Santander Cycle Hire Usage Statistics, 2016

(*The Secret Life of Boris Bikes)*

John Downing IS53048A Assignment 3

Research Question

This assignment aims to analyse the 2016 usage data for London's *Santander Cycles* bike-sharing scheme. The scheme, introduced in July 2010, provides self-service cycle hire for the purposes of making short-term[1], point-to-point[2], journeys. Previous visualisations, for example http://bikes.oobrien.com/london, have tended to focus on docking stations and the availability of bikes at each of these. This study intends to see what information can be gleamed from the raw journey data – in the first instance, descriptive statistics regarding e.g. journeys counts, times, and durations, but also what additional patterns can be detected in the data via opportunities for slicing and dicing.

Data

The data was found by browsing the available sources on the Transport For London (TFL) open data web site (https://tfl.gov.uk/info-for/open-data-users). The motivation for considering these sources was that TFL are known to encourage use of this data (as stated on their webpage) - as well as being sometime collaborators of the Data Science faculty at Goldsmiths. The Santander Cycle Hire Usage Data (http://cycling.data.tfl.gov.uk), specifically, was chosen, as it was felt that this offered the potential for a wide range of analyses and visualisations – including, as it does, elements of time and geographical location.

TFL provide this data in CSV format, broken down into a series of chunks[3] – presumably for manageability. At the time of the analysis, data was only available up to 22nd November 2016, however this still amounted to some 9.45 million observations. The data dictionary for these (although not documented as such by TFL) is shown below.

Field Name	Field Type	Notes
Rental Id	Integer	Unique ID for an observed journey. Ascending sequence. Primary Key
Duration	Integer	The journey duration. Measured in seconds, but always in whole minutes e.g. 60, 120, 180.

[1] The maximum permitted rental period is 1 day; above this TFL can impose a fine.
[2] In the sense that a rental period is a single occurrence of travel between two docking stations, although these might be the same station.
[3] From mid-May 2016, TFL appear to have adopted a "one file per week" strategy – whereas prior to that, the only pattern seems to be the file sizes (e.g. a single file will cover as many weeks as will fit into Excel).

Bike Id	Integer	Unique ID for a particular bike in the cycle-hire scheme.
End Date	Date	The date/time at which the journey ended. Format dd/MM/yyyy hh:mm,SSS.
EndStationId	Integer	ID of the docking station at which a journey began. Foreign key.
EndStation Name	String	Text description of the docking station at which the journey ended.
Start Date	Date	The date/time at which the journey started. Format dd/MM/yyyy hh:mm,SSS
StartStationId	Integer	ID of the docking station at which a journey ends. Foreign key.
StartStation Name	String	Text description of the address of the docking station at which the journey began.

Although these datasets contain the ID of the start and end docking stations, they do not contain their geo-coordinates – so these had to be retrieved as XML from a live feed[4] and were parsed into CSV format using XSLT (Appendix 1). Note that the live feed does not always contain information for all stations, as documented at dockmonitor.blogs.casa.ucl.ac.uk/category/status. Any that were missing were retrieved (where possible) from the dockmonitor.blogs archives, and manually entered. The format for the resulting docking stations data is show below.

Field Name	Field Type	Notes
id	Integer	Unique ID, Primary Key
name	String	Description of address
terminalName	Integer	Secondary unique ID
lat	Float	Latitude
long	Float	Longitude
installDate	Date	

[4] https://tfl.gov.uk/tfl/syndication/feeds/cycle-hire/livecyclehireupdates.xml. It's possible that a canonical reference is available for docking stations data, but if so then it's not obvious from where.

removeDate	Date	
installed	Boolean	Data about current state of docking station, not relevant to this analysis.
locked	Boolean	
temporary	Boolean	
nbBikes	Integer	
nbEmptyDocks	Integer	
nbDocks	Integer	

Initial Processing.

The first step in preparing the journey data was to merge the relevant datasets into a single file. For efficiency, this was done by just catting together the files covering 25 December 2015 to 22 November 2016 together (Appendix 2), and manually removing any entries from 2015[5]. The resulting file was then read into a Pandas DataFrame (Appendix 3) with shape (9454907, 6). Only numerical rows were kept, due to the memory footprint and slow processing speeds involved with python object types. `EndStation Name` and `StartStation Name` were dropped, and `Start Date` was converted into an epoch-style timestamp. `End Date` was also dropped, since this could be inferred from `Start Date` and `Duration`[6]. Variable names were also updated to remove spaces e.g. `StartStation Name` became `StartStationName`.

One thing which became evident at this stage was that one of the files – for the period 31st August to 6th September – had a different format to the others. It had an extra, unused, column, and rather than reference docking stations by `id` it instead referenced them by "logical terminal" – which transpired to be values in the `terminalName` column in the stations data. This file was manually corrected in Excel, using VLOOKUP to replace the foreign keys to terminal names with their corresponding `id` values. In some cases this was not possible, since no docking station matching either the logical terminal or station names in the journeys data could be found. These were left in the dataset. The unused column was dropped.

In addition, it became apparent – as a result of trying to read `EndStationId` and `Duration` into python `int` values – that values for these two fields were often missing.

[5] An attempt was made to read_csv each file in turn, appending content to a DataFrame incrementally - however this proved to be very slow.

[6] In fact `Duration` seems to have just been derived – in the raw data – from the difference between `Start Date` and `End Date`. I found this out during further cleaning of the data; if `Duration` was wrong, `End Date` was wrong in exactly the right way.

The docking stations data was then read into its own DataFrame, and (left) joined with the journeys data so that each observation had geo-coordinates for the start and – where possible - end of its journey. With the core data in place, the next stage was to derive additional features to enable a wider range of analyses. Most obvious was `Distance`, based on start & end latitude & longitude. This was calculated as the "as the crow flies" distance, using the haversine formula (http://www.movable-type.co.uk/scripts/latlong.html). In theory, it should be possible to get a more accurate measurement - since the Google Maps API supports estimation of cycling distances between geo-coordinates in London[7]. However the daily usage caps for this API would have made making the relevant number of requests – for some 332,754 combinations of start and end docking station – problematic.

Values were also derived for `DayOfWeek`, `DayOfYear`, `MonthOfYear` and `HourOfDay` – and subsequently which `Season` (based on month of year) a journey was made in. An additional `JourneyType` category was also added, taking on numerical values for UNKNOWN, CIRCULAR and REGULAR – based on whether the end point of a journey was (a) known and (b) the same location as the start. A `Shared` category was also added, to indicate whether a journey had been made solo or as part of a group[8]. Finally, the ID and timestamp of the *next* journey for each bike was added to the current journey's row – to allow analysis of durations between journeys, per bike, and since TFL will often manually redistribute bikes to aid availability at busier docking stations. So the location at which a bike's journey ends will not necessarily be the location at which its next journey starts.

Initial descriptives flagged up an issue for the `Duration` column, in that some journeys had negative values. Manual checks in the raw data found these to be the result of DST changes, with the `Duration` values being simply the difference between `Start Date` and `End Date`. It was clear from visual inspection of journeys either side of a DST change that dates were in BST not UTC, and since a 10 minute journey which started at e.g. 01:59 on 30th October would have legitimately have finished at 01:09, this had resulted in negative durations. These were easy to correct, by adding 3600 to any negative values. The March 30th DST change (clocks going forward) were harder to spot, programmatically, however on visual inspection there were only 16 of these - and so corrections were made based on hard-coded `RentalId`s.

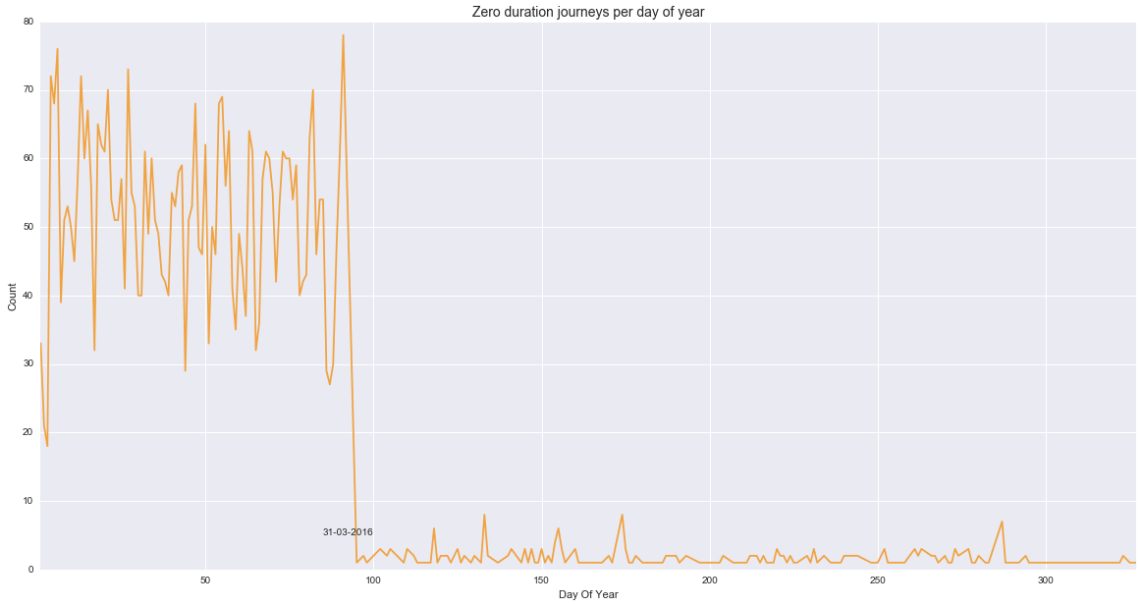
[7] Google Maps was used to sanity check calculations in this study e.g. distance, speed and duration of journeys.
[8] Journeys were grouped by start and end location, timestamp and duration – with the presumption that identical groupings indicated shared journeys. Some of these could have been co-incidental, however on the flip side the requirement for identical durations will have left out some *actual* shared journeys, which may have differed by seconds - so for the purposes of this study it has been assumed that these will average out.

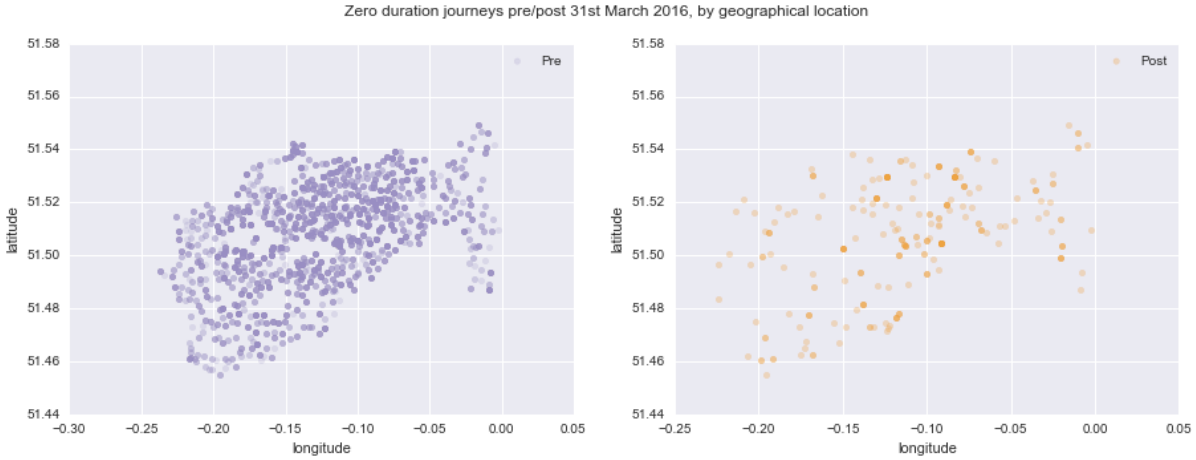
All of this proved to be quite CPU intensive, given the size of the data set, and so at this point the DataFrame was written out to a new file so that further analysis could proceed without having to repeat these steps.

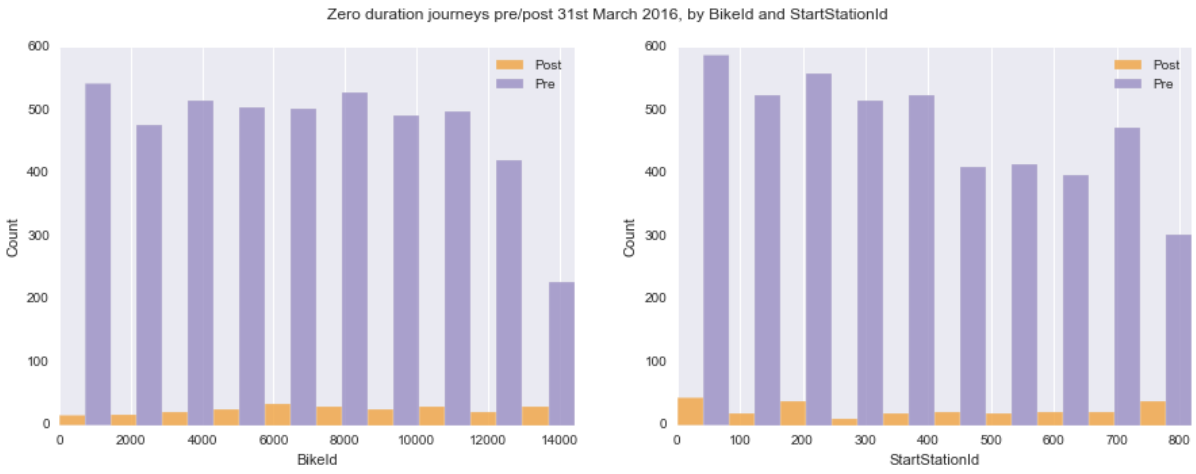
Further cleaning.

From analysis of the DataFrame, it was evident that further data cleaning was needed. Not in terms of the syntax – since the data was by this point well formed – but in terms of some of the `Duration` values being questionable. A number of observations had durations of zero seconds, with some additionally having a distance of zero (starting and stopping at the same location). Since journey duration was only recorded to the nearest whole minute, it is entirely possible that these journeys were either abandoned, due to the user's change of mind (or a faulty bike or docking station), or else were faulty observations. An interesting finding was that the number of such observations dropped off sharply after 31st March 2016 (figure below[9]). Rather than indicating a mass change in the behaviour of London's cycling population, this is perhaps more suggestive of a change in the way the data is captured e.g. the rollout of a new version of software[10]. Potentially it could have been due to an influx of new hardware (e.g. better bikes, or docking stations), however visualisations by `BikeId`, `StartStationId`[11] and geographical location, pre/post 31st March, do not support this. With no way of knowing how to impute duration for these 4962 observations, they were dropped from the dataset.

[9] FYI, Chart colours where applicable have been based on diverging colorbrewer palettes.
[10] TFL have thus far been unavailable for comment..
[11] Assuming that a higher ID in both cases represents a more recently introduced bike / docking station.

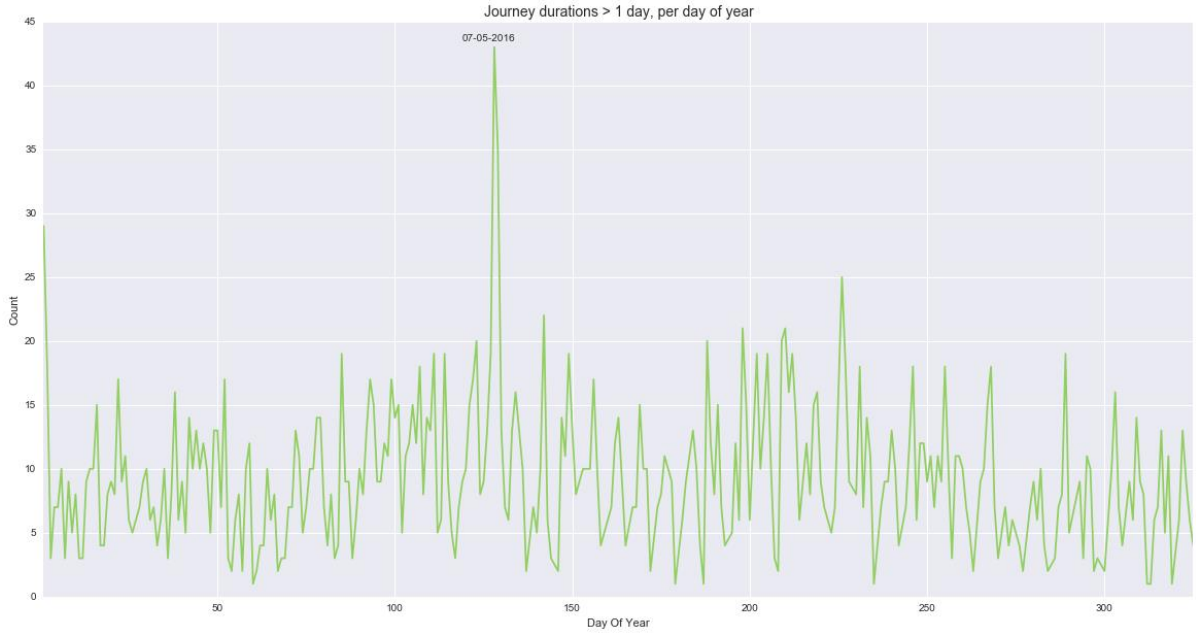

Zero duration journeys per day of year
Count
Day Of Year
31-03-2016


Zero duration journeys pre/post 31st March 2016, by geographical location
Pre
Post
latitude
longitude


Zero duration journeys pre/post 31st March 2016, by BikeId and StartStationId
Post
Pre
Count
BikeId
StartStationId

A remaining 166 journeys had zero durations, but different start and end locations. On the presumption that these were valid journeys, with (in effect) missing data, their duration values were recalculated – using the median journey duration for their specific distances.

Also dropped were 2850 journeys with duration greater than 1 day – since this is supposed to be the maximum permitted hire time. Whereas a duration of zero could be taken to be as a catch-all for "missing value", there is less reason to suspect that individual journey durations should be erroneous. It is of course possible that sometimes people hang onto bikes for longer than a day, and in addition some of these journeys may have been due to TFL taking the bikes overnight for maintenance[12] or redistribution. However these are anomalous patterns of usage – at least within the scope of this study.



Further issues with `Duration` became evident as a result of deriving a variable for the speed of each journey, based on the distance (in km) divided by the time (in seconds). Sorting the durations in descending order revealed that some bikes had, at face value, made journeys at speeds in excess of 300 km/h. For example, a journey on 11 October from Bankside Mix, Bankside, to Normand Park, West Kensington, had a recorded duration of 60 seconds - with Google Maps' estimation of the cycle route distance being 6.5 miles. Clearly both the duration and end date time

[12] A peek at the raw data behind the spike in the chart above (7 May 2016) revealed that a number of these journeys ended at locations ending with "mechanical workshop" - about which no further information was available.

values recorded for this observation are incorrect[13], which makes it less obvious whether to drop it entirely or try and impute more sensible values. Given that there were 11,033 observations where the speed was greater than three standard deviations, it was decided to keep these in the dataset and again impute duration with the median value for their specific distances. This of course assumes that the start and end docking station ids for these journeys are correct (or at least, more correct than their durations) - but this can arguably be justified on the basis that all of the data corrections thus far have been `Duration` related.

Following the recalculations of `Duration` (and subsequently `Speed`), the DataFrame was once more saved to disk. Subsequently, the data was treated as clean for the purpose of further analysis.

Analysis

The final DataFrame covered 9,447,095 journeys using the following variables:

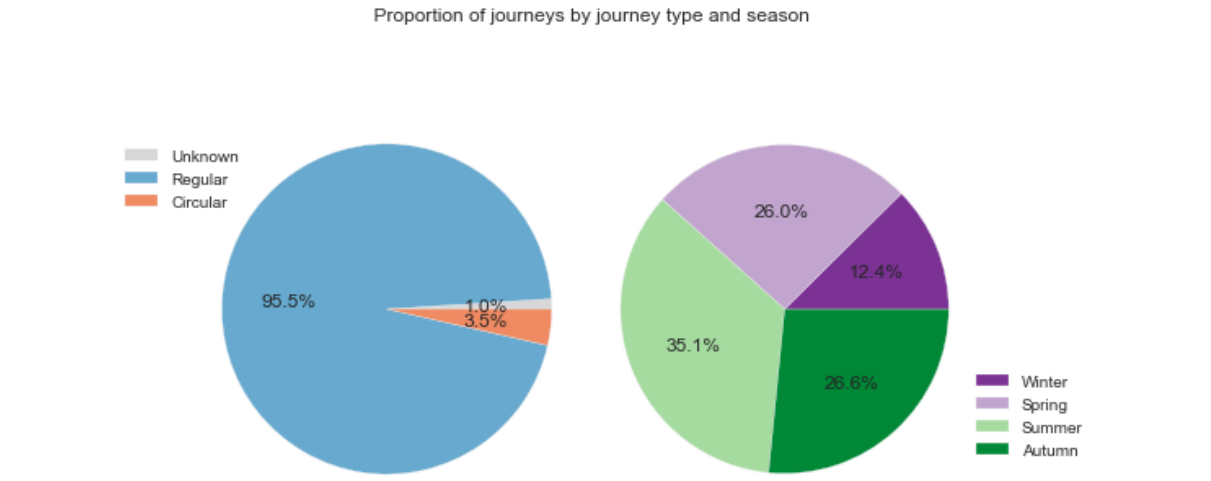
Variable Name	Python Type	Variable Name	Python Type
RentalId	int32	HourOfDay	int32
BikeId	int32	DayOfWeek	int32
StartStationId	int32	DayOfYear	int32
StartDateTime	int32	MonthOfYear	int32
EndStationId	float64	Season	int32
Duration	float64	StartLat	float64
Distance	float64	StartLong	float64
Speed	float64	EndLat	float64
JourneyType	int32	EndLong	float64
Shared	bool	NextStartStationId	float64
GapToNextJourney	float64	NextStartDateTime	float64

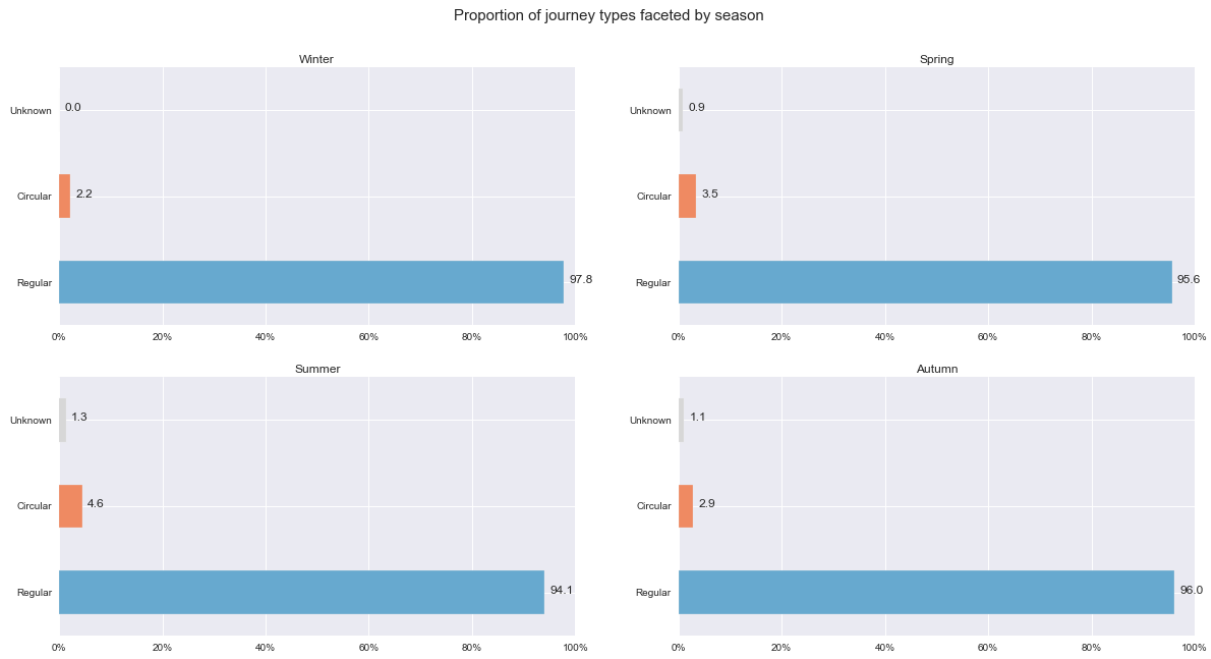
Of these journeys, 9,018,848 (95.5%) were regular journeys - having a known start and end location, where these are not the same - 333,972 (3.5%) were circular journeys - having a known start and end location, where these are the same - and 94,275 (1.0%) unknown journeys - where neither of the above could be determined due to missing location values.

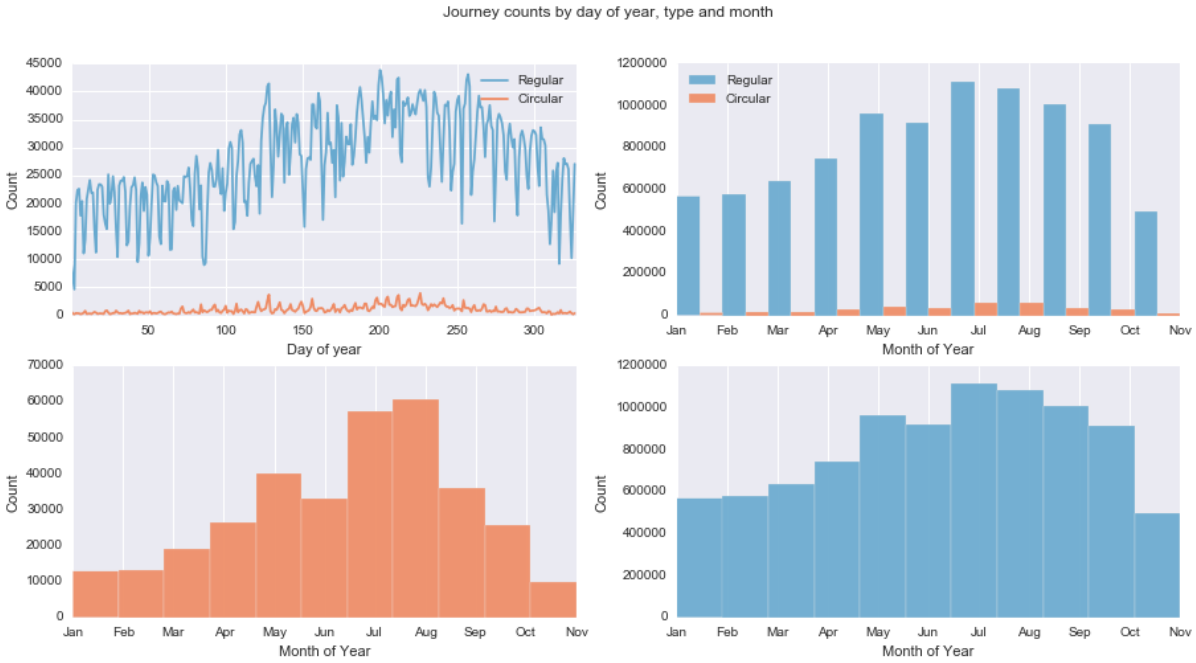
A journey was made on average every 2.99 seconds. The highest percentage of journeys were made in summer (35.1%), followed by autumn (26.6%), spring (26.0%) and winter (12.4%). The

[13] The raw data was checked to verify that this was not a parsing error or otherwise a processing miscalculation.

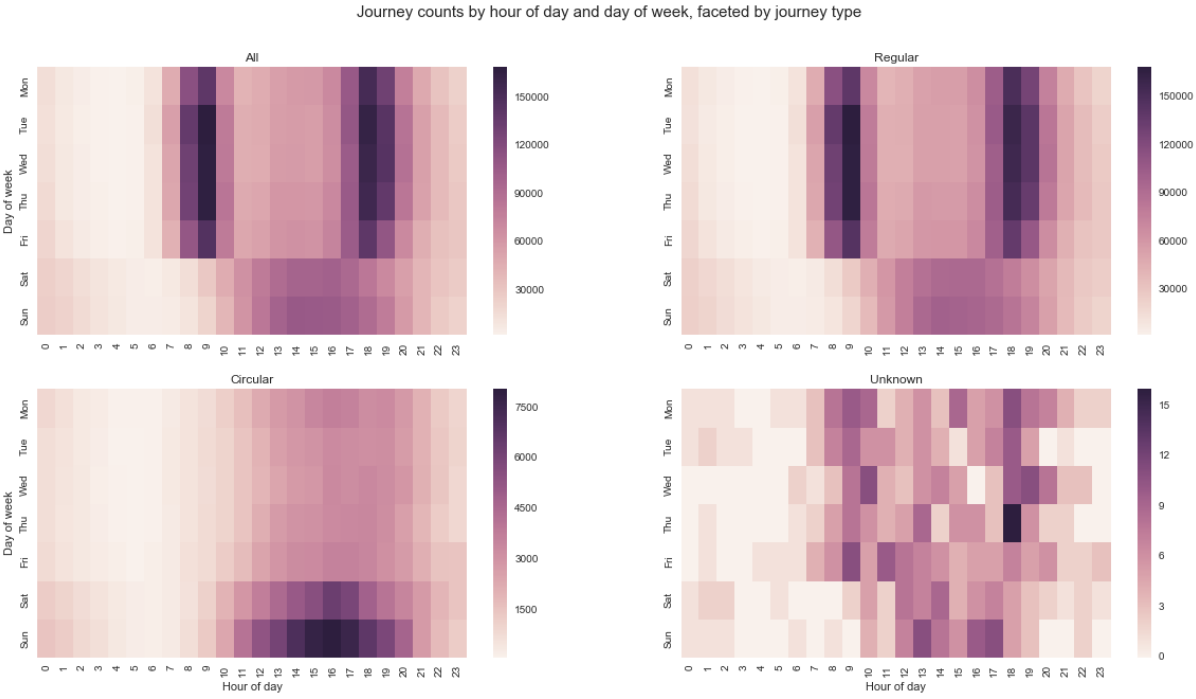
proportion of journey types per season did not vary greatly, with the proportion of circular journeys rising to 4.6% in summer and falling to 2.2% in winter. The distribution of circular journeys did however differ from regular journeys, with the latter displaying greater kurtosis.





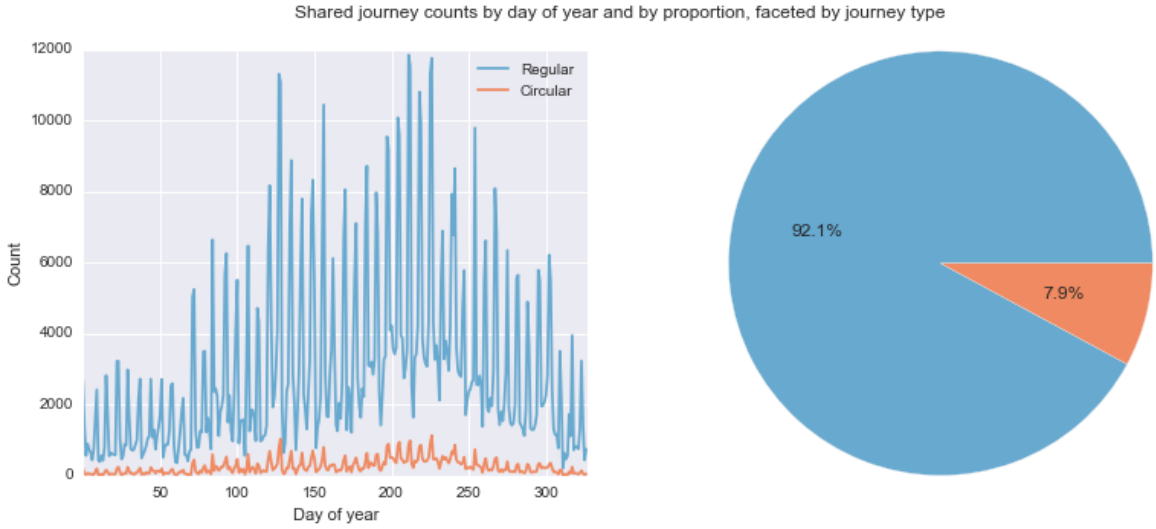


Regular journey counts showed a marked increase during the morning and evening commutes, and to a lesser extent on weekend afternoons. Circular journey counts conversely showed an increase on weekend afternoons, especially Sundays around 4pm.



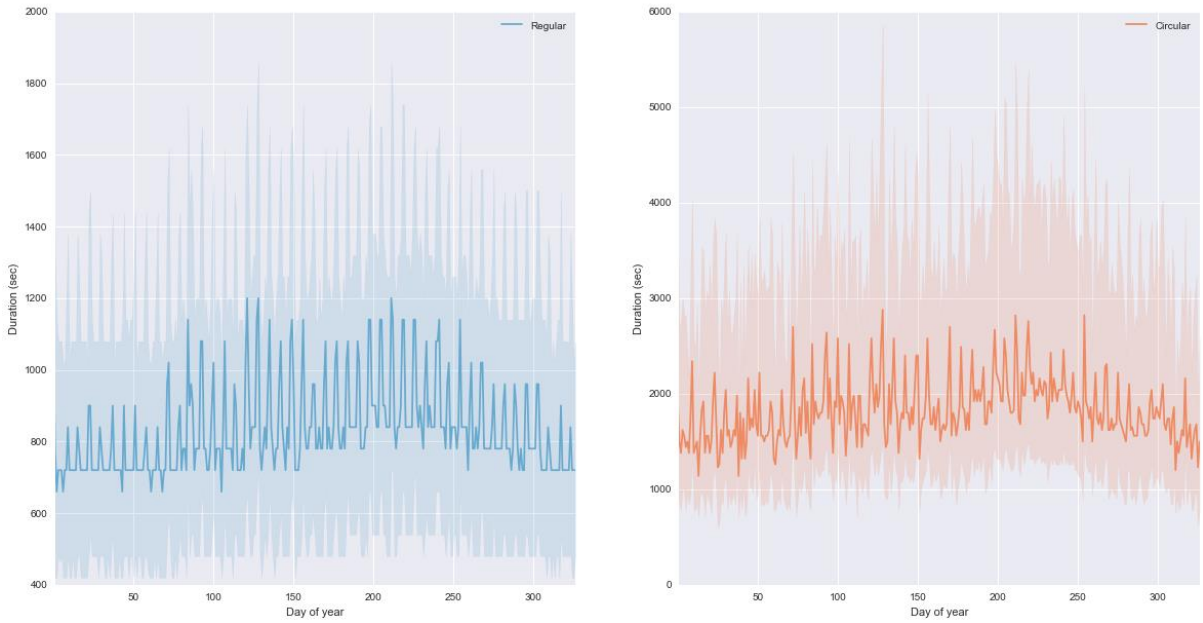
However there did not appear to be any obvious difference in geographical location between journey types – where it might have been expected that circular journeys would occur more around e.g. tourist hotspots or leisure locations.

Circular journeys made up a higher proportion of shared journeys (7.9%) than in the overall population. The number of shared, regular, journeys also spiked noticeably during the summer months.

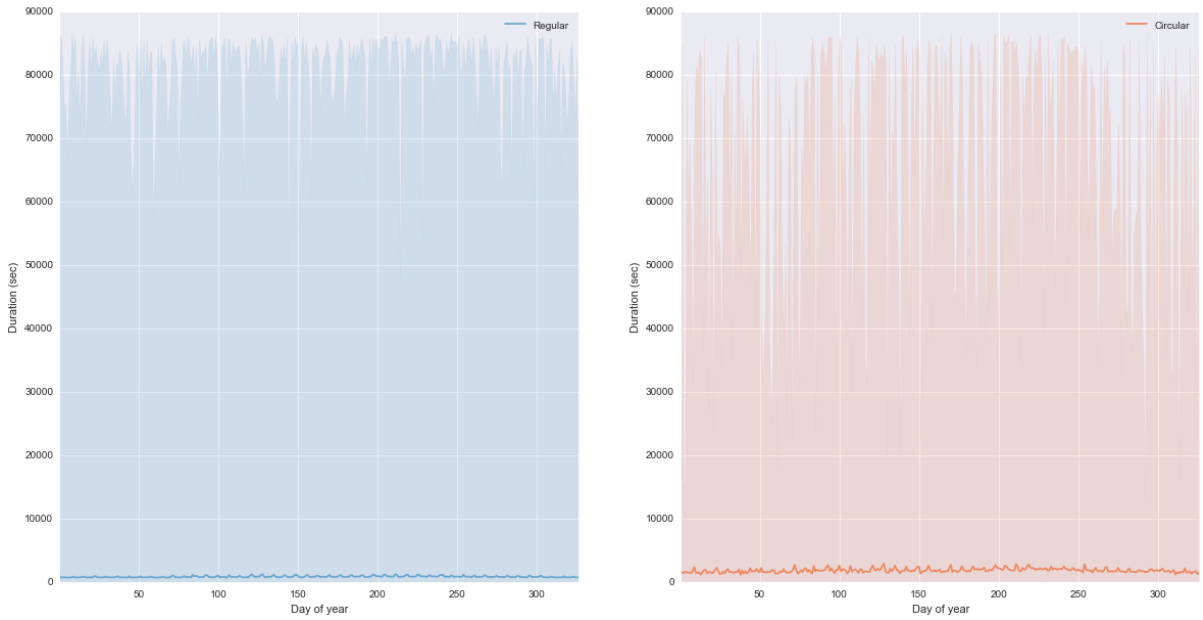


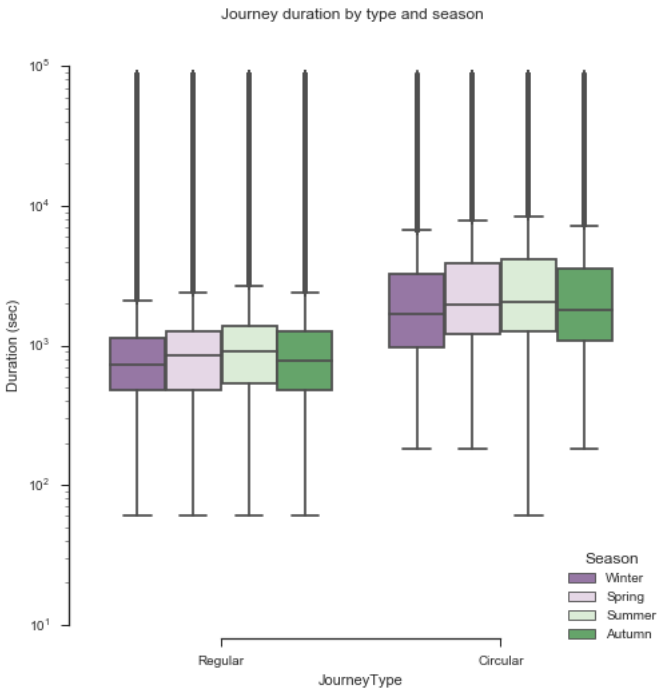
Journey durations were consistently higher for circular journeys (M = 55.6, SD = 84.6 minutes) than regular journeys (M = 20.2, SD = 37.2 minutes) – irrespective of season. A t-test confirmed a significant difference on this metric (t = -251.37, p < 0.001) between these two groups.

Median journey durations by day of year and journey type (with IQR)



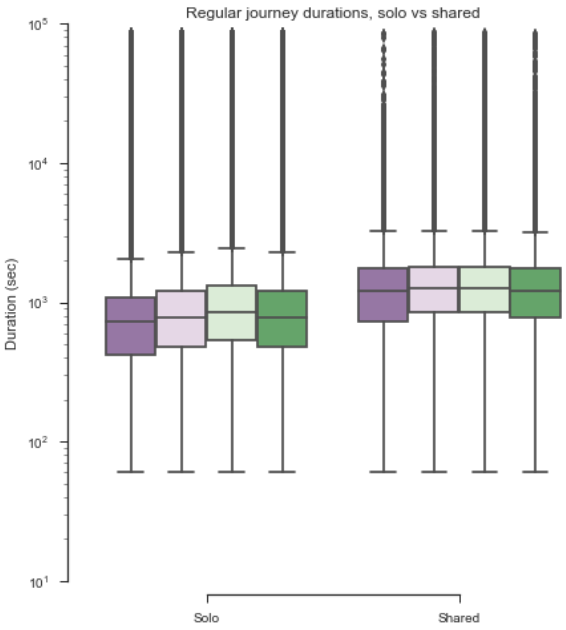
Median journey durations by day of year and journey type (with min/max range)

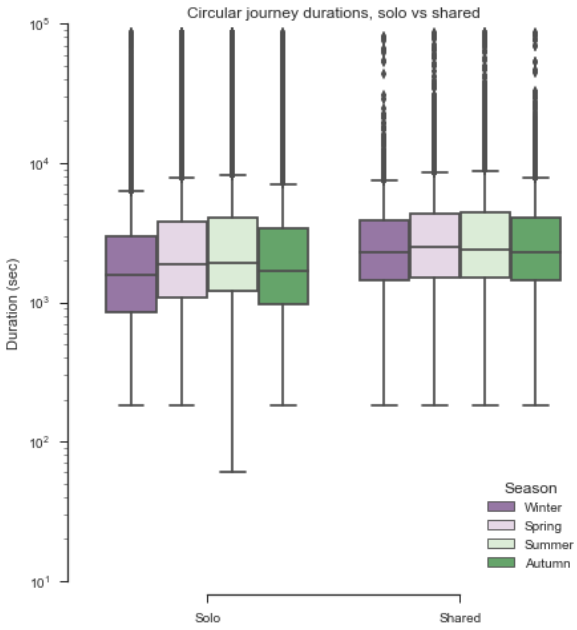




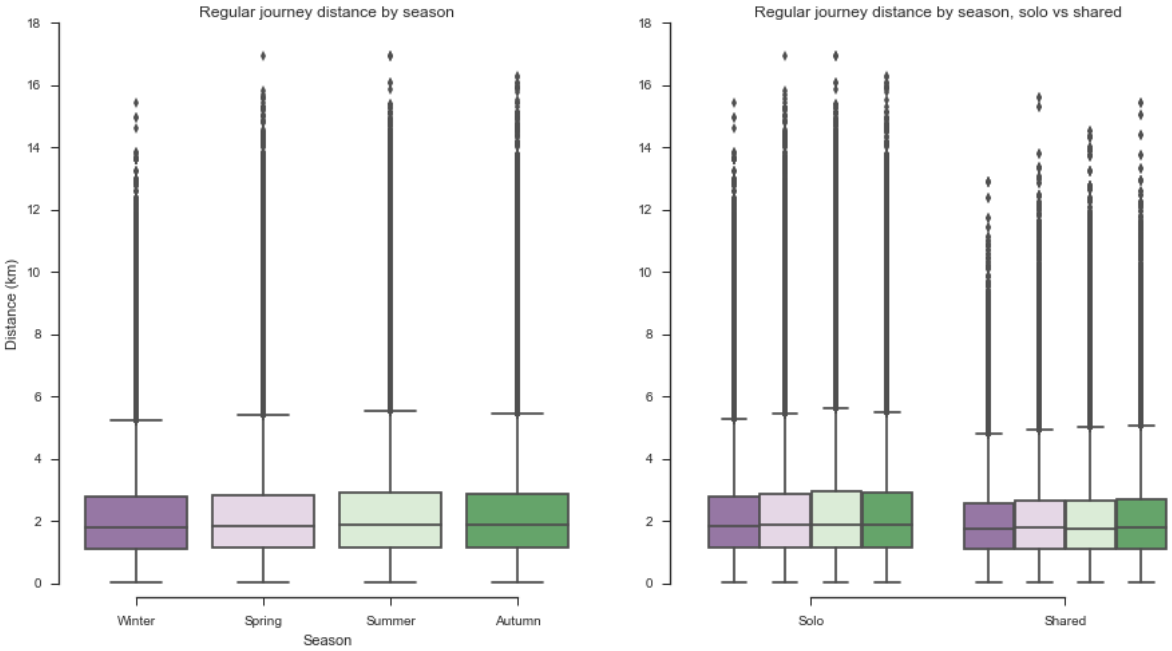
Journey duration varied more for circular journeys than for regular journeys (note the log scale on the figure to the left). However other than in summer, the IQR for regular journeys was not markedly different between seasons.

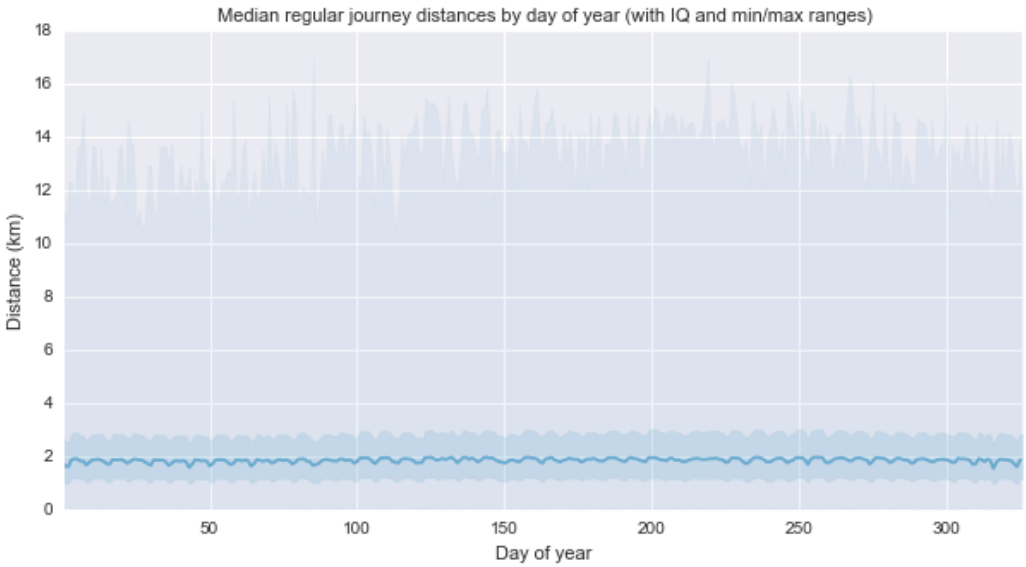
Journey duration was generally higher for shared journeys, however, the IQR for circular journey durations was comparable whether solo or shared. Regular journey durations and IQR, by comparison, were consistently higher for shared journeys.



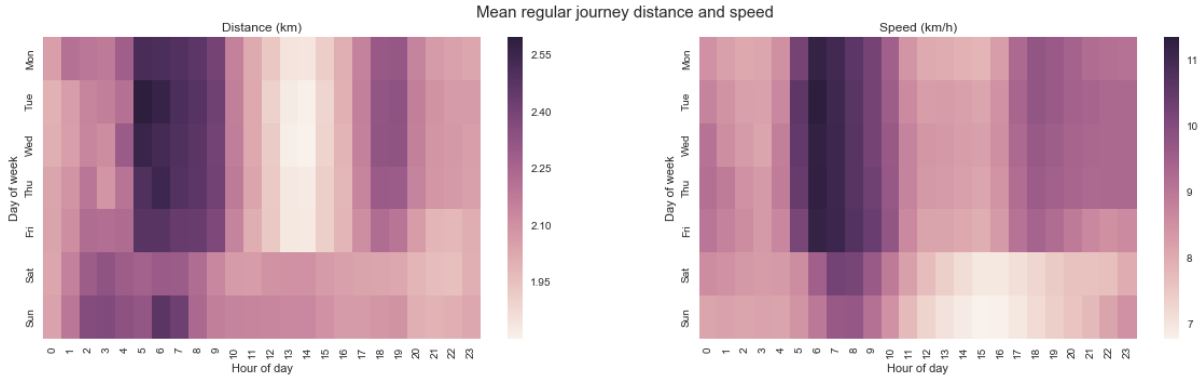


Journey distances (which could only be calculated for regular journeys) were not significantly different between seasons – in fact they were remarkably consistent on even a daily basis (X = 2.17, SD = 1.39 km). Regular journey distances were also consistent whether solo or shared. Given that regular journeys durations were higher when shared, this suggests that shared journeys were slower.

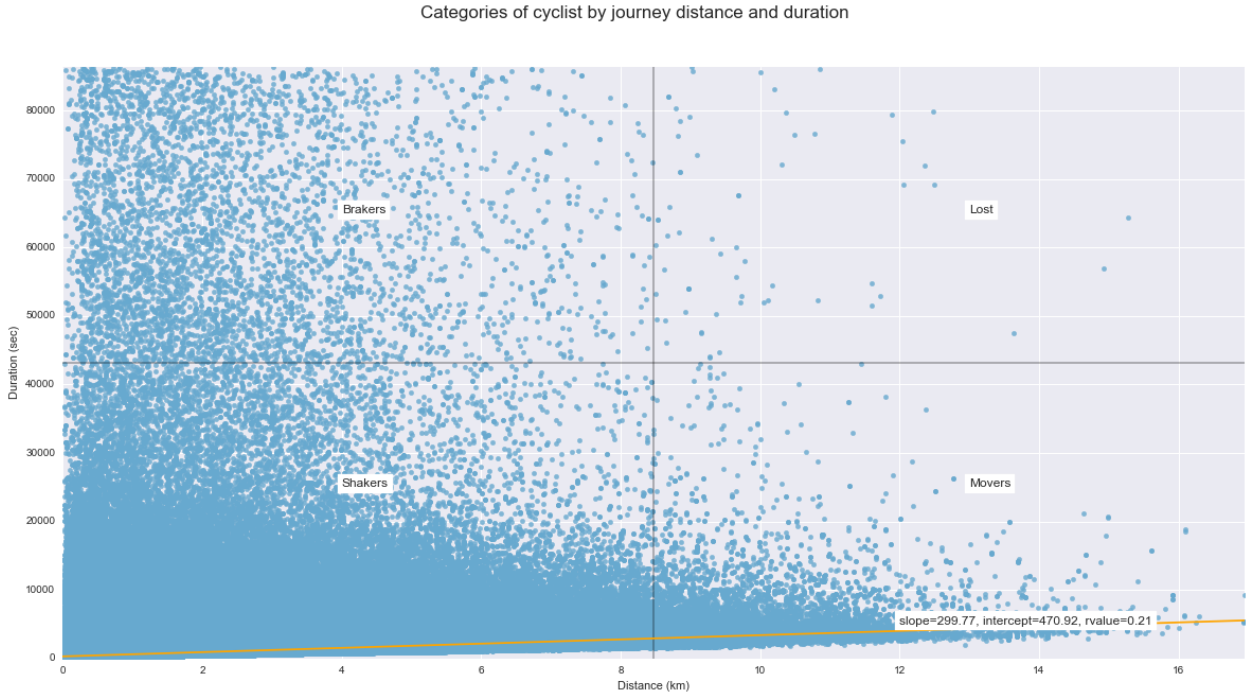




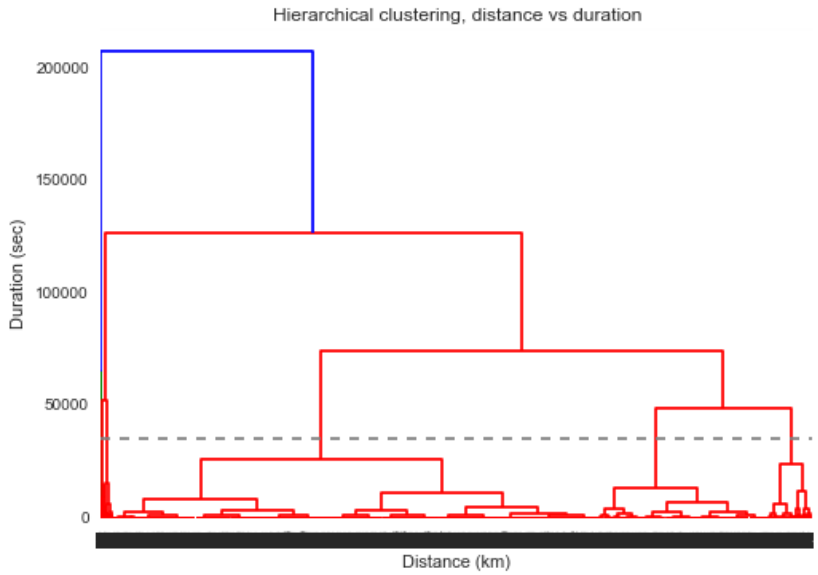
Mean journey speeds peaked during the early hours of the morning weekday commute, in tandem with mean distances. Speeds were also higher during weekday evenings, but with distances tailing off after 8pm. Mean distances for the evening commute were slightly lower than for the morning commute, which is perhaps suggestive of people cycling in to work, but getting home by other means. Both speeds and distances also rose, albeit more briefly, on weekend mornings – generally between 6 and 9am.

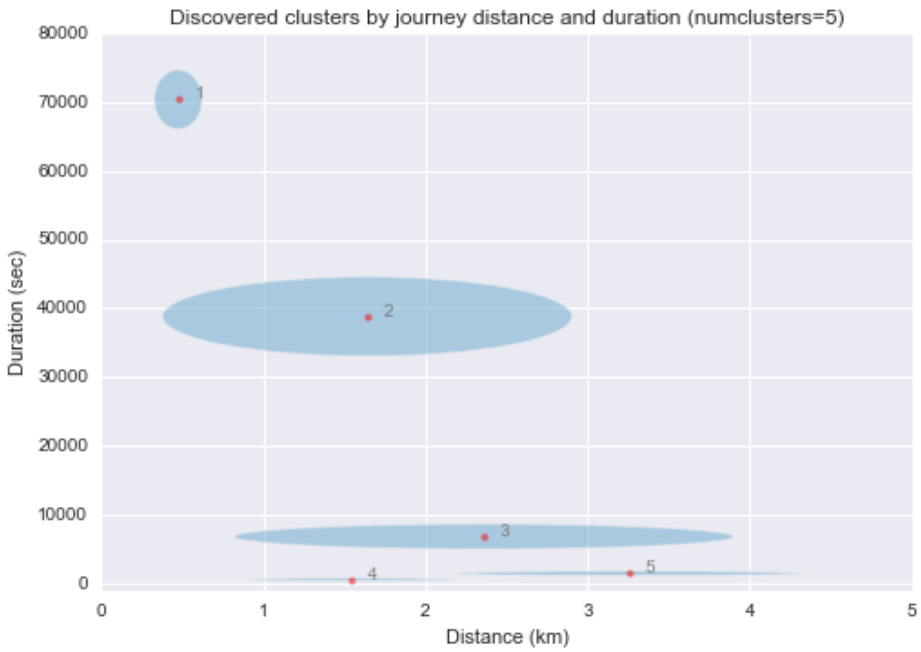


Less consistent was the relationship between distance and duration. As might be expected, there was a positive correlation between the two, however it is clear from the scatterplot, below, that there are more types of usage than just people heading from A to B at a steady rate.



Hierarchical clustering identified the potential groupings below. Note however that there does not appear to be a cluster representing the "hop on, hop off" types of journey – which might be expected to make up the bulk of hires. Potentially this is either because of an unrepresentative sample (10000 observations were used, rather than full dataset - due to PC memory limitations) or maybe these shorter journeys were too mixed to form a consistent group[14].

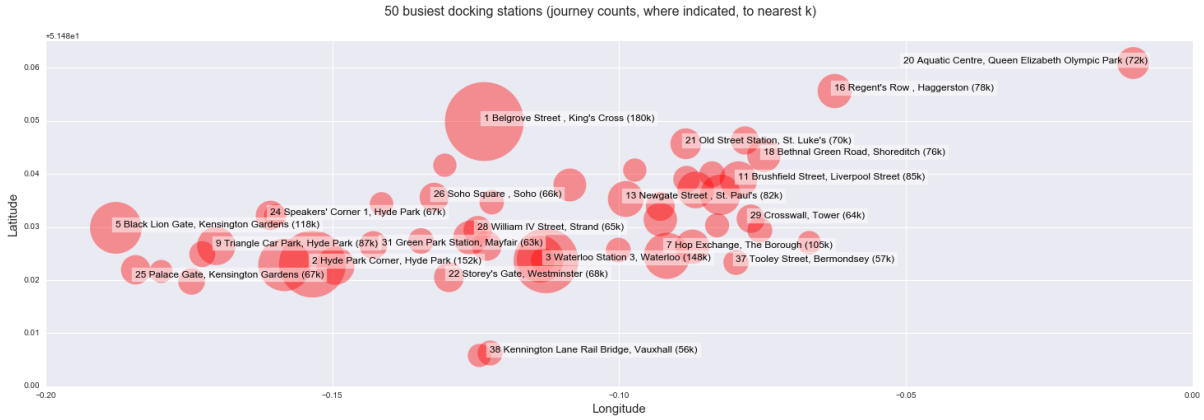




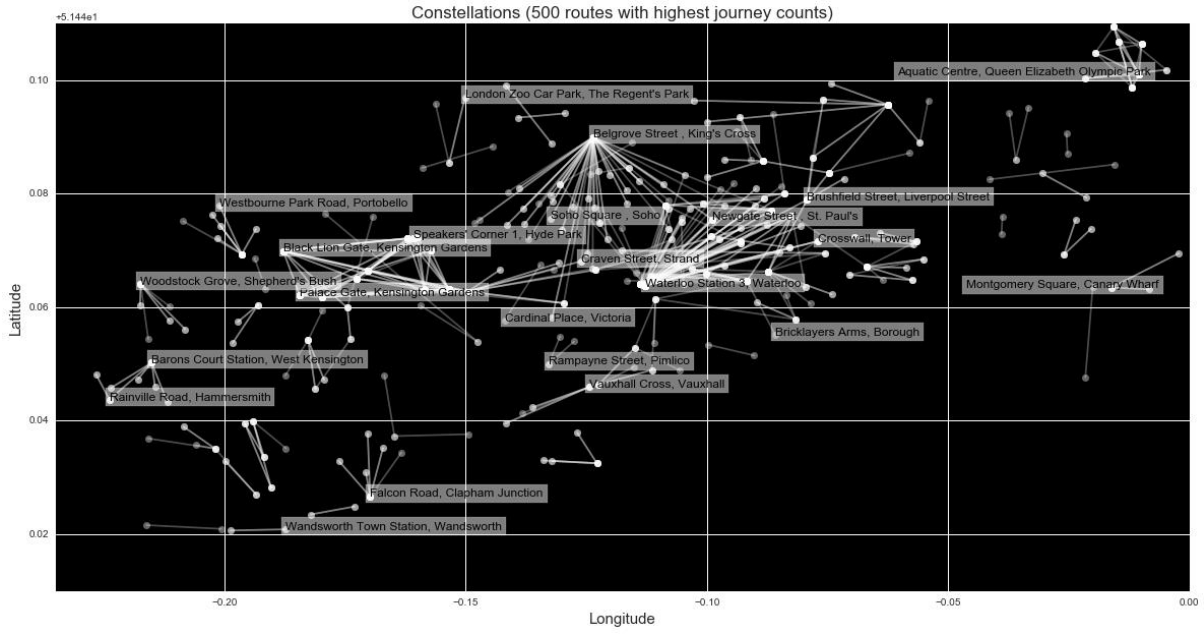
[14] Or else mystery option (c), this is an instance of hierarchical clustering in search of an appropriate application.

Minutiae

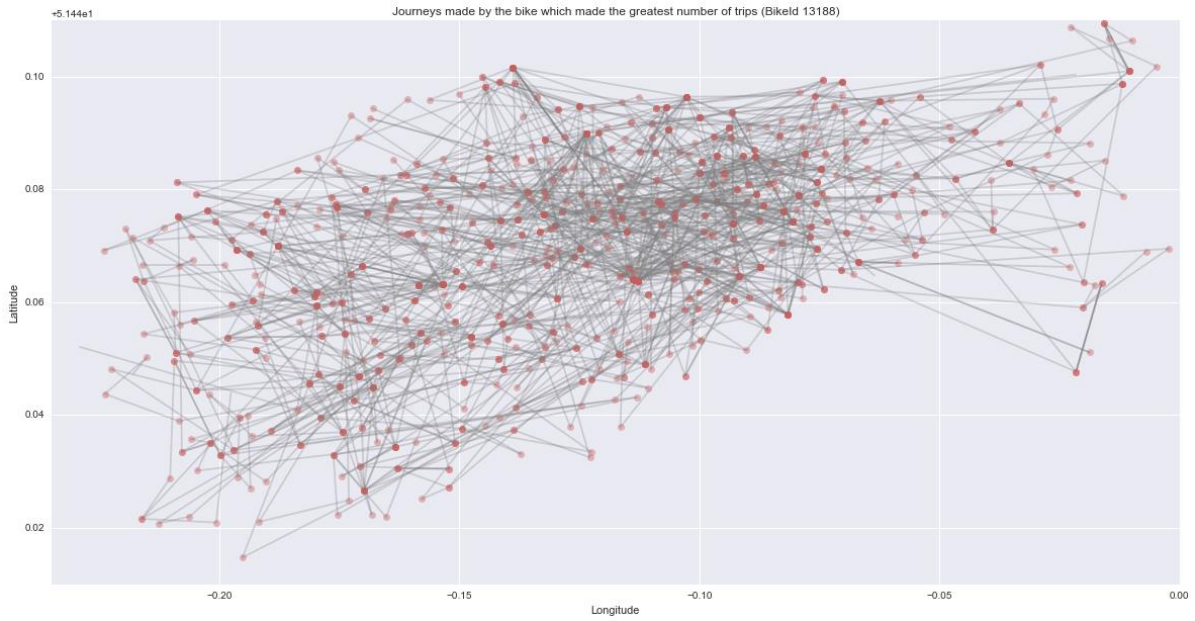
Belgrove Street, King's Cross, was the busiest docking station – with 180,066 journeys either starting or ending there. Belgrove street was also the place where bikes most liked to be; of the 12708 recorded `BikeId`s, only 201 did not check in at some point during the year.



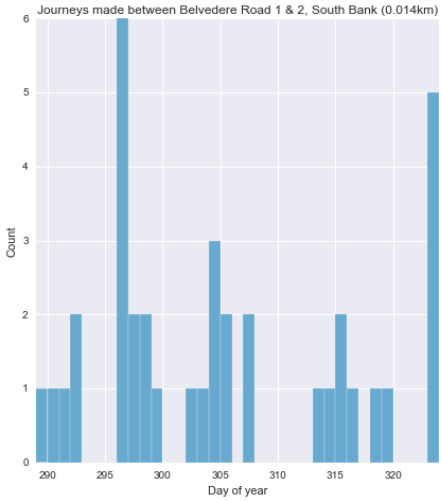
The busiest single route was between Black Lion Gate, Kensington Gardens, and Palace Gate, Kensington Gardens – with some 8,193 cyclists making the 0.9km journey. Journeys between various points within Hyde Park were close behind, and clearly from the chart below the areas around mainline stations had probably the most activity overall, just with a wider variety of individual routes to and from them.

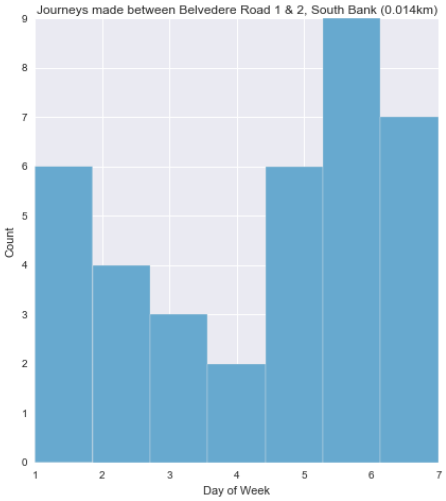


Bike 13188 made the most trips, a total of 1243 journeys covering 2560.95 km – a respectable average of 2km per trip[15]. The longest trip was in fact made by Bike 13499 on 26th March, making the 16.95km journey from Putney Rail Station to the Olympic Park in Stratford in 2.6 hours.



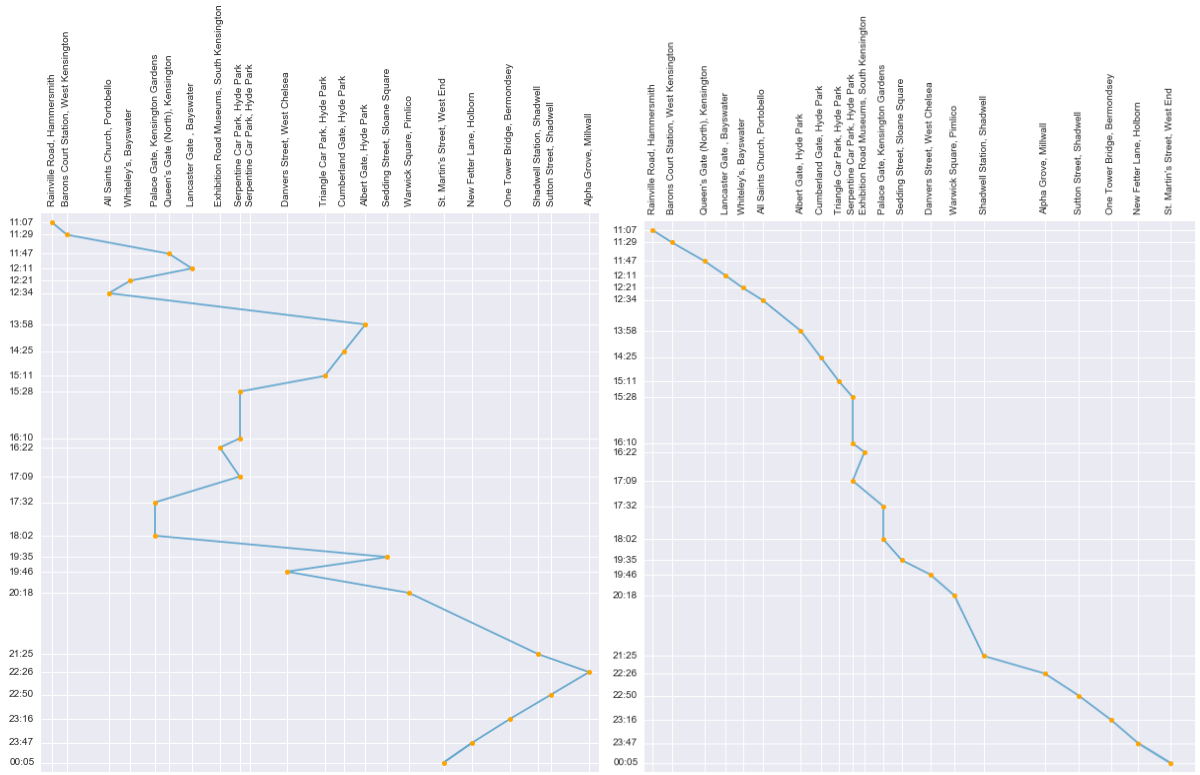
37 journeys covered the shortest distance (0.014 km) between Belvedere Road, South Bank, and Belvedere Road 2. This highlights a problem with the categorisation of journeys into circular and regular types, in that it is highly likely that these two docking stations are either in fact the same location, or else are close enough to each other that cyclists consider them to be the same place they started at. As such, these should probably have been counted as circular journeys.





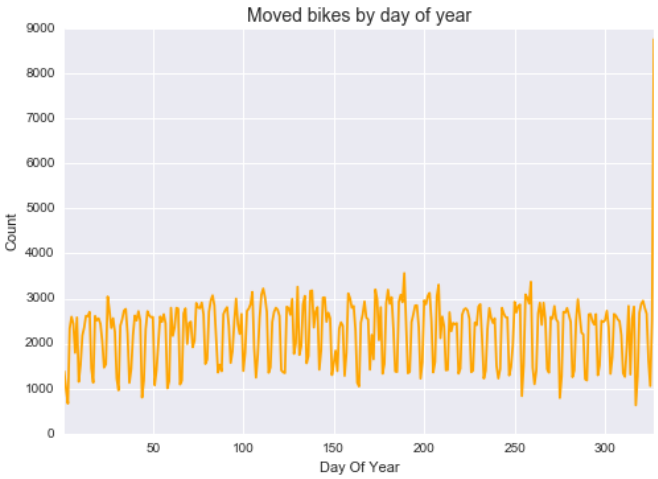
[15] The bike which made the shortest number of journeys, by comparison, is Bike #3 – which made two journeys in Mile End around the 2nd January, and hasn't been seen since.

Bike 7200 made the most journeys in one day, a total of 23 – on 31st July. Two representations of its journey are shown below[16], the first preserving longitude for docking stations, on the x-axis and the second displaying them in the order they were visited. In the latter case, the uneven spacing of points on the x-axis represents the different distances between docking stations.



The biggest group journey was made on 5th May, when 12 riders made a journey from Whitehall Place, Strand, to Crosswall, Tower around 1pm.

Finally, 833,971 bikes started their journeys at a different location to the one they ended their previous trip at. Presumably this is due to the unseen hand of TFL redistribution; the most visible spike in the chart on the right was the date of a Southern Rail strike – where some 296 bikes were moved from Waterloo Station, and 291 from Belgrove Street, Kings Cross.



[16] My (not entirely successful) attempts at something like a Marey Chary..

Conclusion/Evaluation

Overall, I think that the choice of dataset was reasonable. As was hoped, the elements of time and geographical location provided the opportunity for more varied types of chart (line, heatmap, scatter*map*, etc.), which were behind some of the more interesting insights that arose – such as the sudden drop-off in zero duration journeys, and the "scribble" chart of all routes made by one bike in a year. And the discovery of internal categories such as circular vs regular, and solo vs shared, journeys, served to make the descriptive charts more information-rich.

I don't think that any great insights came out of the data – but then this was something of an exercise in seeing how much could be done with comparatively little (in terms of attributes, if not volumes). This maybe limited the possibilities for explanatory visualisations; I think that any follow-up analysis would have to integrate more sources of information (weather, news, local topology etc.), and then focus on very specific aspects of journey data such as particular routes and/or timeframes. A more fine-grained analysis would better support scientific inquiry, whereas attempts at correlation and clustering did not in reality add much value, perhaps due to the heterogeneity of the data.

In terms of process, then I did not anticipate the impact of having 10 million rows of data, and consequently spent as much time parsing and cleaning (and waiting for) data as I did analysing and visualising it. In future I think I might spend more time up-front on initial analysis of a smaller subset, with a view to getting a better understanding of how it all fits together. For example, rather than assuming that rows with no `EndStationId`s were just ordinary journeys with missing data, I could have traced out a picture of the routes before and after these stages to see if they had any relation to each other[17]. This, coupled with the lack of a specific research question up-front[18], meant that I struggled a little to find a coherent (or maybe just interesting) narrative on the explanatory side. I think that the potential is there in the data[19], but would need to be more theory driven – in the sense of using up-front knowledge of a particular route/area/use-case, rather than being purely data driven.

[17] In a late-breaking analysis, it seems as though some of these "unknown" journeys occured part-way through a bike's other journeys – so perhaps indicate aborted attempts at docking, and as such should have been dropped from the dataset.

[18] To be fair, the nature of the exercise was always more "fishing trip" than experiment.

[19] Having found no-end of quirks and factoids, most of which didn't make it into the minutiae section.

Appendix 1: Conversion of live feed of XML docking stations data.

The stylesheet used for the conversion of XML to CSV is shown below. Note that this is lifted more or less directly from http://stackoverflow.com/questions/365312/xml-to-csv-using-xslt.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text" encoding="utf-8" />

  <xsl:param name="delim" select="','" />
  <xsl:param name="quote" select="'&amp;quot;'" />
  <xsl:param name="break" select="'&amp;#xA;'" />

  <xsl:template match="/">
    <xsl:apply-templates select="stations/station" />
  </xsl:template>

  <xsl:template match="station">
    <xsl:apply-templates />
    <xsl:if test="following-sibling::*">
      <xsl:value-of select="$break" />
    </xsl:if>
  </xsl:template>

  <xsl:template match="*">
    <!-- remove normalize-space() if you want keep white-space at it is -->
    <xsl:choose>
        <xsl:when test="name() = 'name'">
            <xsl:value-of select="concat($quote, normalize-space(), $quote)" />
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="normalize-space()" />
        </xsl:otherwise>
    </xsl:choose>
    <xsl:if test="following-sibling::*">
      <xsl:value-of select="$delim" />
    </xsl:if>
  </xsl:template>

  <xsl:template match="text()" />
</xsl:stylesheet>
```

The command used to perform the conversion is shown below. This was run on an instance of Ubuntu 16.04, running inside VirtualBox on a Windows 10 desktop PC.

```
xsltproc livecyclehireupdates.xsl livecyclehireupdates.xml > stations.csv
```

Appendix 2: Combination of chunked journey data into single file.

The command used to combine the files is shown below. This was run on an instance of Ubuntu 16.04, running inside VirtualBox on a Windows 10 desktop PC. Grep -v was used to remove the headers from each file, as part of the process. A header row was added back in, manually, upon completion.

```
cat * JourneyData*.csv | grep -v "StartStationId" > JourneyData.csv
```

Appendix 3: Jupyter Notebooks.

Regards the Jupyter notebooks provided with this submission:

- Notebook 3i was used to read in the data and make some initial corrections.
- Notebook 3ii was used for initial analysis and further cleaning of the data.
- Notebook 3iii was used for the analysis proper and visualisations.