to be plotted.
- The code can produce a plot for each day and a human can compare the plots to see if they spot patterns.

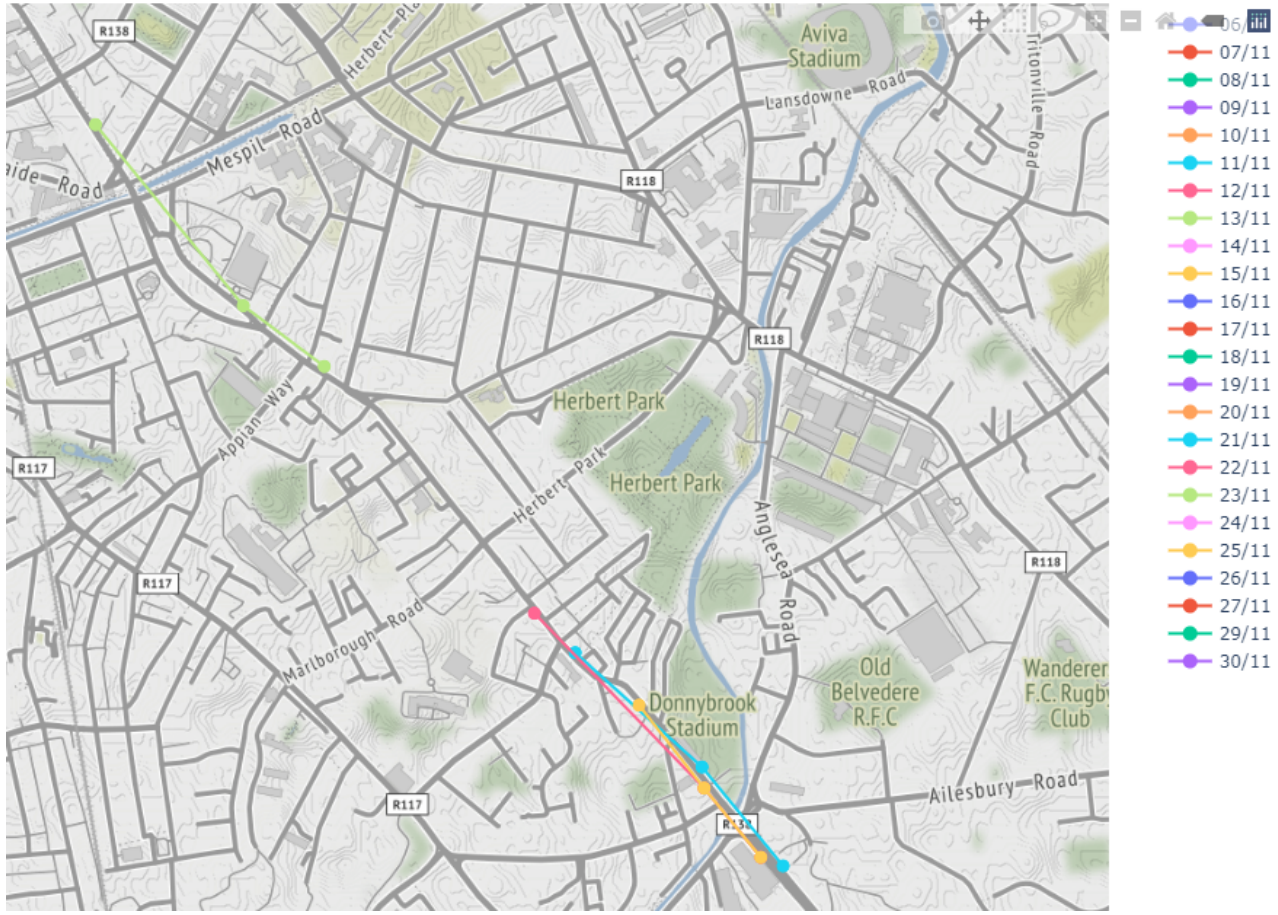
Most Variable Stretch per day



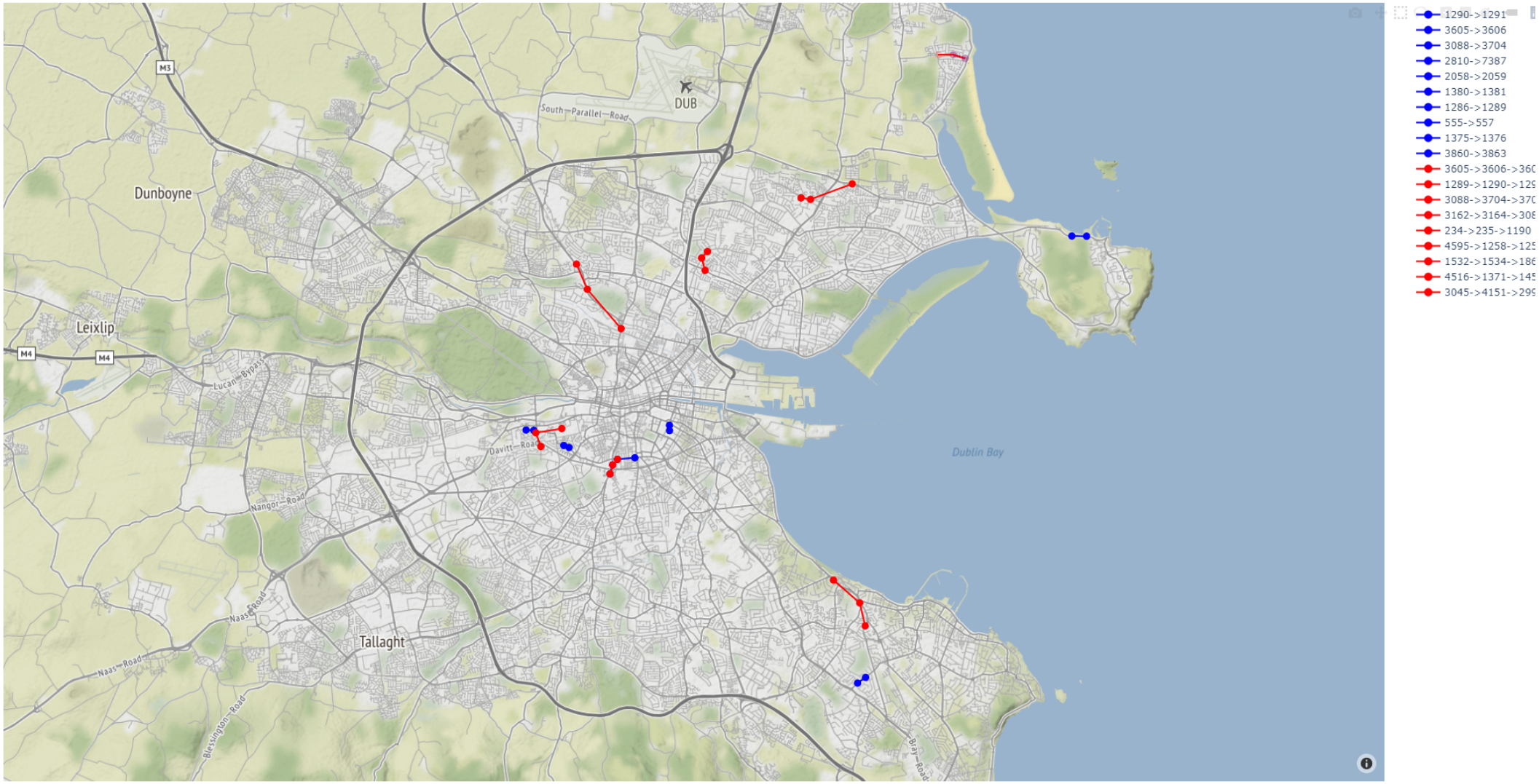
Most Variable Stretch per day

- For this plot I read all the processed days and took the most variant stretch on that day and plotted it.
- The hopes for this is to see if any pattern emerges where stretches occur repeatedly thus identifying them as problem areas.
- In this instance I did see an obvious occurrence of such, the stretch of the Stillorgan road around Donnybrook bus garage and Donnybrook stadium is the absolute most variable section on several days demonstrating that it is a consistent problem area.

Analysis identifying Donnybrook as Problem Area



The Most Variable Pairs Vs Triples



The Most Variable Pairs Vs Triples

- The previous slide showed a map onto which was plotted the most variable pair journeys, and the most variable triple stop journeys.
- The aim of this was to see if there was overlap, did a pair of stops having high variability translate into a triple of which that pair was a subsection of also being highly variable.
- The previous figure does show that in some instances this is the case however there are issues with the visualisation such that lines are superimposed causing them to be obstructed.
- Going forward this could be solved by scaling down the line widths as they’re drawn so that lines which are drawn over can still be seen.

Example of Overlapping High Variance

Reflection on Process

- The progression of the code written show's unfamiliarity with the libraries used. Code written by the end of the project is structured better and uses the tools available to better effect by nature of better familiarity.
- Similarly if I begun again, I'd design my methods more conscious of GPU acceleration and more powerful language features from the outset whereas as it stands it would be an undertaking to refactor all the current system the benefits of which would not be significant enough.
- Because this is a research project I was focused on the viability and acting as a proof of concept, if I were to develop this as an actual tool I would be more conscious of UI/UX.

Going Forward

- Fix the overlapping lines visibility
- Currently only measures between stops which the bus actually stopped, this shortcoming doesn’t account for stops which the buses simply pass.
- Generalise the code for multistep journeys so that the stretch can be of any length not just currently 3
- Find a robust way of detecting a journey over the whole length of a route to see if conclusions can be drawn from connection between interstop stretches and overall route performance.
- Web-Interface to allow users to view the data in a simplified manner.

Conclusions and Usefulness

- Is the data I presented useful?
  - As seen in the mean times/variability across days and the example of Donnybrook being a problem area we can see that the data backs up our expectations.
  - It’s otherwise hard to say what kind of value this would present to decision makers, I’ve tried to present data in a human readable way it’s outside the scope of this project to draw any conclusions into how it might be used.
- The scripts I’ve written could be used at the end of each day, or week, to generate human readable data.
- Given a larger sample size more robust conclusions could be drawn.