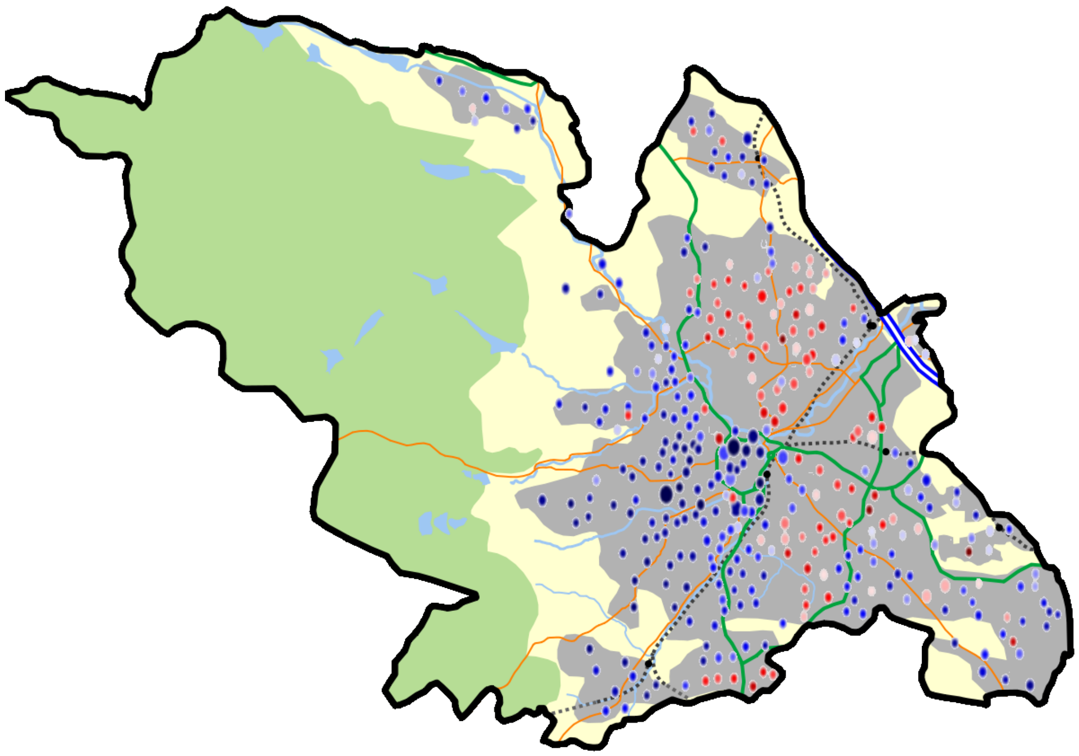
Unemployment and Public Transport: A Modelling Software

Matthew Rowland, Connor Sheppard, and Jordan Foster

17th May 2018



Executive summary

The challenge addressed throughout this project was; what are the economic barriers to employment for those in low income areas of Sheffield? More specifically, where is the best place to invest in new employment, so that it can be reached by those without access to a personal vehicle? Because of this, the project became focussed on how well different areas, specifically LSOAs, in Sheffield are connected without using a car, i.e. how easily can someone get from where they live to where they work, using only the bus system, cycling, or walking?
To tackle this challenge required the use of graph theory, a mathematical technique which plays an important role in our lives, though we rarely know it. In short, it is what allows SatNav devices to work; by putting in start and target destinations the SatNav can use graph theory to look at all the different ways to complete the journey, selecting the fastest, cheapest or shortest route.
With around 60,000 routes connecting all 345 of Sheffield's LSOAs, finding the best route which connects every LSOA is a challenge best suited to a computer. As such, a model was developed which was able to act like a SatNav; finding all the possible routes and then selecting the best ones. Applying some more mathematical techniques allows the model to compare different LSOAs where employment could be created and select the one which is easiest to reach for the majority of the target demographic. Therefore, the model can locate the best place to invest in new employment. By changing the data the model looks at, to for example the LSOAs in Chesterfield, the best place to invest in employment there can also be found.
With time, this model could be developed to tackle core issues such as:

- The placement of new housing.
- Where new infrastructure is required.
- Analysing the efficiency of existing transport links.

The key items needed for the advancement of the model include:

- GPS data for bus/tram stops and for active bus/tram services.
- Resources to aid in the development of a general user interface i.e. making it so the user is not required to "run a programme".

In summary, this project has shown that it is possible to develop a model able to locate optimal sites for employment investment. In addition to this, this project has laid the foundations for a model capable of an in-depth socio-economic analysis, able to consider housing, transport and infrastructure of any town or city, not just Sheffield.

Part I
Introduction

At the beginning of this academic year the Sheffield City Region Combined Authority (SCR) presented the University of Sheffield (UoS) with a question; what are the economic barriers to employment for those in low income areas of Sheffield due to public transport costs? Specifically, the task was to investigate these issues using data analysis and computer programming techniques. On completion of the task it would serve as proof of the concept that such a complex real-world problem could be represented with a relatively simple computational model; yielding meaningful results.

Throughout this project attention was focussed on public transportation links with an aim to create a model of Sheffield which could map various Lower Layer Super Output Areas (LSOAs) and their connections. Through this, the way in which different demographics were able to travel to different areas of Sheffield could be analysed. Sheffield alone was chosen from the SCR to limit the complexity of the problem and allow for the model to be further developed before its applications to other areas within the region.

To create a computational map of Sheffield required the use of a mathematical technique; graph theory. This technique was chosen for the ease with which it can be applied in computational work as well as the way in which it depicts relationships, allowing for even the most complex issues to be tackled. It was recognised that availability of transportation takes a vital role in enabling and encouraging individuals to access employment. Some of the barriers to employment public transportation presents include: cost, travel time, availability and ensuring individual needs (e.g. disabled access) are met. To improve the accuracy and applicability of the model developed throughout this task the factors mentioned and several others were considered.

By further developing models like the one developed in this project, information could one day be extracted which could aid the SCR in investment decisions; specifically, regarding the implementation of new (or by contrast the removal of) public transport links and identifying locations for new business opportunities.

Part II
Method

The original question posed by the SCR could be considered in many different ways. In this report the problem was interpreted as how to find areas in which new low-skilled employment could be created so that as many of those seeking that level of work could reach it via public transport. Upon receipt of the brief and noting the key aspects of the problem, it was decided that graph theory was the most suitable approach. Graph theory is used throughout mathematics and computational modelling to construct graphs or maps, which depicts the relationship between any two connected points. The points and connections are often referred to as nodes and arcs, respectively. In our case each node, or point, was used to represent a lower layer super output area (LSOA) while each arc was used to represent the time of the shortest public transport route between LSOAs. The aim was to create a map of Sheffield using graph theory and locate items of interest, such as easily accessible locations. The specific computational approach we used is called Dijkstra's algorithm. It finds the shortest 'distance' between two nodes where distance can be one variable, like time, or a weighted combination of multiple variables, such as time, cost and the number of buses required to make a journey. This allows us to model various relationships between all LSOAs in Sheffield. The purpose of the map was to find how all LSOAs in Sheffield were connected to each other and then finding the minimum weighted 'distance' between all of these points. From here additional points could be added to represent potential new job sites and then tested to see which was most easily connected; thereby locating where was the optimal site for job creation.

We focused solely on Sheffield rather than the whole SCR. This simplified problem significantly and meant a model could be developed in an environment where mistakes could be easily located and corrected, before upscaling the model to the entire SCR. Sheffield's size and infrastructure provided enough of a challenge to develop a robust model while not being so challenging to prevent the development of the model.

To simplify the task further, Sheffield was also divided into smaller areas. After discussing the available data with SCR it was noted that LSOAs would be suitably sized to the task, with an average population of 1500 residents. It was established that the nodes forming the map of Sheffield would be represented by the LSOA population weighted centroid data, as this would be a reasonable approximation of where majority of people live and therefore would access a particular public transport route. The data pertaining to the population weighted centroids of LSOAs was found from the 2011 census, via the Office for National Statistics (ONS).

The major assumptions made by the model thus far are summarised as:

- The location of the majority of the population and public transport links are approximated by the LSOA population weighted centroids.
- LSOAs are either relatively small, or the centroid sits on a population centre.
- Public transport is used in place of a car.
  - The target demographic are unlikely to have access to their own vehicle
- Any bus can be caught in any direction.
  - A significant achievement would be to eliminate this, and allow the consideration of one-directional buses.

- There is one public transport link between points. Made to simplify the data acquisition.

The necessary data was extracted from the 2011 census and can be split into four categories:

- Population breakdown data, which is at LSOA level and is the lowest resolution available to the general public.
- Locality data; the population weighted centroid longitudes and latitudes of the LSOAs and the unemployment fraction of an LSOA.
- Workplace data: the total jobs in each LSOA and the jobs broken down by census job category in each Middle Layer Super Output Area (MSOA).
- Conversion Data: which Output Area (OA) corresponds to which LSOA to which MSOA. This is required as not all of the data is available at all levels, so conversion between output areas is necessary.

In order to create the map of Sheffield, we decided to make use of Google Maps' database. This was used to form the arcs connecting different nodes. To import this data into Google Maps required a piece of code capable of converting the centroid census data into a format which was compatible with Google Maps' Application Programming Interface (API).The API allows us extract the necessary data from Google's servers e.g. time taken between LSOAs for buses. We set out locating a piece of code able to carry out the task, and were able to find a such a module. This is the 'googlemaps' module; and is availible in an open source format on the internet. This allowed for the automation of the arc creation process through Google Maps' API. It was then possible to construct a map of Sheffield, showing times and distances of the shortest public transport routes between the centroids of Sheffield's LSOAs.

After a mid-project review, it was possible to highlight several key areas of interest to the SCR. We identified which of these suggestions would be of most use to SCR, but could be implemented within the remaining timeframe of the project. Those which were suggested but not feasible to complete within the remaining time frame alongside several others are documented in the Future of the project section.

By the addition of a new layer, the model could now assess current employment sites appropriate to the target demographic and locate any pre-existing problems (e.g. are there difficulties reaching these locations from other deprived regions). To do this required access to data which identified occupation and location. Due to this data being disclosive it was received at MSOA level. Consisting of a number of LSOAs and on average 7200 residents, therefore MSOAs have a lower resolution compared with LSOAs. This change in resolution meant the existing code had to be adapted in order to work with both MSOA and LSOA formatted data.

We also chose to model the variation of transport links at specific times. Variation with time had to be considered to allow for shift workers to be included in modelling process and as a general improvement to the accuracy.

Cycling, walking, and driving were incorporated into the existing model. By altering the coding of the model, the user was given the ability to choose the Google Maps' API transport method, so changing from buses to cycling 'distance', allowing exploration of all forms of transportation offered between locations. All of these modifications were encompassed within the code and it was demonstrated how conclusions could be made using real employment data. This model has been applied to areas throughout the entire region and has successfully demonstrated the full capability of the model. Though still only a prototype, the model has the potential to extract far more accurate and interesting results with little development. These developments, as well as a discussion on the potential power of the model are found in the *Future of the project* section.

Part III
Sheffield LSOA Public Transport Results

Much information can be found and examined from the outputs of the model. Included are plots and data which give specific and holistic information and deductions about transit throughout the city, as well as a showcase of many of the features and capabilities of this modelling technique. Firstly, information that can be gleaned from an analysis of the raw census data is presented to give a holistic view of the city. After this, a breakdown of the results of our modeling are given, finally followed by possible recommendations following on from this model. All data given pertains to the Sheffield City Area, as shown in Figure 1.

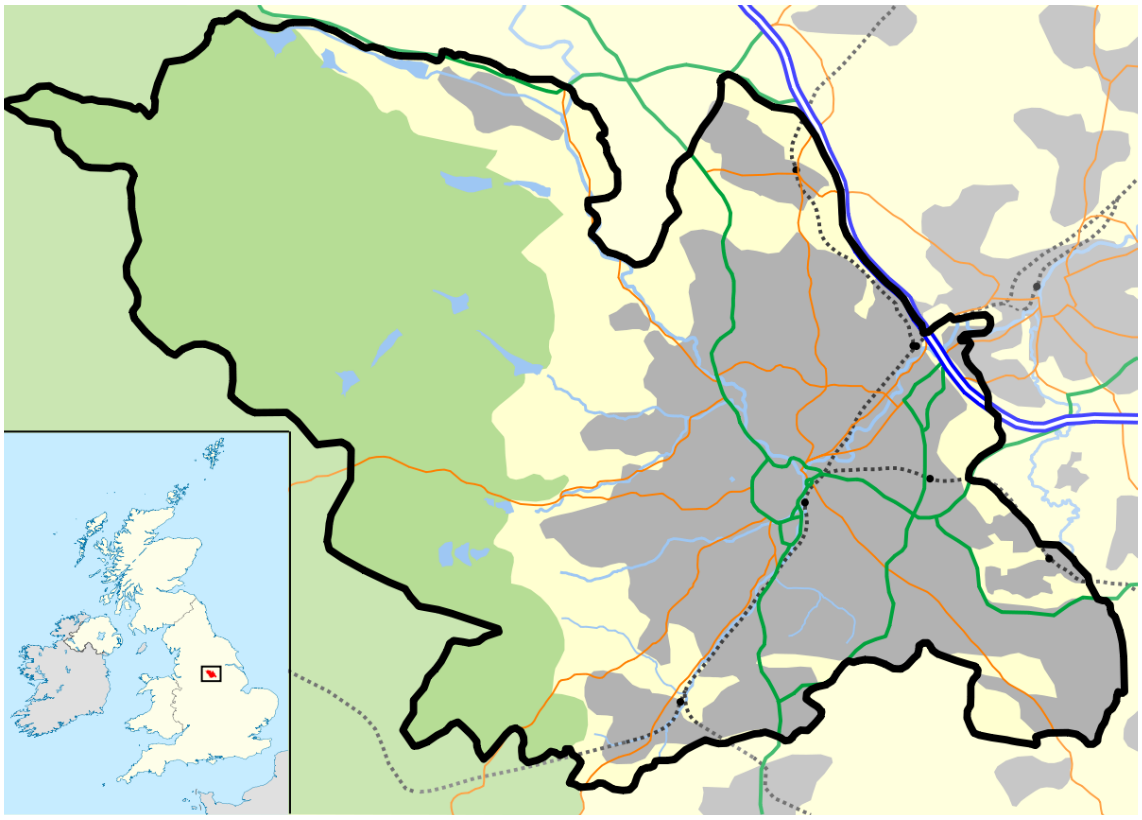


Figure 1: Map of the Sheffield City Area, provided for contextual reference.

The results here can be divided into two broad categories: information that could assist with improving access to existing jobs, and information relating to the optimal placement of new jobs.

III.I Information from census data

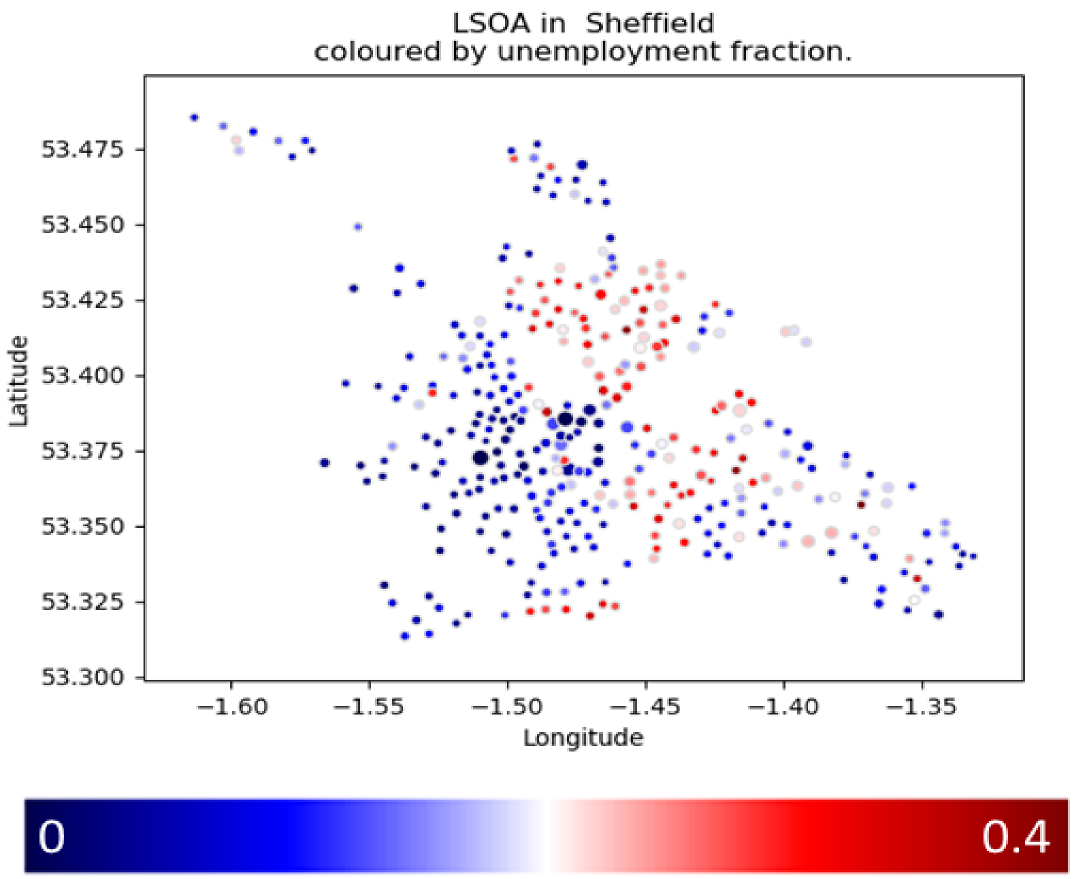


Figure 2: LSOAs in Sheffield coloured according to the unemployment rate, highest in dark red.

These map-like figures show the chief property listed on a colour scale as shown below the figure. Numbers above the maximum value are represented as the most extreme colour, although these values have been chosen to ensure the colour range is a meaningful indicator. The size of the point is in proportion to the total resident population in the area it represents. Whilst the longitude and latitude do not map perfectly onto a flat surface, at the scales used the mapping is adequate for visual aid purposes.

These visual representations of the city allow us to infer much about how the city is really broken down. From Figure 2, a representation of the unemployment in LSOAs across the city, three chief areas are clearly visible. These correspond to the Hillsbourgh, Shiregreen and Pitsmoor area surrounding the Northern General; the Area around the Arbouthorne between Tinsley and Gleadless partially bounded by the A6102; and the Woodseats-Whirlow corridor. Hence clearly visible is a split between the eastern outer third and and the central and western parts of the city proper.

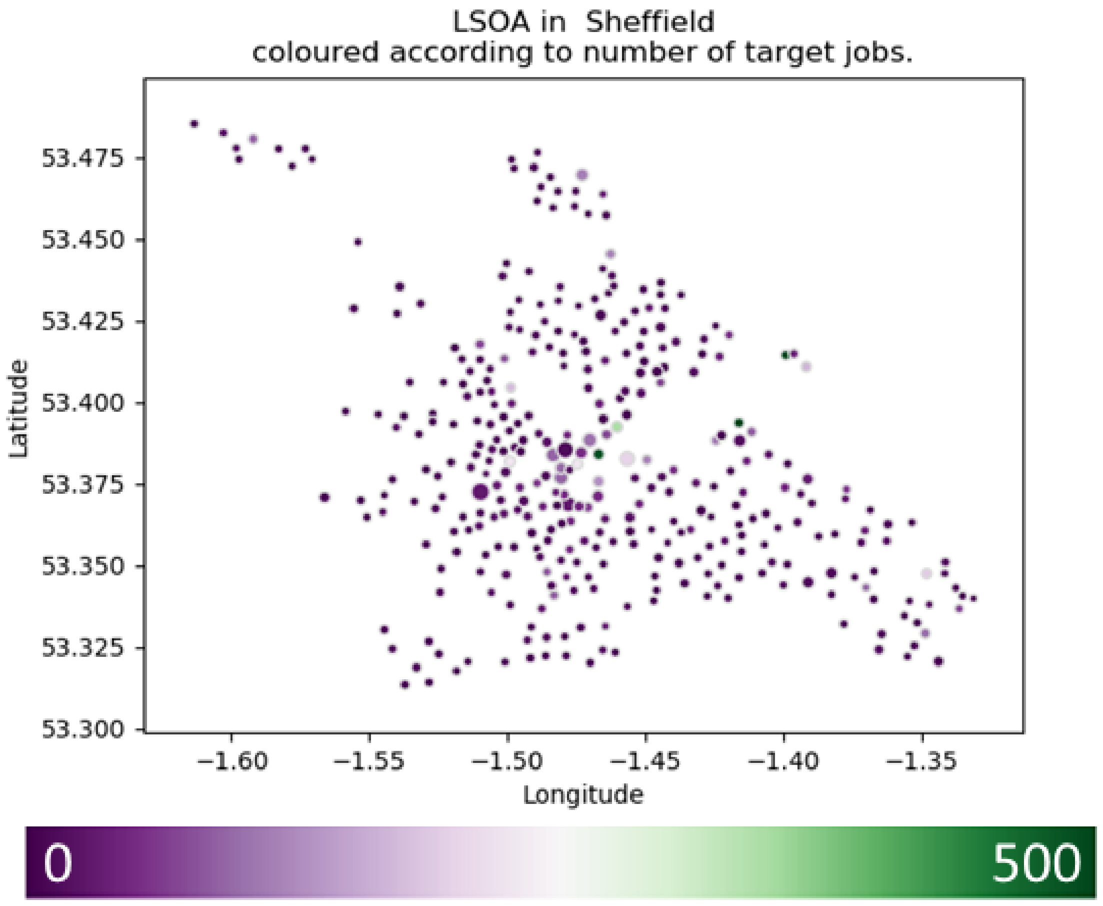


Figure 3: LSOAs in Sheffield coloured by target jobs

Figure 2 shows low-skilled job distribution. Visible here is the concentration of low-skilled jobs in the city centre, with spots centred in various other locations, such as Stocksbridge, Mosborough, Chapeltown and Hillsborough. When viewed in conjunction with Figure 2, it is not unreasonable to deduce that the nominal Sheffield region does not function as one economic unit, and in light of this, measures to take account of the clustering would likely refine the accuracy of the results.

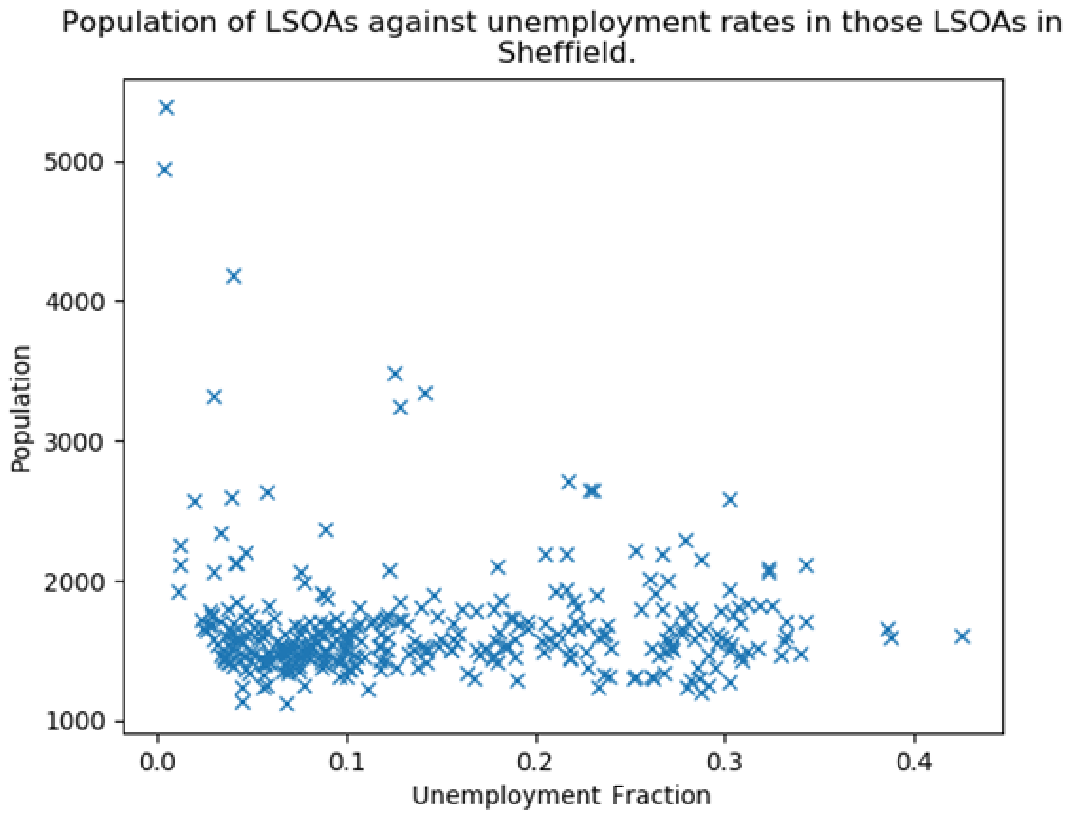


Figure 4: Population of LSOAs in Sheffield platted against unemployment rates there. Correlation= -0.058, p=0.279.

From these graphs we see that there is no real correlation between the population of an area and the unemployment levels. This implies that the areas considered do not exist in disconnect from the outside environment: if they did we might expect that a smaller community would have a higher rate of unemployment as locally limiting factors of population size would limit the economic potential. Given this is Sheffield, this is not surprising, such an effect would only likely be noticeable in an area both very rural and impoverished. For a greater analysis, measures of income may prove useful here, although such measures are beyond the scope of this project.

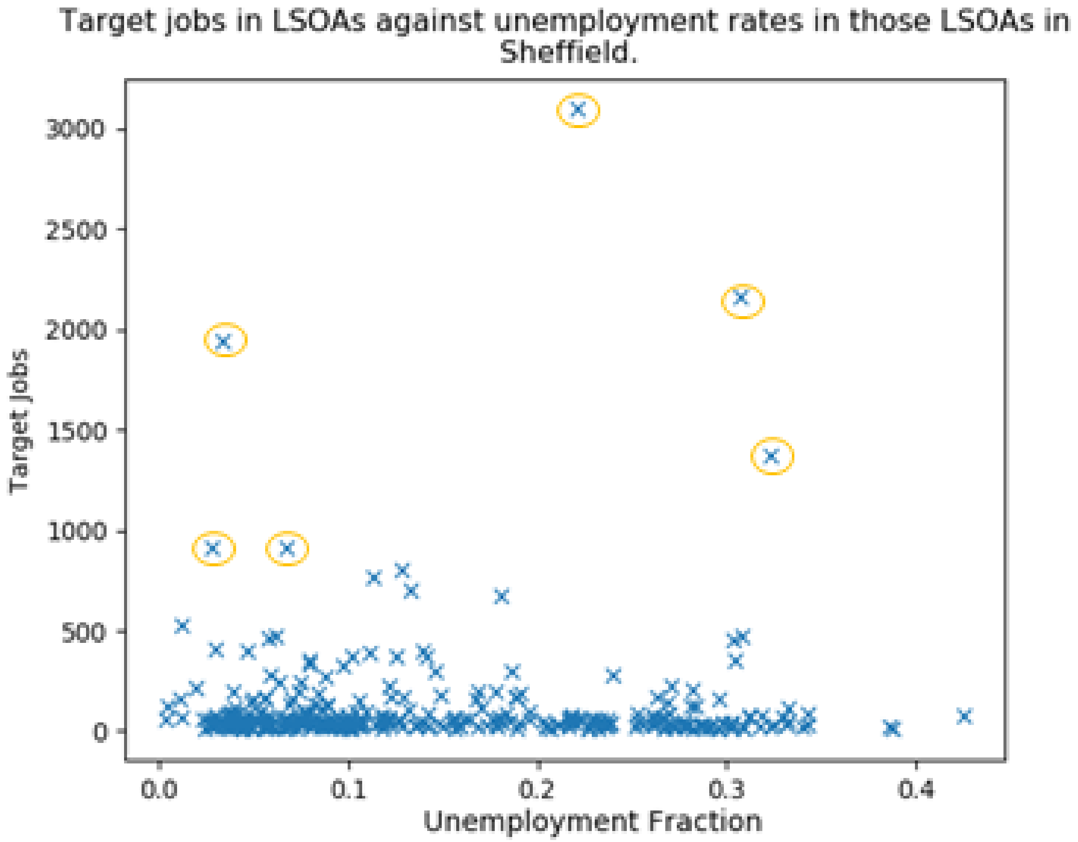


Figure 5: Target Jobs at LSOAs against unemployment there. Correlation=0.023, p = 0.664 High job points ringed for emphasis.

In Figure 5 we also clearly observe no correlation between the number of low skilled jobs and the unemployment fraction of an area. Clearly visible, rising above the majority are a handful of “job hotspots” (ringed for clarity). These are also distributed along the unemployment rate spectrum, which demonstrates that low skilled workers evidently transit between locations, and that employment is not “hyper-localised” to the LSOA level.

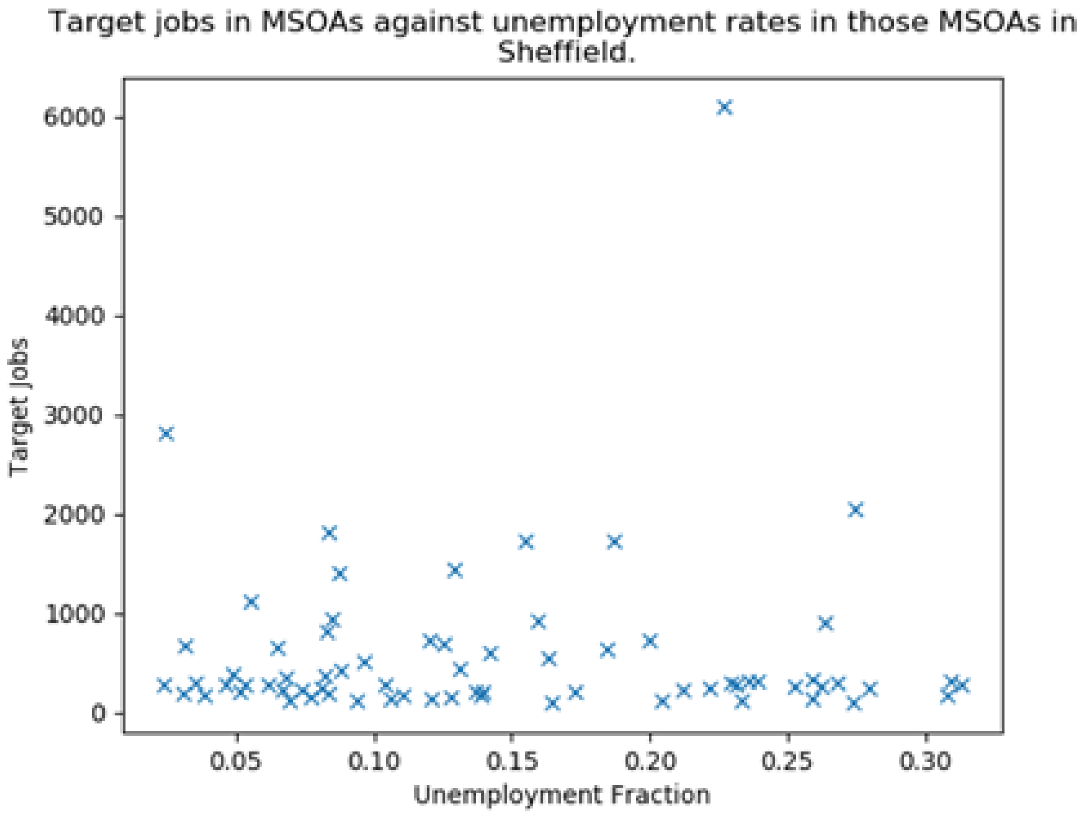


Figure 6: Target Jobs at MSOAs in Sheffield against unemployment there.

At the MSOA level, we observe the same phenomenon, as can clearly be seen in Figure 6. With a wider area as the locality unit it may be possible to infer the size of region that individuals generally move about for work purposes, although these regions may be so large that variance within is more significant than variance between. A seperate study into the clustering of the LSOAs in spacial, economic or transportative areas may give more useful areas for consideration.

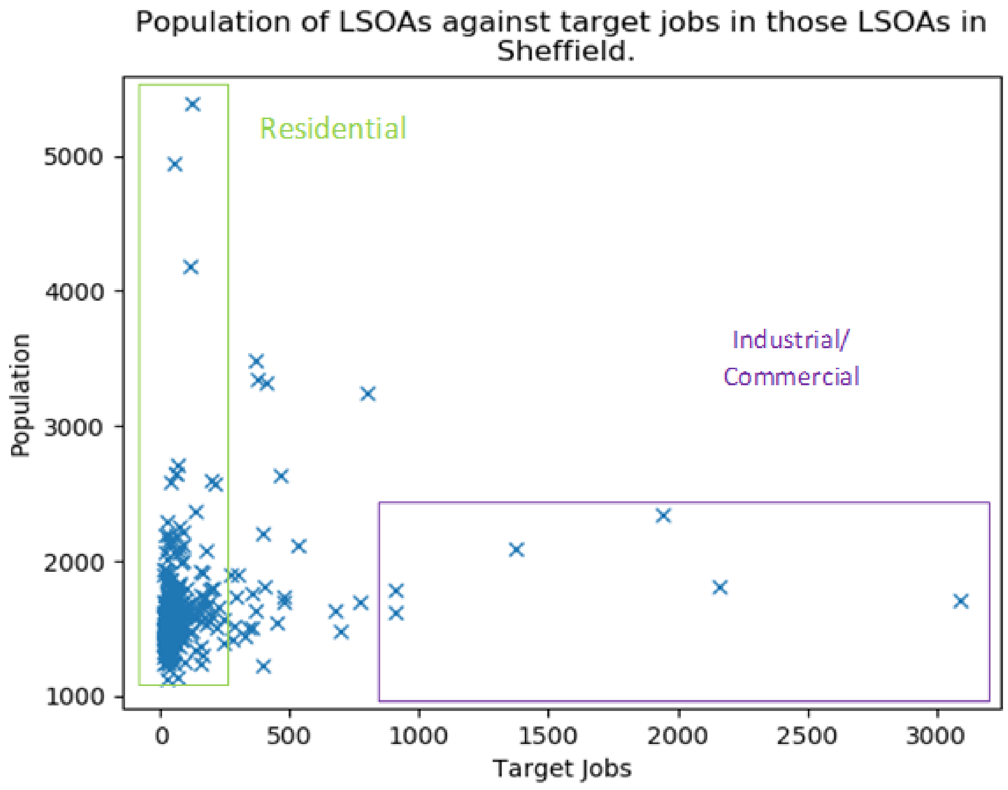


Figure 7: Population at LSOAs in Sheffield against Target Jobs there. Correlation = 0.168, p-value =0.0017

In Figure 7, a mild correlation exists between area population and low-skilled jobs. This is somewhat unsurprising; the presence of people predicts jobs, to a small extent. A degree of polarisation between these two factors evidently exists however, given no points populate the upper centre and right sections of the plot. The boxed areas denote areas of the plot populated by more residential or industrial regions. This is not a hard-and-fast distinction, as the nodes are defined by LSOAs - which ultimately are determined by residence. However, it does give an indication as to how industrialised or commercialised an area is upon comparison with other areas. This plot also allows us to see easily that there do exist areas where employment of the type we are interested in is concentrated heavily. Upon inspection of Figure 2, we can see that these areas are distributed across the city, primarily in the eastern half, with the higher-job locations often featuring at the centre of an area. Considering this, we see that in Sheffield, even in small areas, it is apparent clustering occurs. For a more detailed analysis of what is going on here, and thus how access to work can be improved, identifying apparent economic clusters which have formed naturally, and where people working in such jobs work relative to their homes is important. This is to find if real clusters do exist, and to what extent that is the case.

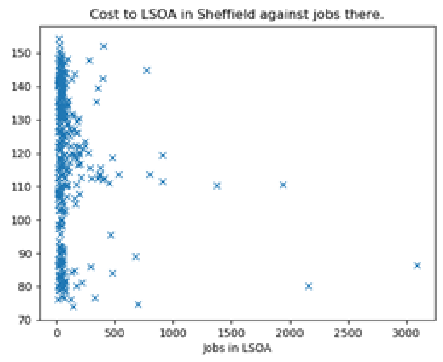


Figure 8: Cost to an LSOA against the jobs at that LSOA, all data.

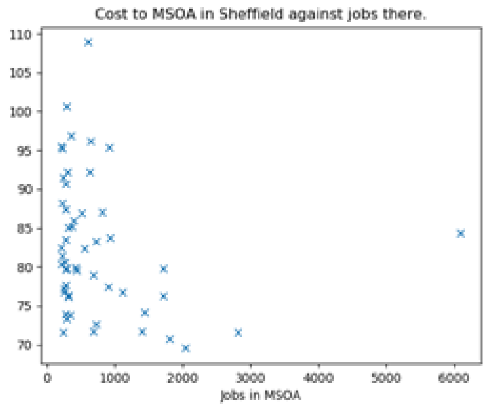


Figure 9: Cost to an MSOA against the Jobs already there, sites with less than 200 jobs excluded

III.II Modelling Results

Here we have data obtained via the use of the routing algorithm, on the y-axis is the "cost" to a node- left dimensionless as dimensions depend on the cost coefficients chosen. In this case the coefficient set considers time and converts money into time at the National Living Wage, hence the dimensions are effectively minutes. This is to say: the cost on the y-axis of these plots is the time one travels for added to the time one would have to work at the Living Wage to earn the fare. The most striking element of the plot is the line along the left-hand side representative of very low job LSOAs. Figure 9 features Sheffield under the same costing and time conditions, but at an MSOA level with areas with less than 100 jobs omitted.

Whilst there appears to be a large difference in cost between the two graphics, this is accountable considering that the scale is in minutes as discussed above. Additionally, the collectivisation of LSOAs leads to centroids that were more "out of the way" being closer to population centres and thus better transport infrastructure. Given the place with the most jobs has a cost to that MSOA of 85 minutes, lowering this average can be most easily achieved by assessing the bottlenecks that occur when you consider the figure of ~ 85 minutes as the maximum tolerable cost.

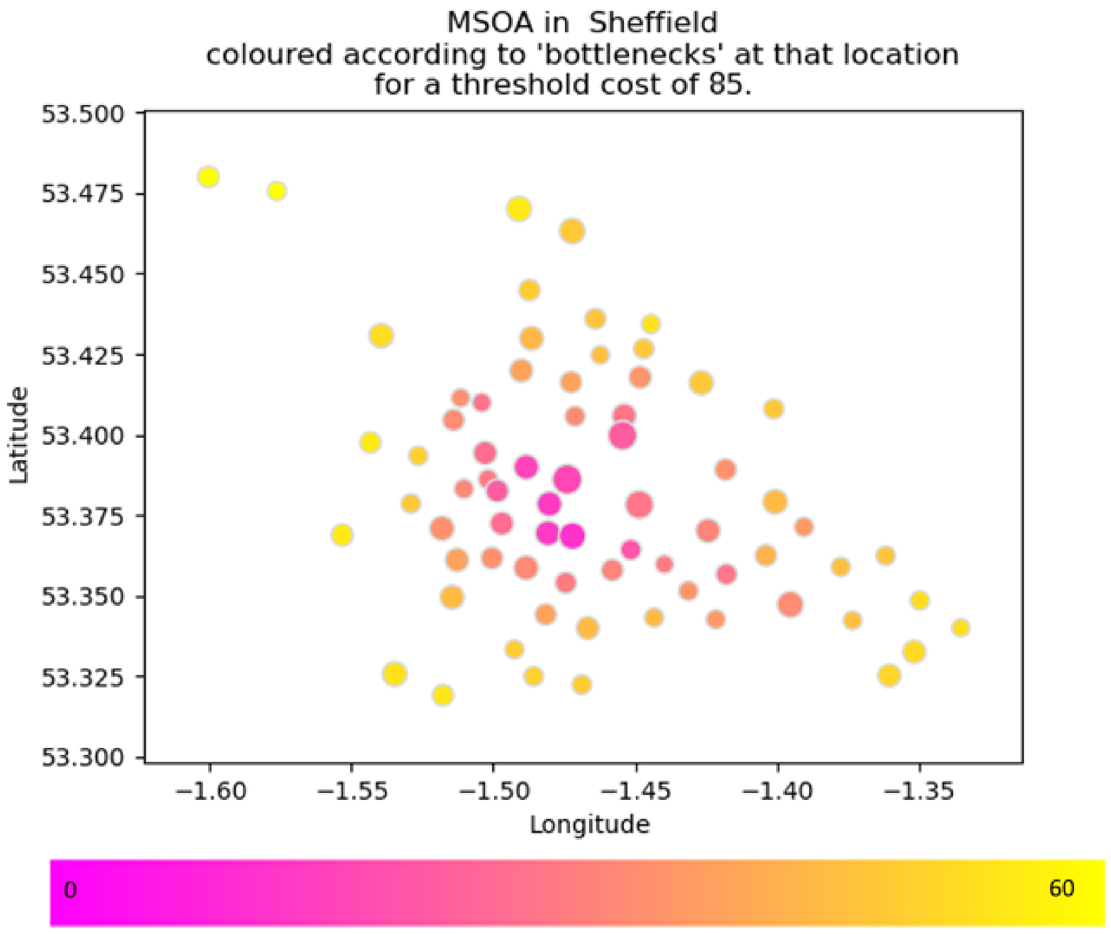


Figure 10: Colour-map of bottlenecks for a cost of '85'. A bottleneck is a point at which the traveler cannot proceed any further along their journey as they have reached the limit of their tolerated cost. This has been assessed for every node on the map, with totals found.

Figure 10 shows a colour-map of bottle-necking nodes for this cost. It can be observed that of the already noted areas of low employment, Shiregreen and the Northern General area seem to be the least well connected to the rest of the map. Thus, we can conclude that to increase accessibility to existing low-skilled jobs, further work on public transport in these, and other areas listed in the Bottleneck files (as discussed in the *How To* section of this report) may be beneficial. Also visible as not enjoying optimal public transport access is the area of Stocksbridge, which has a low unemployment rate. This points again to the notion that the nominal Sheffield area is not one perfectly fused economic area, and that presumably, the physical separation of Stocksbridge from the rest of the city has a manifest effect on the economics of transport-using workers.

III.III Transport and Job Site Recomendations

Results show that MSOAs E02001632, E02006844, E02006843, E02001630, E02001650, E02001652, E02001636, E02001653, are the best sites to have jobsites at, given that they have the lowest cost to reach of any MSOA per head of unemployed population. These eight sites have between 200 and 2,200 jobs at present, so there exists a range of well-connected potential sites, with different current economic environments and locations both in and out of the city centre. These are simply the eight lowest cost to get to, their viability otherwise has not been evaluated.

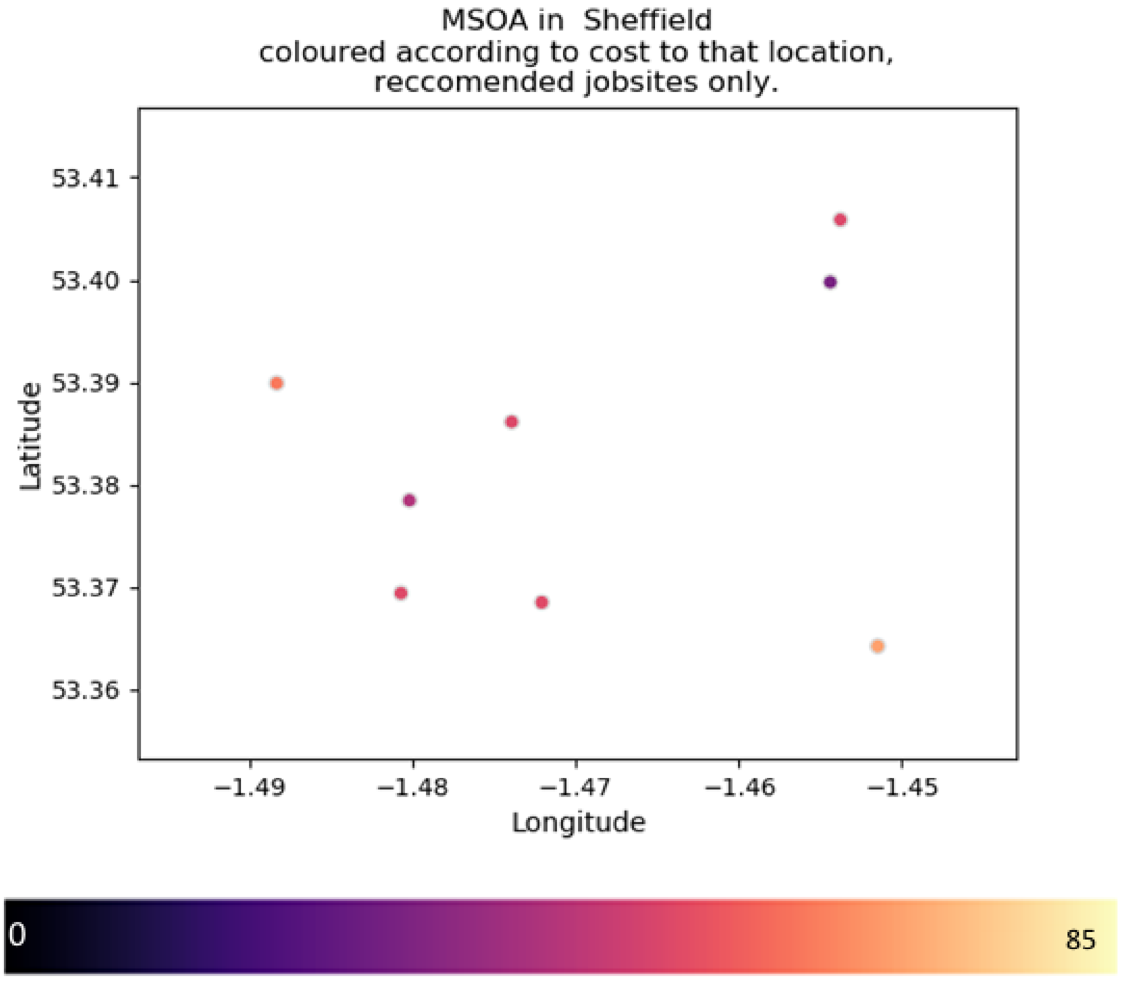


Figure 11: Potential new jobsites. The eight lowest mean cost per unemployed head of population, coloured by cost to that location. (Size is not indicative of any factor in this plot)

These sites take between 65 and 90 combined currency-time minutes on average per head of unemployed population to reach, which is not small, however one must remember that this is inclusive of the time taken to earn the bus fare as well as transit- this does not represent an hour and a half commute for the average person. Furthermore, in reality a person would be more likely to get a job closer to home, this is simply a measure of the general accessibility of a point to the population we are interested in.

As already mentioned, transport in Shiregreen and around the Northern General may be a significant factor in the unemployment issues in those areas. For more specific recommendations, model improvements as discused in the *Future of the Project* section will be required. Significantly we find that many places in Sheffield, such as Stocksbridge and Dore are not well connected to the city, but do not experience very high levels of unemployment, which indicates that the Sheffield Area is not neccesarily a unified economic block, at least for low skilled workers.

Part IV
Future of the project

As it stands the model has surpassed what it was originally hoped to be; merely a proof of the concept that mathematical techniques could be used to tackle the large scale issues a city faces, such as where to invest in new employment. During its development, ways of improving the model became apparent that were outside of the available timeframe. In this section both the short and long-term aims of the model are discussed, as well the work required to achieve them.

In short, the model developed throughout this project works similarly to a SatNav, locating all possible routes between a start and end point before selecting the best route. The route which is considered ‘best’ can be changed depending on the preference of the user, e.g. someone unable to walk far will favour a route which minimises walking, even at the cost of increasing travel time or expense. By looking at all the routes to and from every LSOA in Sheffield the model can create an effective map of Sheffield, showing the optimal routes. By applying the appropriate mathematical techniques it can then compare how well different LSOAs are connected to all others; leading to a decision on where mathematically is the best place to invest in new business/employment i.e. the model can find which LSOA is easiest for the majority of people to get to cheapest/quickest.

The model can consider four different modes of transport at any time of day: walking, cycling, driving, and the bus system. Being able to consider how the time of day impacts a route is important in gaining a more accurate picture of what it is to travel around Sheffield. With this the results come ever closer to accurately representing a real commute.

In the short term, the accuracy of the model could be increased quite significantly. Introducing small adjustments such as the way the model determines which route is best, or adding in position (GPS) data for public transport would help provide a more accurate picture of Sheffield. The key points in the short term development are:

- Breaking down routes into smaller segments to remove the assumption of all journeys being possible to carry out both forwards and backwards (A to B, B to A).
- Including GPS data; significantly improving the accuracy of suggested routes.
- Altering the way the model evaluates time will allow it to include the convenience of a route as well as its time and cost, leading to it ‘thinking’ more like a commuter would.
- Add information on areas which are preferred for investment in employment (e.g. brownfield sites, entrepreneur zones, etc.).

Currently, the model finds all the possible routes between LSOAs on the assumption that if a journey can be made in one direction (say A to B) then the same journey can be made in reverse (B to A). This is not always true, nor is it accurate to assume a journey made in one direction will be the exact same in reverse. An example of this is circular bus routes, where the journey time from A to B will be very different compared with B to A. Finding a way to further break-down routes into smaller segments should help the model deal with this.

If the model could be given access to position (GPS) data for public transport then the routes which it forms between LSOAs could not only be a lot more accurate, but update with time too. Updating with time would mean that the model was able to consider how different routes varied over longer periods of time, such as months or even years. This would be an important step in reducing random data, again contributing to the accuracy of the model.

If your SatNav provided you with two routes, one which was quicker but involved toll roads and another which was free but took longer, which would you choose? Your answer will depend on the time saved and the additional costs of the toll roads. Answers will vary from person to person, but the answers all come from the same idea; weighing the value of time/monetary cost against the convenience. To include this sort of thinking in the model would require changing the way in which it evaluates time and would likely require some research into different individuals evaluation of time. With its implication this change would bring the model closer to considering different routes as a commuter would.

Finally, for the short term more information is required on areas within Sheffield that are desirable to build on or preferably avoided. The position of things such as greenfield/ brownfield sites, greenbelt limits and entrepreneur zones would allow the model to highlight particular sites which are preferred for job creation/investment and avoid the opposite. Implementing all of the mentioned short term changes would go a long way in improving the accuracy of the model within Sheffield, as well as paving the way for the long term potential.

In the long term the model could do more than just locate the best sites for employment investment; where is best to create additional housing or infrastructure, (e.g. where a bridge could be introduced to ease congestion or where housing is needed most) could also be considered. As well as this existing housing and infrastructure could be looked at with the intention of finding areas which required additional funding. Simply put, the model developed in this project has the potential to be a powerful decision-advising tool. Key long term development plans include:

- Introducing a general user interface, making the model simpler to operate.
- Adapting the current model to look for optimal places for new housing and infrastructure.

The computer coding behind the scenes which allows a programme, game or app to run is not accessed directly, but through a general user interface. It can be thought of as a way of directing a programme, without speaking the computer's language. Without them we would all find common apps or websites such as Google far harder to use. The model is no exception and requires its own general user interface. Time and resources are required for this, but it would mean someone of any technical level would be able to use the model.

The most ambitious step for the development of this model is adapting its use for housing and infrastructure; allowing it to locate where it is best to introduce both. Since the same general information is needed as in the current model, it should be possible, however, it will take considerable time and resources. If this could be achieved then it would be a small task to analyse existing housing and infrastructure. This would mean the model could highlight areas where additional housing is most required or where infrastructure needs to be updated/installed.

Part V
Conclusion

Initially, the aim of this project was to prove that the original question posed by the SCR could be answered with graph theory; to actually answer the question seemed outside of this team's abilities. However, thanks to an ongoing dialogue between the SCR and UoS, it was possible to overcome many challenges; updating tasks and aims as it became clear what the SCR could benefit from most. As a result, the model developed throughout this project has far surpassed the initial aim. Not only is the model able to actually locate optimal sites for investment in employment in Sheffield, it is able to do it for any city within the SCR.

As well as being able to locate the best places for employment investment, this project has served as proof of concept for the wider aim; being able to tackle the large scale challenges a city faces using graph theory. With time, the model could be used to find areas where it would be best to develop new housing, or where the implication of new infrastructure could be most useful. In the long term it would even be possible to take this idea and develop it into a model which is able to analyse existing infrastructure, highlighting weaknesses and suggesting improvements.

To conclude, this model is able to find the best places to invest in employment, as well as showing that graph theory can be used to simply and accurately represent the challenges a city faces.

A Appendix A - Glossary

- Application Programming Interface (API)- A set of functions and/or procedures that allow the creation of applications which can access features or data of an operating system.
- Arc - See also; Graph theory, Node. An arc is the connection showing the relationship between two nodes within a map created through graph theory.
- Brownfield site - Real property, the expansion, use or redevelopment of which may be complicated by the presence or potential presence of harmful chemicals or pollute.
- Computational model - A mathematical model in computer science which requires considerable computational resources to study a complex system through simulations.
- Correlation - In the statistical sense, this is the interdependence of various quantities or variables.
- Database - A structured set of data held within a computer, especially one which can be accessed in several different ways.
- Dijkstra's algorithm - See also; Graph theory. A mathematical process used for finding the shortest routes between nodes within a mathematical map e.g. the shortest route between two points using a road network.
- Dimensions - A measurable extent of a particular kind, such as time, length or force.
- Entrepreneur zones - Areas in which business development is encouraged and assisted by government bodies.
- General user interface - A 'front cover' for programmes and technology which allow for a user to interact with the technology.
- GPS data - Global positioning system data, which allows the user to determine to a high accuracy their position, velocity and time.
- Graph theory - See also; Arc, Node. A mathematical technique which is often used in computational issues. By creating a 'map' showing how different items are connected relationships between different points (nodes) can be analysed.
- Greenbelt limit - Used to define and regulate urban expansion, protecting environmental areas.
- Greenfield site - Undeveloped land within a city or rural area, typically being considered for development of an urban nature.
- Infrastructure - The basic physical structures and organisations required for the operation of a society (e.g. roads, buildings, bridges, etc.).
- Input data - The initial data which is required for the operation of something. Layer - In computer science different functions are split into segments, named layers.
- LSOAs - A lower layer super output area is a geographic region often used in the collection and analysis of data.

- MSOAs - See LSOAs. Middle layer super output areas, larger in size than LSOAs. Node - See also; Graph theory, Arc. The points which represent topics or items with a mathematical map created using graph theory.
- Null hypothesis - In a statistical test this is the theory that there is no significant difference between two populations. Any difference that is observed is down to random, or statistical error.
- OAs - See LSOAs. Output areas, geographic areas smaller than LSOAs. These contain data at a high enough resolution to be considered identifying.
- Population weighted centroid - See also; OAs, LSOAs, MSOAs. A summary reference point used to represent the centre of a location with respect to the population living there.
- Prototype - A first or preliminary form of development, upon which further iterations are built.
- P-value - See also; Null hypothesis. The calculated probability of finding the more extreme results when the null hypothesis is found to be true. The definition of 'extreme' depends on how the null hypothesis is being tested.
- Resolution - See also; LSOAs. The degree to which two points can be distinguished. In this case it refers to the distinctions which can be made between individuals in a larger data set (such as an LSOA).
- Socio-economic analysis - A consideration of how economic processes affect and are shaped by social processes.
- Variable - An element, feature or factor that is subject to change.

B Appendix B - User Guide

The complete code can be found at the following repository:

https://github.com/MDRowland/UK-Public-Transport-Unemployment-Code-Repository-Python-/releases/tag/v1.0

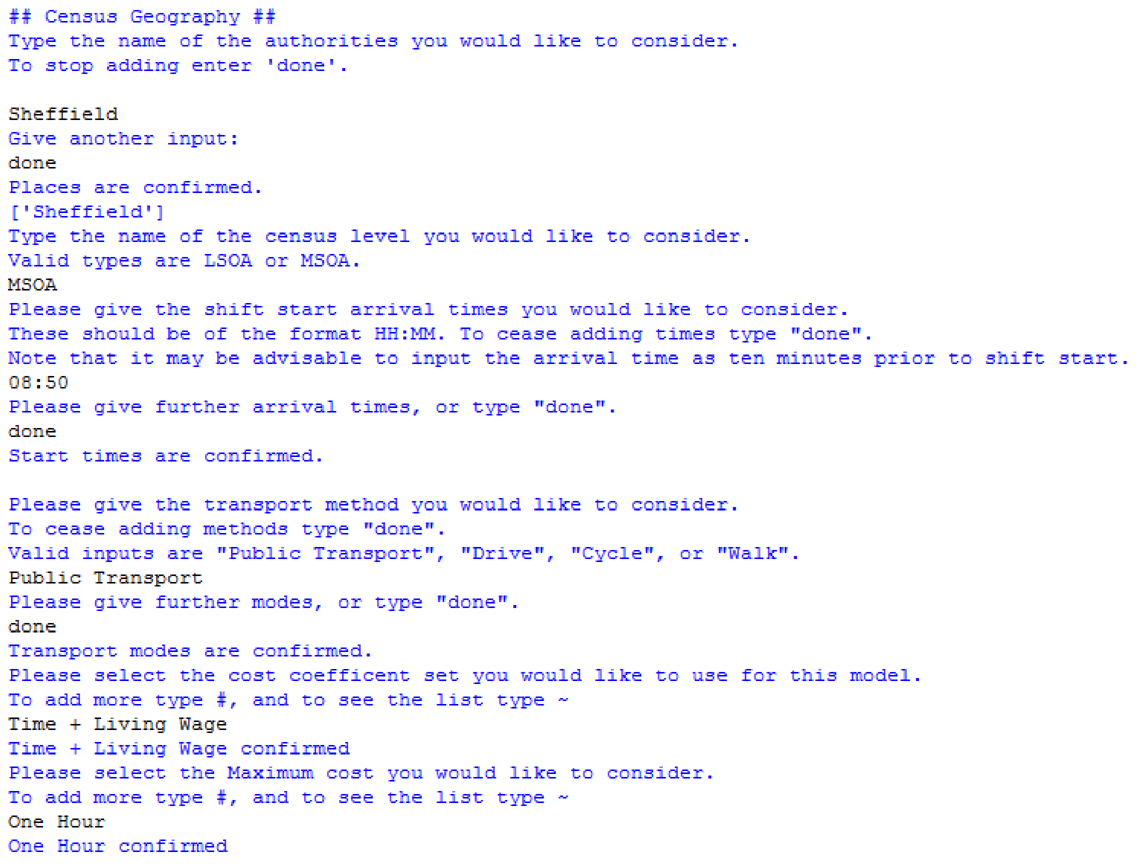
The Code Model has been arranged to be relatively easy for users unfamiliar with Python or computer coding in general. A basic understanding of computer use, and patience are the two key ingredients. The interface is type-based, with outputs given in both the shell and in files saved to sub-folders in the directory the code is in. Thus, do not place the code in a location with very limited storage, a minimum of 50Mb is recommended. As another general note, this is a prototype: as such it is not foolproof nor completely debugged, and whilst it is the ambition of the developer to ensure it is as universal as possible, do not be surprised if errors do occur.

The currently included areas are Sheffield, Barnsley, Rotherham, Doncaster, North East Derbyshire, Chesterfield, Derbyshire Dales, Bolsover and Bassetlaw. Inclusion of other areas is easily achievable with a basic understanding of Microsoft Excel. For the five excel sheets, simply add a worksheet with the name as the name of the area, and include all of the data pertaining to that area, leaving room for the headers as is visible in other sheets.

The model has been built in Python 3.6, and should not be used in previous versions of Python. At the time of writing, Python 3.7 is completing its development and will be in general use in the near future: it is not expected that this will alter the functionality of the model. For information on Python, including basic instructions on installation visit: www.python.org

Firstly, download the code from the GitHub link given above, by selecting "Clone or Download" and selecing "Download as ZIP". You may also Clone the code to your own repository, if you so desire. Then, extract the contents of the zip folder to the desired loction. If the code has not been used in its present location before, running "`Setup.py`" will initialise all the data and settings necessary to run the model. If, for any reason, sections of the storage are deleted or corrupted, or if you have made any alterations to the structure of the directory or content of the settings files that lead to a cessation of function, running this again will restore the default settings. Be advised, this will also clear any coefficient sets, weighting thresholds, or anything else that is user-defined and permanently stored. "`Setup.py`" will not allow you to reuse API access either, once the supply has been exhausted you must wait until midnight in GMT-7, the timezone of Google's servers.

"`Start.py`" will run the model in its entirety. The start script functions by asking the user for the required input information, and calling all of the other scripts with the information allowing them to perform their specialised functions in sequence. Firstly, inputs are requested, with instruction given. The Local Authority name is the first required input, then a selection of which census level you require, either LSOA or MSOA. Further options include arrival time, of which multiple are selectable, weighting coefficient and maximum acceptable weight. Also selectable is the transport mode, four options are available; Public Transport, Cycling, Walking, and Driving. These were included for completeness, and whilst multiple can be considered for one set of weighting coefficients, this may not be advisable.



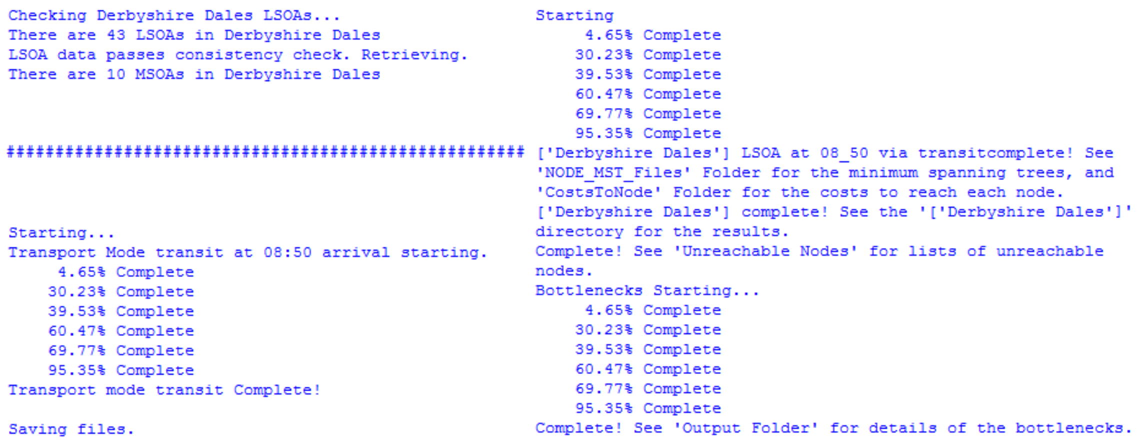
```
## Census Geography ##
Type the name of the authorities you would like to consider.
To stop adding enter 'done'.

Sheffield
Give another input:
done
Places are confirmed.
['Sheffield']
Type the name of the census level you would like to consider.
Valid types are LSOA or MSOA.
MSOA
Please give the shift start arrival times you would like to consider.
These should be of the format HH:MM. To cease adding times type "done".
Note that it may be advisable to input the arrival time as ten minutes prior to shift start.
08:50
Please give further arrival times, or type "done".
done
Start times are confirmed.

Please give the transport method you would like to consider.
To cease adding methods type "done".
Valid inputs are "Public Transport", "Drive", "Cycle", or "Walk".
Public Transport
Please give further modes, or type "done".
done
Transport modes are confirmed.
Please select the cost coefficent set you would like to use for this model.
To add more type #, and to see the list type ~
Time + Living Wage
Time + Living Wage confirmed
Please select the Maximum cost you would like to consider.
To add more type #, and to see the list type ~
One Hour
One Hour confirmed
```

Figure 12: An example of the interface, all fields filled.

This will then create various files. Most useful to a general user will be the "`PopCosts`" files- the mean cost per unemployed head weights to reach each locality, the "`BOTTLE`" files- containing the bottlenecking metric for each node, for the variables given. These are all found in the "`Output Files`" folder and sub-directories.



```
Checking Derbyshire Dales LSOAs...
There are 43 LSOAs in Derbyshire Dales
LSOA data passes consistency check. Retrieving.
There are 10 MSOAs in Derbyshire Dales



#####################################################


Starting...
Transport Mode transit at 08:50 arrival starting.
     4.65% Complete
    30.23% Complete
    39.53% Complete
    60.47% Complete
    69.77% Complete
    95.35% Complete
Transport mode transit Complete!

Saving files.
```

```
Starting
     4.65% Complete
    30.23% Complete
    39.53% Complete
    60.47% Complete
    69.77% Complete
    95.35% Complete
['Derbyshire Dales'] LSOA at 08_50 via transitcomplete! See
'NODE_MST_Files' Folder for the minimum spanning trees, and
'CostsToNode' Folder for the costs to reach each node.
['Derbyshire Dales'] complete! See the '['Derbyshire Dales']'
directory for the results.
Complete! See 'Unreachable Nodes' for lists of unreachable
nodes.
Bottlenecks Starting...
     4.65% Complete
    30.23% Complete
    39.53% Complete
    60.47% Complete
    69.77% Complete
    95.35% Complete
Complete! See 'Output Folder' for details of the bottlenecks.
```

Figure 13: A standard format output to the user from a set of input information

For the graphical output, running the file “`GraphsOutput.py`” makes plots of most variables, including all those used in this report. A more advanced user may be able to alter some of the numerical parameters, as is documented in the code appendix, although the present ones should serve adequately for most purposes.

C Appendix C - Code Discussion

C.I The Input Data

There are five key data sources used in this model. They are all stored in .xlsx format in the "Excel Sheets" folder. In every workbook, the data of interest has been separated manually into sheets named by the local area authority they belong to. This is done to reduce the computing time required to get data and make it possible for a user to manually replicate the simple structure in the files for other areas outside the SCR+ region used in the data.

C.I.i Workplace Data

This data concerns the jobs in an MSOA, with associated name and MSOA code, broken down into job census group, with a total also provided. This is used in conjunction with the LSOA totals data to find the target jobs in each LSOA/MSOA. For the MSOAs, this is just the actual figures of the jobs in those locations. The LSOAs are projected based on the total jobs in the LSOA as a fraction of the totals in the MSOA. This data is found in `SCR_LSOA_Workplace_ Totals.xlsx` and `SCR_MSOA_Workplace_Data_Split.xlsx` , respectively. Whilst this is not necessarily a perfect method, it is simple and not an unreasonable approximation.

C.I.ii Population Data

The data in `SCR_LSOA_Estimates.xlsx` is a breakdown of population by LSOA and also by age group- but the latter is not used in our model. More advanced models for various purposes of course could take advantage of the availability of this data.

C.I.iii Deprivation and centroid data

The file `SCR_GeoCode_DeprivationFraction.xlsx` contains the "Income rate", a measure of how deprived of income the population is, the unemployment rate, and the latitude and longitude of the population weighted centroid of the LSOA.

C.I.iv Conversion data

The file `SCR_LSOAConversions.xlsx` contains a list of every single output area (the smallest census area) and which LSOAs, MSOAs, and Local Authorities they are situated in. This is used to convert between LSOAs and MSOAs in this model; however, if data for the other factors used in this model can be used at the Output Area level, this could also be used to obtain a far better model of the transport network and unemployment.

C.II DataExtractor

The modules `openpyxl` and the `os` module are used in this script. `openpyxl` is an open-source python module that can be used for reading and writing Excel spreadsheets. All the Excel sheets are used in this script, documented above. After opening these sheets, and setting associated variables, between lines 38 and 62 the script checks for the existence of various data and sets the tags `NeedLSOA` and `NeedMSOA` to `True` or `False`, which triages the necessity of creating this locality data for later in the script. The numbers of LSOAs and MSOAs for the currently considered area are pulled from a previously created and stored dictionary, which is printed in the console to the user along with information regarding the existence and completeness of the data. Completeness is assessed as whether the file length of the area files matches the number of those areas. Whilst this does leave some obvious room for error, it is adequate providing this model is used as intended; as such this is the standard method used throughout this model. If the data is correct, in the LSOA case the data is extracted from file as it is needed to build the MSOA data. If it does not exist, or it is incorrect, then it must be created.

Between line 76 and 124, the LSOA data is created. For every line of data in the appropriate worksheet in `SCR_GeoCode_DeprivationFraction.xlsx`, the LSOA code, the name of the MSOA it is situated in, and the coordinates are extracted. Then, for `SCR_LSOA_Estimates.xlsx`, `SCR_LSOA_Workplace_Totals.xlsx`, and `SCR_MSOA_Workplace_Data_Split.xlsx` every line is looped over until the current LSOA or MSOA name is found. The looping is necessary as the data is not necessarily in the same order, depending on how the files have been edited or formed. From these sheets, the appropriate row, the total number of jobs in an LSOA, and the target and total jobs in the MSOA, respectively are found. Then this data is written to the output file and added to the array in the following format:

```
[LSOA Code, Total Population, Unemployment fraction, LSOA Longitude, LSOA
                   Latitude, Target Job Number]
```

Target jobs are projected from the MSOA the LSOA sits in as this data is not available at this level. This is done by multiplying the total number of jobs in the LSOA by the fraction of jobs that are target jobs in the MSOA. Then metadata is created for this in the same folder, consisting of a file with a single line, documenting what each space-separated variable is. Whilst this has not been incorporated into the code any further, it could be if so desired. As before, the triaging system is employed to determine whether MSOA data is needed, with the additional criterion that unless Node Type has been set to MSOA, the MSOA data will not be found. Again, each LSOA is looped over in the previously generated array: and for each by looping over the `SCR_LSOAConversions.xlsx` file, the corresponding MSOA is found. If this is the first time it has been found, the array is initialised with the data in the same format as above. If it is not, the population is added to the currently known population, and the new unemployment fraction is found by re-weighting according to the new data. New population weighted centroids are also found by moving the current population weighted centroid according to the population in the MSOA so far and the LSOA being added. Target jobs are added the first time the MSOA is generated as we have this data at this level. This data is then saved to file.

C.III Arc Finder

The module googlemaps is key here: this module gives access to the Google Maps API. Other requisite modules are `time`, `sys`, `datetime`, `os`, and `re`, all of which are included in most standard Python distributions. Firstly, the script generates a "`Nodes`" array containing the data from all desired areas at the desired census level as obtained in the Data Extractor. Then the arrival times specified by the user are converted into Unix Time for the coming Monday. Whilst a user having more power to specify the day of the week, ultimately this was decided upon for reasons of simplicity. A tuple with multiple API access codes is defined, and the "`z.txt`" programme file is called to determine which, if any, of the codes are currently useable, as these codes are free and allow 3000 daily server requests. Please note that this is changing, and as such this system may become unnecessary.

Using a `try:  except:  finally:` structure the script loops over each arrival time and mode of transport specified, tests for the existence of data by the standard method already described. The script then takes the first node in the list and finds the connectivity information to every other node using the population weighted centroids. Then the second node is considered, and the connectivity to every other node except the first is established. The first is not tested for as this assumes that each arc is a two-way street. The code proceeds in this manner for the remaining nodes, once a node has been the "origin" it is no longer considered as a destination. This is to save time and computing power for large sets. Thus, the total number of arcs is given by:

$$TotalArcs = \frac{1}{2} * Total\,Nodes * (Total\,Nodes - 1)$$

The information extracted from the returned list of dictionaries is the total time taken and distance travelled, the number of public transport changes, and for Public Transport, the time and distance walked. Data is then output to file, with the Arc name being the two node codes, without a space separating them. Each line in the Arcs output file is then of the form:

```
[Arc Name, Origin Code, Destination Code, Total Time, Total Distance, Number
                  of changes, Time Walked, Distance Walked]
```

Line 102 prevents exceeding the daily limits on use for each code by checking the number of uses it has had thus far, and line 116 ensures that a code that does not exist is not called. The `except:` clauses control two of the more common errors and give the user an advisory output. The always executed `finally:` clause closes open files and updates `z.txt` in the programme files.

C.IV Dijkstra's Algorithm

This script requires `sys` and `NumPy`, standard modules of which basic understanding will be assumed. Firstly, the function that performs the graphical algorithm is defined. After values are initialised, the current best arc is set to a null arc of infinite length. Then all unused arcs are looped over, and those which connect to one and only one node that has already been connected to are tested against first the null arc and then the lowest cost arc. Cost is assessed via the user-defined coefficients, which are multiplied by their respective variables of `Total Time, Total Distance, Number of changes, Time Walked, Distance Walked` as found earlier. The cheapest arc is then added to the minimum spanning tree, the node it is connected to is added to the list of accessed nodes, and the direction travelled along the arc stored in another list. Direction is defined as +1 when moving from the Origin to the Destination as defined in the Arc Finder, and -1 for the other way about. For each node, the cost to get from the "Source" node to every other node is stored in a file by the name of the source node. This function is looped over for every node in the considered area/s, and as such a minimum spanning tree (MST) and a cost to node recording is made for every single one, as many will be unique. The Dijkstra function relies on the node document only for the total number of nodes. This means that if nodes that aren't listed in the nodes file are connected to by arcs or nodes listed are not, the programme will be unaware and thus potentially stop short of a complete MST. This should not happen if the model is used as intended. Finally, an output is printed informing the user of the completion of the current time and Transport method, as this script is called from start for these variables.

C.V Population Costfinder

This script is very simple, it first loops over all known nodes to calculate the number of unemployed people at those locations, and then finds the mean and standard deviation of cost to get to a node per head of unemployed population from the Cost To Node files previously calculated. Then outputted is a single file containing the code of every node and the mean cost to reach it from all others, and the standard deviation in that.

C.VI Unreachable Nodes

This script uses the cost to node files for each node, created by the Dijkstra's algorithm script. It then runs down each node in this file and if the cost to reach it is greater than the acceptable user defined cost, then it is added to the unreachable nodes list, which is printed out into an output file for each source node.

C.VII Bottle Neck Finding

Firstly the script creates a map of each minimum spanning tree, consisting of two lists. For a node in the "Initial" list, the node with the same index in "`Translt`" is the node one closer to the source node. Thus, the direction of travel down the branches of the Minimum spanning tree is established, and later in the script this allows one node to yield another. The source node yields itself.

Then, the script starts at an "unreachable node", as defined by the script above. It then moves down the branches toward the source. The first reachable node it encounters is marked as a bottleneck. Using a tally list and a node code list, the number of times each node acts as a bottleneck is recorded. This is done for every node in every unreachable nodes file for every node as a source, by looping over the Nodes tuple, present in the active memory as it is generated in start.py.

C.VIII Start

After retrieving the LSOA and MSOA dictionaries from the files they are kept in, the script requests all user defined inputs. Firstly, the Local Authorities required are tested, and checked for validity against the names of the LSOA dictionary. Multiple can be selected, and selection is ended by typing "done".

Next, the Node Type, either LSOA or MSOA is selected. Only one can be chosen from here, this was done for simplicity. Then the times of arrival are requested. The input is checked for the correct HH:MM format, and if correct added to the tuple. Again, multiple can be selected, and selection is ended by typing "done".

Transport modes are selected in an analogous manner. The user is prompted and given four options. These options are translated by a dictionary into the string that the goooglemaps module expects for each method. This string is then added to the tuple, ensuring that methods are not added twice. Inputting a lowercase p will also prompt the script to use public transport, this was added for ease of testing.

Various user defined parameters are next selected. As they all work in a very similar manner, only the first will be explained. The user is prompted to pick a cost coefficient set and offered the ability to create new ones or view the list of existing coefficient sets. Selecting creation of another runs the "`Cost_Coefficient Creator.py"` script. Selecting to view the list uses the tabulate table generation module to list the available sets with names and values. Otherwise, inputting a valid name selects that cost coefficient set. A wrong input prompts the user to try again.

Then, the subscript "`DataExtractorFinal.py`" is run in a loop over each Local Authority, to find the nodes in the requested areas. After this is complete, the internet connection is checked, and then "`FinalArcfinder.py`" is run. This connection check is run to prevent a failure of the algorithm because of a bad connection, and so that this can be made clear to the user as an issue.

The final non-calling operation of the script is performed: the Nodes tuple is generated by extracting all nodes from the desired areas from their files to be used in the subsequently called scripts.

Lastly, the following scripts are called for each transport method and arrival time, in the order listed

- `djikstra_test.py`
- `PopulationCostfinder.py`
- `Unreachable_Nodes.py`
- `Bottleneck_check.py`

C.IX Cost Coefficient Generators

As all three of the coefficient generators work in an analogous manner, it is adequate to document the largest Cost_Coefficient Creator.py. Firstly, the currently stored coefficients are opened, and converted to dictionary format. Then, in an infinite while loop asks for an input, with the following results:

- "#"`yields the currently saved names of the coefficient sets in a list format, by looping a print command over the list of names obtained from the dictionary.`
- `Tilda yields a comprehensive table of the coefficients using the tabulate module as above.`
- "*"`exits the programme`
- `Any other string input is registered as a name for a new coefficient set.`

Following on from this input, the script requests numbers for the values the set requires. These must be numbers, which is checked for, but can have any value. This, if correct, is added to the dictionary to file and the option to create another set is offered. Finally, the dictionary is saved to the file it was derived from and everything closed.

C.X Setup

The Setup script is designed to be used on first download; instructions for this are available in Appendix B. The script uses an if: else: structure to create the folder structure needed with os.makedirs. Then, if they do not exist, each of the Programme files are created in turn with their default settings. Subsequently, the Excel Sheets are moved into the correct folder, and scripts run to find the columns that the target jobs sit in, and the numbers of LSOAs and MSOAs. Finally, files are saved and output messages given.