

Brunel University London
Department of Mechanical, Aerospace and Civil Engineering
College of Engineering, Design and Physical Sciences

Modelling Railway System Capacity with Statistical Analysis, Machine Learning and Discrete-Event Simulation

by
Harry J. Munro

Supervisor: Dr Ali Mousavi

July 2017

Dissertation submitted in partial fulfilment of a
Masters of Science Degree in Engineering Management

1 Abstract

Forecasting dwell times accurately is important for modelling the capacity of current and future transport systems. This activity is a key enabler for achieving the optimal allocation of investment into transport systems in order to deliver appropriate capacity improvements.

This study analyses data collected from the London Underground in order to identify a number of factors that influence the dwell time for through running stations. The most significant factor was shown to be the total number of passengers boarding and alighting. A number of factors weren't included such as the degree of automation, station, signalling and platform to train interface characteristics.

Evidence has been presented to show that a K-nearest neighbours regression algorithm most accurately models the mean dwell time, given the factors included in the study.

New evidence has been presented to show that the dwell time can be best estimated by sampling from a gamma distribution. This distribution can be employed in discrete-event simulation in order to assess the capacity of a railway system.

A case study of the London Underground's Victoria Line was modelled using discrete-event simulation implemented in Python. The achievable capacity of this line was predicted up to 2050 assuming a linear relationship between the number of boarders and alighters and the population of London. Losses in achievable capacity were demonstrated as the number of boarders and alighters were increased. A significant limitation of this experiment was the assumption that the number of passengers on the Victoria Line increases linearly with London's population.

This thesis also demonstrates that assessing capacity does not require significant investment into off-the-shelf software tools, but that satisfactory results can be achieved through open source programming. This has implications for how organisations responsible for modelling capacity develop this technical competency. This is of importance to organisations looking to improve efficiency and achieve cost-effective modelling.

Further work should seek to improve on the comprehensiveness of the dataset, introducing a greater number of factors and data from different railway systems around the world. This will lead to more robust models and greater dwell time forecasting accuracy. Data on door open and door close cycle times will lead to more accurate dwell time models as well as more precise measures of passenger loading. It would be beneficial to have a central and easily accessible repository of building information management data, including station and platform to train interface data, which would significantly improve the ability to model these factors. This would improve the forecasting power of a dwell time model.

2 Acknowledgments

I would like to thank my parents, Prof Neil and Dr Bernadette Munro, for their massive support and encouragement.

Thank you to my girlfriend, Grace Owens, who has offered tireless support and has put up with my endless observations about the tube while we travel around London together.

Thank you to Dr Ali Mousavi for reviewing this thesis before submission and for providing the inspiration for a significant part of it through his course on discrete-event simulation.

I am also incredibly grateful to my colleagues Gabriel Smith and Simon Chung at Transport for London for supporting me throughout the writing of this thesis, and to the team at Systems Performance Engineering for putting up with my endless study days out of the office.

3 Glossary

Abbreviation	Description
TPH	Trains per hour
RORI	Run-out run-in time
RORIF	Run-out run-in time at full-speed
SAF	Service affecting failure
LU	London Underground
TfL	Transport for London
RODS	Rolling origin and destination surveys
CUPID	Contract performance information database
LCH	Lost customer hours
NETMIS	Network management information system

4 Table of Contents

1 Abstract	1
2 Acknowledgments	2
3 Glossary	3
4 Table of Contents	4
3 Table of Figures	6
4 Table of Tables	7
5 Author's Declaration	8
3 Statement of Aim, Objectives, Methodology and Expected Outcome	9
4 Introduction	11
4.1 Rationale - why the investigation is important	11
4.2 Scope of the investigation	12
4.3 Problem with modelling large scale systems.	14
4.4 Hypothesis and/or key questions and why they are important	14
5. Literature Review	15
5.1 Dwell Time Literature	15
5.2 Simulation Literature	17
6. Methodology	21
6.1 Dwell Time Modelling	21
6.1.1 Data Sources	21
6.1.1.1 Dwell Times from NETMIS	21
6.1.1.2 Delays on the Tube from CUPID	23
6.1.1.3 Passenger Data from RODS	28
6.1.1.4 Rolling Stock Technical Specifications	30
6.1.2 Distribution Analysis Methodology	32
6.1.2.1 Parameter Estimation	32
6.1.2.2 Scenarios to Test	35
6.1.2.3 Analysing the Data	36
6.1.3 Factor Analysis Methodology	36
6.1.3.1 Pre-Processing Data	36
6.1.3.1.1 Matching NETMIS with RODS	36
6.1.3.1.2 Filtering Data Points	37
6.1.3.1.3 Resampling Data	39
6.1.4 Regression Analysis Methodology	41
6.1.4.1 Multiple Linear Regression Discussion	41

6.1.4.2 Decision Tree Regression Discussion	41
6.1.4.3 K-Nearest Neighbours Discussion	42
6.1.4.4 Neural Network Discussion	43
6.1.4.5 Validation	44
6.2 Capacity Simulation Case Study	44
6.2.1 Model Qualification and Assumptions	45
6.2.2 System Diagram	49
6.2.3 Operational Recovery Time	51
6.2.4 London Population Growth	52
7 Results	56
7.1 Dwell Time Modelling Results	56
7.1.1 Distribution Analysis Results	56
7.1.2 Factor Analysis Results	60
7.1.2.1 Visualising the Data for Factor Analysis	60
7.1.2.2 Correlation Results for Factor Selection	68
7.1.3 Regression Analysis Results	68
7.1.3.1 Multiple Linear Regression	68
7.1.3.2 Decision Tree Regression	69
7.1.3.3 K-nearest Neighbours Regression	71
7.1.3.4 Neural Network Regression	74
7.1.3.5 Regression Results Summary	77
7.2 Victoria Line Simulation Case Study Results	77
8 Discussion	81
8.1 Dwell Time Modelling Analysis	81
8.1.1 Distribution Analysis	81
8.1.2 Factor Analysis	81
8.1.3 Regression Analysis	82
8.2 Victoria Line Case Study Analysis	84
9 Conclusions and Recommendations	86
10 References	89
11 Appendices	92
11.1 Appendix A - Code Used for Processing	92
11.1.1 Script for Merging raw NETMIS and RODS datasets	92
11.1.2 Script to Resample the Merged Dataset	98
11.1.3 Script to Test Goodness of Fit for Multiple Distributions	101
11.1.4 Regression Models Scripting	104
11.1.5 Victoria Line Discrete Event Simulation	107
11.2 Appendix B - Distribution Analysis Results	113
11.3 Appendix C - Sample of Merged Dataset	116
11.4 Appendix D - Sample of Resampled Dataset	118

3 Table of Figures

Figure	Page	Description
1	16	Conceptual model of factors that influence dwell time
2	19	Different categories of simulation
3	20	Practical capacity versus desirable reliability
4	26	The relationship between the service affecting failure rate and line availability for London Underground lines
5	26	Good service on all lines on the London Underground
6	29	Number of boarders and alighters on the London Underground at Holborn station
7	30	The dependency of mean dwell time on the number of boarders and alighters
8	38	Correlation coefficient of total boarders and alighters to dwell time
9	40	The effect of resample period on correlation strength
10	42	Decision tree model overfitting
11	43	Uniform versus inverse weights
12	46	Gamma distribution representing dwell time
13	47	Dwell time sampling process
14	49	Simulation logic
15	50	Queueing logic for resource requests
16	51	Victoria line system logic
17	54	London population forecast
18	60	The distribution of resampled mean dwell times in the dataset
19	61	Distribution of resampled mean boarders and alighters
20	62	The dependency of dwell time on total boarders and alighters
21	62	The distribution of service affecting failure (SAF) rates
22	63	The dependency of mean dwell time on SAF rate
23	64	The distribution of standing capacities of different rolling stock in the dataset
24	64	The dependency of mean dwell time on standing capacity
25	65	The distribution of seating capacities of rolling stock in the dataset
26	66	The dependency of dwell time on seating capacity
27	66	The distribution of train density statistics
28	67	The dependency of dwell time on train density
29	69	The dependency of R-squared and validation correlation on decision tree depth
30	72	Uniform vs inverse distance K-nearest neighbour models based on the number of neighbours

31	74	The dependency of validation correlation and r-squared on neural network hidden layer size
32	78	Predicted Victoria Line achievable capacity as London population increases
33	79	Analysis of capacity constraints at each station on Victoria Line in 2015
34	80	Analysis of capacity constraints at each station on Victoria Line in 2050

4 Table of Tables

Table	Page	Description
1	22	Typical dwell time data
2	23	Resampled dwell times
3	24	Labelling of line IDs and direction codes
4	25	Data on service affecting failures on the London Underground
5	27	Train service affecting failure rates of London Underground lines
6	31	The calculation of train density statistic
7	31	Rolling stock seating and standing capacities
8	32	Distribution descriptions
9	39	Correlation between mean dwell time and total boarders of alighters for different dwell time resample periods
10	47	RORIF and peak boarders and alighters for Victoria Line
11	53	London population forecast
12	58	Top five distributions to use for the dwell time
13	57	Distributions tested for goodness of fit for the dwell time
14	68	Pearson and spearman correlation coefficient results for different factors
15	69	Decision tree testing results
16	72	Uniform vs inverse distance K-nearest neighbour models based on the number of neighbours
17	75	The dependency of validation correlation and r-squared on neural network hidden layer size
18	77	Summary results of regression models tested
19	113	Distribution analysis results
20	116	30 row sample of the merged NETMIS and RODS dataset
21	118	30 row sample of resampled data

5 Author's Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted or otherwise incorporated material which is the work of others, I have included the source in the references. Any views expressed in the dissertation, other than referenced material, are those of the author.

SIGNED: DATE:

3 Statement of Aim, Objectives, Methodology and Expected Outcome

The aim of this study is to identify factors that influence the dwell time on the London Underground and to produce a model which can be used to predict the dwell time. A common distribution which the dwell time follows is aimed to be identified. Additionally this study aims to demonstrate how dwell time modelling can be applied to capacity modeling through discrete event simulation.

The objectives of this study are to:

- Test the hypothesis that dwell times tend to follow a specific distribution.
- Review of factors which influence dwell times.
- Design and build a dataset using available London Underground data and any other available relevant data.
- Test the correlation of shortlisted factors to the dwell time.
- Develop a model to estimate the distribution parameters of the dwell time.
- Develop a discrete event simulation model which implements the dwell time model for capacity analysis of a railway system.

The methodology for this involves:

- The creation of a script to analyse the distribution of dwell times using maximum likelihood estimation of the parameters followed by calculation of the coefficient of determination of the estimated distribution against the real world data as a means of assessing goodness of fit. This method enabled the testing of 84 different probability distributions against the test data.
- Scripts were developed to merge and analyse different London Underground datasets in order to carry out a statistical analysis to identify the principal factors that influence the dwell time.
- Machine learning regression models were identified and tested in order to create a model for the dwell time.
- A discrete event simulation model, implemented in Python using the SimPy library, was used to produce a simulation of a railway, demonstrating the implementation of the machine learning dwell time model.

The expected outcomes were as follows:

- It was expected that dwell times would tend to favour a specific distribution and that this distribution could then be used in simulation studies.
- It was expected that a number of factors would influence the dwell time, in particular the number of passengers boarding and alighting trains was expected to be a primary driver for the dwell time.
- It was expected that a model could be produced to forecast dwell times given the identified factors.
- It was expected that it was then possible to implement this new model in a discrete-event simulation and use this to predict capacity into the future for a railway line.

4 Introduction

“If you do not know how to ask the right question, you discover nothing.” - W. E. Deming

4.1 Rationale - why the investigation is important

For London, increasing transport capacity to match increased levels of crowding is vital to ensure that economic productivity is not negatively affected, which would then have secondary effects on the wider UK economy (Greater London Authority, 2009).

The capacity of a railway is largely dependent on the frequency at which trains can pass through bottlenecks within the system. Constraints on train frequency are essentially determined by three factors: the run-in run-out time (RORIT), the dwell time and the operational parameters of the site. The RORIT time is, in turn, determined by the signalling system, track and train parameters. The dwell time, involving a significant human factors component, can be constrained by a number of factors including passenger behaviour and platform to train interfaces. Operational parameters, such as the routing of trains and the changing of drivers, are also significant parameters contributing to the limiting of capacity.

The ultimate capacity of a system is generally limited by one region known as the bottleneck (Goodwin 2015). The railway system will not be able to achieve greater capacities on a given line than that can be achieved at the bottleneck. There are parallels between this theory of bottlenecks and the management paradigm *The Theory of Constraints* (TOC). TOC states that there is always one constraint on a business' performance, and that the key to unlocking progress is to eliminate the constraint. Once a constraint is removed a new one will appear, hence the need for a continuous improvement philosophy and the on-going removal of constraints (Ox and Goldratt 1986).

From the railway operator and procurer perspective, railway performance modelling and simulation is an essential activity for both the design of requirements for future systems as well as the verification of suppliers' systems. From the supplier's perspective railway performance modelling is essential to understand whether the suppliers' systems will meet the requirements of the customer. Incorrect forecasting of dwell times causes problems for train service operators, since many features of the system such as the timetable, the planned number of trains and their performance parameters are designed around forecasts of the dwell time. Inaccurate modelling will incur not only

programme rework and wastage of resource but also loss of confidence in modelling results and reduced trust of modelling predictions.

A dwell time, or run-out run-in time (RORIT), saving of one second on average on London Underground's Victoria Line equates to economic savings of £917,000 per year (Goodwin 2015). Transport for London is spending £20billion on capital expenditure investment between 2015 and 2021, with the three largest programmes on the London Underground all targeting an increase in capacity (TfL 2015). The majority of this spend is on decreasing the RORIT component of capacity through upgrading rolling stock and signalling system technology, which enables trains to run closer together safely. There is therefore significant benefit in understanding the dwell time and making efforts to increase capacity - while ensuring that efforts are both realistic and achievable.

Systems engineering can be defined as "the creation and monitoring of requirements" (Tortorella 2015). Computer simulation can be used to design and optimise these requirements at low cost. There has been a long and extensive history of using deterministic computer simulation at the London Underground in order to design and validate engineering requirements, primarily using railway engineering simulator (RES) (Rail Engineer, 2013). With the advent of increasing computer power and the ready availability of open source discrete event simulation methods such as Simpy (Simpy, 2016), as well a continuous development of RES, it is viable to run stochastic computer simulations using the Monte-Carlo method. Stochastic simulations use random number generators to sample data with a degree of inherent uncertainty. The dwell time is an example of one such type of data, which is considered random in a stochastic simulation. One distinct advantage of stochastic simulation over deterministic simulation in railway systems engineering is the ability to model reliability. Reliability is seen as a significant component of service quality at the London Underground. Between 2011 and 2016, for instance, a reliability enhancement programme reduced lost customer hours on the tube by 38% (TfL 2016). This is an important area in system design as is the ability to predict the capacity of complex systems such as junctions, termini and depots, which are difficult and labour intensive to model deterministically.

4.2 Scope of the investigation

The optimal method for estimating the capacity of a bottleneck is to be able to assess the run in and run out time (RORIT), the dwell time and (for complex bottlenecks such as junctions) the signalling logic. Technology exists to accurately calculate RORITs. For example London Underground uses an in-house simulator - Railway Engineering Simulator (RES) - which is able to accurately simulate train

system performance and factors in train, signalling and energy characteristics (Engineer 2013). However there is currently no accurate method available for calculating a dwell time. Being able to accurately forecast dwell times at key bottlenecks will be a key enabler in being able to predict line-wide capacity. This facility is essential in order to optimise design requirements and verify performance of suppliers, ultimately resulting in a more cost effective deployment of resources.

For example, a metro line might be running 24 trains per hour, which is constrained only by a bottleneck at a single location. The peak capacity of this bottleneck is 24 trains per hour, however all of the other locations on the line when considered individually have a peak capacity of 28 trains per hour. If the constraint at the one bottleneck location can be lifted to 28 trains per hour, then the whole line will be lifted to 28 trains per hour. If work is carried out at non-bottleneck locations to improve their capacity, then no capacity improvement line-wide is actually delivered so the investment yields no return.

There is also the problem of model accuracy in industry. All models require a degree of abstraction, but too much abstraction can result in the signalling logic of the system not being effectively captured, thus a principal constraint of the bottleneck may not be adequately assessed. Monte Carlo methods and discrete event simulation can help solve the problem of assessing complex systems, but computing power is a fundamental limitation to modelling complexity, as well as the cost of building models and the time available to those doing the modelling. Thus, there exists a need for models which are valid, which require a degree of quality that includes signalling logic, dwells and RORITs - but which are also simple enough to be practical given time, cost and computing power resource constraints.

This investigation will thus begin with a comprehensive study of dwell times on the London Underground. Based on available data, the principle factors which affect dwell times will be established. The objective is to demonstrate the creation of a model from these parameters which can be used to predict the mean dwell time.

The distribution of the dwell time will also be investigated. The objective is to see if there is any variation in distribution shape between different sites and which distribution can be picked to reasonably estimate dwell times.

Following the investigation into dwell times, a case study will be demonstrated combining the dwell time modelling with a simulation approach to assessing capacity.

4.3 Problem with modelling large scale systems.

Train movement is not simple to model even when a single train is considered. It is even more challenging to model the interaction between different trains on the same line. A continuous simulation approach which models trains as agents that update states in relation to other states and agents is needed.

As previously mentioned, London Underground uses Railway Engineering Simulator (RES) to calculate inter-station run-times. This simulation can be specified to take discrete time steps as low as is required, however usually one second is used. As a result of the simulation needing to update its state at this level of frequency, the possibility of monte-carlo simulation becomes an impractical with current computing power.

Discrete-event simulation serves as an effective tool to model complex systems in a monte-carlo fashion, for which continuous simulations may struggle.

4.4 Hypothesis and/or key questions and why they are important

There are a number of key questions that this thesis will attempt to answer. Firstly, it is hypothesised that a model can be created for predicting the dwell time distribution at through running stations. There may be a number of factors that contribute to this. These factors will be investigated based on available London Underground data and a model will be developed for prediction purposes. Secondly, a case study stochastic simulation will be produced to forecast future capacity on a London Underground line.

5. Literature Review

5.1 Dwell Time Literature

At the London Underground there are several definitions of dwell time. One definition is wheel stop to wheel start. However dwell time can also refer to door open to door close, and also the time available for passengers to board the train (Wong and Key 2014). For the purposes of this study, dwell time will be defined as the time from wheel stop to wheel start where passengers are boarding and alighting as extensive data is available on the London Underground estate for this purpose.

Boarding and alighting rates have had significant research focus in recent years. A report commissioned by London Underground Limited tested the boarding and alighting rates of a new concept deep tube train in a mock-up laboratory setting with volunteers acting as passengers (Holloway, Roan, and Tyler 2013). This report found passenger boarding rates had some casual association with door width and stand-back distance (the distance passengers are allowed to stand from the edge of the platform).

Another report prepared for London Underground Limited identifies seven significant passenger, train and platform related variables which have an effect on dwell time, quantifies these factors and proposes a model to predict passenger boarding and alighting rates (Community of Metros and Imperial College London 2013). The study had some success in validating its model for alighting rates against laboratory data, but was less successful validating the models predicted boarding rates against laboratory data. A key limitation of this study was the lack of validation against real-world data.

A significant limitation of all the studies on boarding and alighting rates is that researchers treat either the dwell time or the rate of passenger flow as fixed variables that can be stated in a deterministic manner. In practice the dwell time is a random variable. Modern computing power has unlocked the ability to more easily carry out stochastic monte-carlo simulations, for which understanding the distribution of random variables such as dwell times is key to their accuracy.

However boarding and alighting rates do not present the complete story. In practice, transport planners often use dwell time forecasts based on estimated boarding and alighting rates as well as forecasts of passenger numbers. This does not paint the entire picture, since there are additional factors determining the dwell time in addition to the actual rate of boarding and alighting. A recent

report described dwell time management as the “next frontier in terms of increasing capacity”(Community of Metros and Imperial College London 2015).

One Dutch study managed to estimate dwell times in real time with high levels of accuracy - predicting the dwell time in a case study from 85.8% to 88.5% during peak times and 80.1% during off-peak times. This study subjected a database of short stop times on a Netherlands’ network to statistical analysis. The study suggests that dwell times (defined by wheel stop to wheel start) should be separated into peak and off-peak. Passenger numbers were excluded from the analysis. A non-parametric k-nearest neighbours model was used with the factors train length, dwell times at previous stops and dwell times of preceding trains (Li, Daamen, and Goverde 2016). This same study contains a thorough review of previous studies which have attempted to estimate dwell times and the data.

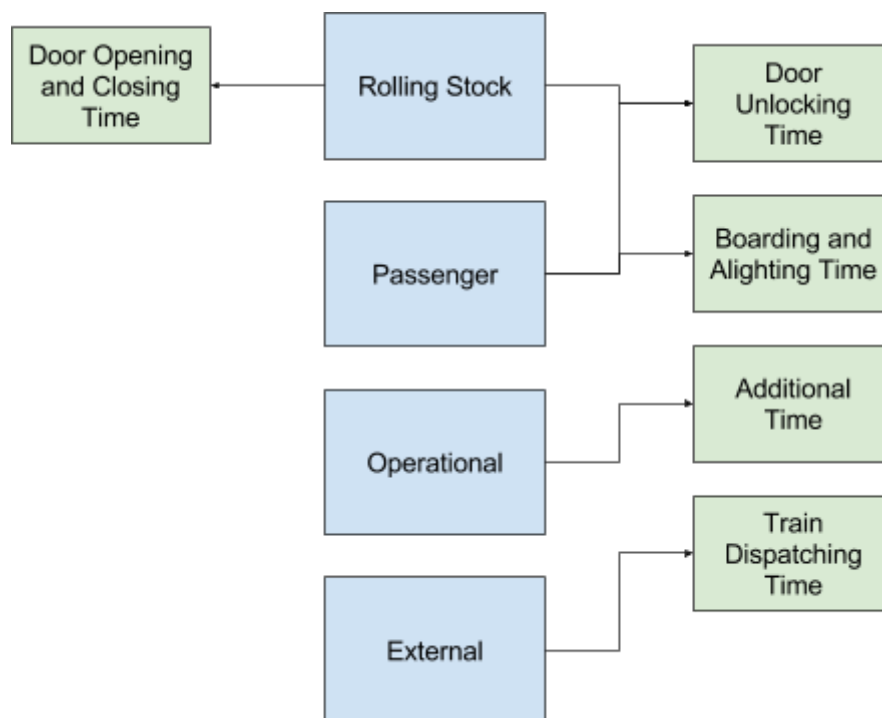


Figure 1 – Conceptual Model of Factors that Influence Dwell Time (Dewei, Winnie, & Rob, 2016)

Based on the literature reviewed, there are currently no published studies investigating the distribution of the dwell time, considering it as a random variable. The distribution of the dwell time is important from a modelling perspective. In order to create models that better represent the real world the dwell time needs to be sampled from a probability distribution. This could be because

there is a focus on making studies useful for operations, thus the need to precisely and deterministically predict the dwell in order to provide live predictions. From a modeller's perspective, considering the dwell as a random variable and being able to predict the parameters that describe this distribution is useful.

There are also a number of key factors which have been identified as under reported. These include the influence of platform edge doors (PEDs), platform layout (e.g. number of exits), platform width and the effectiveness of station assistant train services (SATS) on reducing dwell times (Wong and Key 2014). SATS are members of staff that are positioned on the platforms and are meant to help maintain consistent passenger behaviour on platforms, in theory improving reliability and safety.

5.2 Simulation Literature

Discrete event simulation has been continually developed since it first emerged in the 1950s. As simulation technology and computing power has grown, the complexity and power of such simulations has increased drastically. One publication highlights that the amount of new software appearing supporting discrete event simulation has been decreasing and the software packages available are consolidating down to a few organisations (Hollocks 2005).

A number of studies have attempted to demonstrate a simulation approach to railway modelling. One study used an Arena based model to analyse urban freight on the Newcastle-Upon-Tyne (Motraghi and Marinov 2012). The authors took a decomposition approach to modelling the railway, by breaking the railway down into sections that are then simulated and analysed as individual components. The results from this study present findings on station utilisation and the percentage of time trains spent working versus waiting, so demonstrating how efficiently the railway is utilising its resources. The model ignores the interactions of rolling stock with signalling systems and other rolling stock entities and it greatly simplifies the movement of trains between stations. This could lead to some inaccuracies in the model, particularly as more stations are linked up and errors in the simulation will accumulate..

Another study looked at simulating complex railway networks which have different speed limits of certain points (Lu, Dessouky, and Leachman 2004). This study used a velocity augmenting algorithm to capture the change of speed in trains between stations and auto-regulating the headways between trains as would be done in reality. These algorithms were shown to significantly improve the performance of the system by reducing the average flow time and average delay time. This approach has the advantage of more accurately assessing the time which trains take to travel

between stations and accurately spacing trains. However it significantly increases the complexity of modelling. The authors found computing power to be a limiting factor since the time of this study processing power has significantly increased. This study also developed a graphical framework for modelling railway systems. Graphical modelling was shown to be an effective method of forming the model abstraction before building a simulation model. The model was validated by comparing to real world performance and was deemed to be an “adequate approximation to the current system performance”.

One study implemented an event-based simulation to test new control systems within a railway already at peak capacity (Grube, Núñez, and Cipriano 2011). This dynamic simulation was implemented in an object oriented style using MATLAB. The same authors also demonstrated a model to assess the ability of more intelligent a train management systems to improve a system’s capacity (Nunez et al. 2010) when infrastructure constraints prevent more traditional capacity increases through a greater number and faster trains or improved signalling systems. Similarly, an object oriented approach to simulating railway networks was described in another paper in order to determine the validity of timetables and the design of railway control policies (Paolucci and Pesenti 1999).

Many of these simulations explicitly model a train’s position relative to stopping points and other trains, usually with the goal of assessing timetables or railway control systems. It may be unsuitable to apply Monte-Carlo simulation due to the greatly increased computer processing requirements.

One study reviewing the reliability of railway models argues that traditional performance and reliability metrics fail to adequately describe perceived service reliability for high frequency systems (Landex 2012). The author demonstrates methods of predicting network delay propagation.

The study also provides an interesting conceptual framework for whether simulation models are “operational”, “tactical” or “strategic” in their nature as can be seen in figure 2. This model is based on a previous university report (Kass 1998).

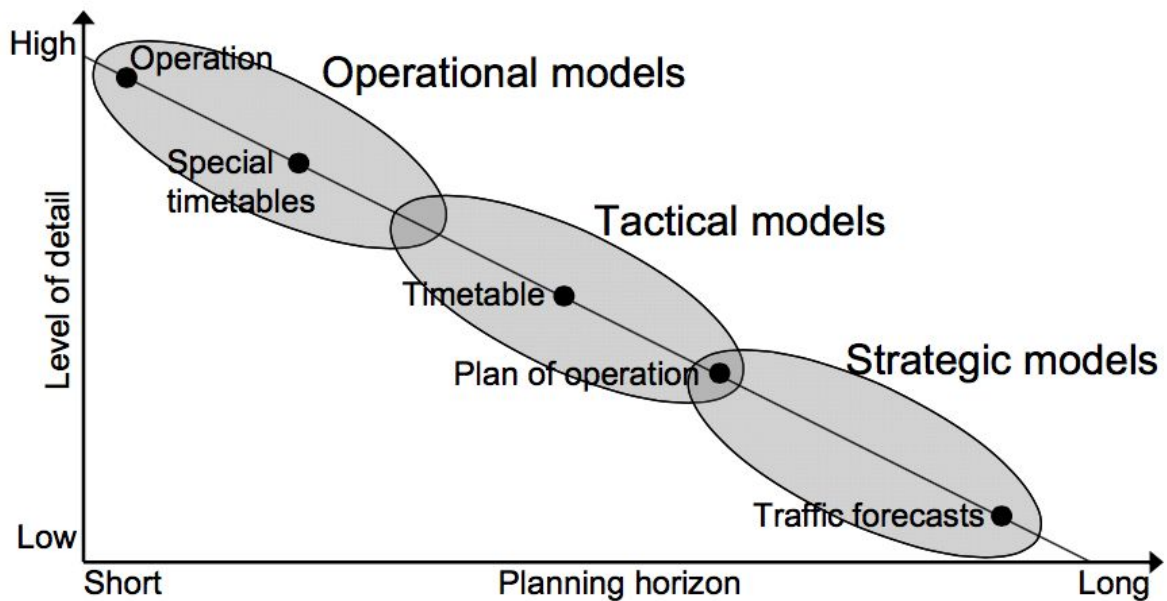


Figure 2 - Different categories of simulation (Kass 1998).

The author argues that operational and tactical models can be used to assess the impact of delays and delay propagation. These models include software packages such as Opentrack (Nash and Huerlimann 2004) and Railsys (Siefer 2008).

Simulations and models always represent an abstraction of reality. The degree of abstraction to use depends on a several factors:

1. The question being answered (modelling requirement)
2. The computing power available (technical resource)
3. The time available to build a simulation (human resource)

The capacity of a railway system must always be considered in the context of reliability. There exists a trade-off between the two parameters. Considering both extremes helps visualise this. Running no trains on a network will yield a reliability of 100%, that is, the operator is guaranteed to meet their target of trains! Conversely increasing the capacity on a network well beyond the constraints of a system could yield a capacity of 0%, that is, the trains run but the operational targets are never met as the system cannot cope with the increased capacity. This relationship is succinctly described in a recent conference presentation which argues that railways should aim to operate in the “practical capacity” region which is normally around 60% to 70% of the “theoretical capacity” of the system (Kontaxi and Ricci 2009). This relationship is described in figure 3.

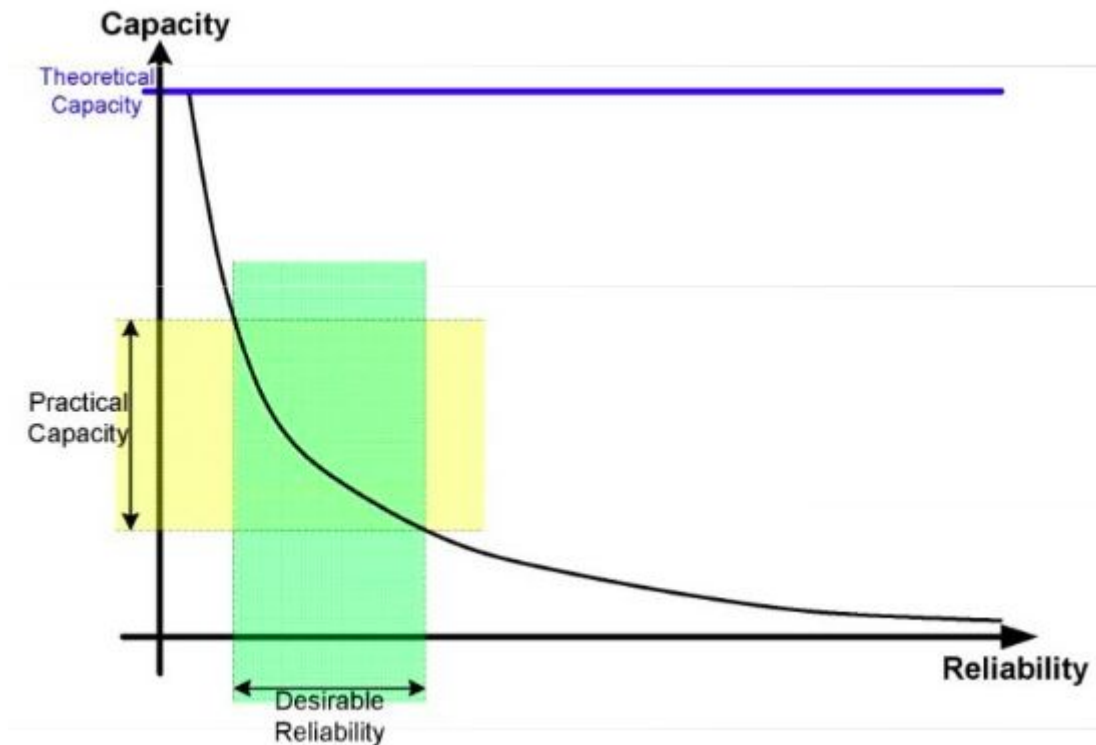


Figure 3 - Practical capacity versus desirable reliability (Kontaxi and Ricci 2009).

One interesting study investigates how to model capacity at an airport runway using a discrete event simulation model. The objective of the study was “to develop guidelines for improving runway throughput while minimizing aircraft holding times” (Mukkamala et al. 2008). The study investigates a number of discrete-event simulation rules such as first come first serve to explore whether capacity is influenced by these factors - first come first served was shown to be the most efficient strategy. Different arrival times of aircraft and sequences were also tested. This study leans towards assessing what strategies can be used to best maximise capacity on runways, however it does not test what the absolute maximum capacity of the system is or what level of reliability can be expected from the performance. The focus of the results is on throughput per hour (which is identical to a common railway capacity measure) as well as the average and maximum holding (delay) times. This study does not allude to what simulation software is used to produce the results.

6. Methodology

6.1 Dwell Time Modelling

“Performance can be judged only by a statistical study of historical times” (Edwards Deming 2002).

The methodology was first started by exploring whether dwell times on the London Underground tend to follow a specific distribution. Following this an investigation will be carried out to try and quantitatively identify factors that influence the dwell time on the London Underground. Once the data is gathered three forms of analysis will be carried out:

1. The distribution of dwell times will be plotted visually in order to establish what distribution may fit the data if any. A goodness of fit test, e.g. the Kolmogorov-Smirnov test, will then be used on the data to compare it to the estimated distribution.
2. Multivariate analysis will be used to identify significant factors with a p-value success criteria of less than 0.05.
3. Regression analysis will be used to form a predictive model. Linear and non-linear regression models will be tested. A randomly selected 80% of the dataset will be used for developing the model and the remaining 20% of the dataset will be used for validation.

Primary data will be gathered from London Underground data sources. This is an advantage as the data are readily available for this project. The disadvantage of limiting the study to London Underground data is that it will the project is fundamentally constrained to working with the data which the organisation choose to gather.

6.1.1 Data Sources

A number of different data sources were utilised in this study and are described below.

6.1.1.1 Dwell Times from NETMIS

In order to carry out a study on dwell times these dwell times first need to be acquired. Fortunately the entire London Underground network is electrified and has track circuits. Track circuits are simple electronic components which detect the presence of a train on a section of track. The track circuits feed information to the signalling systems which controls the movement of trains around the network. These track circuits also provide data to control centers and centralised databases. Thus, there exists data across the entire London Underground network for every train and every

movement that occurs. Since this data provides information on when trains arrive and when they depart platforms, the wheel stop to wheel start dwell time can be calculated simply using equation 1.

$$DwellTime = WheelStartTime - WheelStopTime$$

Equation 1 - Assumed Dwell Time Calculation

This track circuit occupation data is known in London Underground as NETMIS data.

The month of November 2015 was used as a sample for this study, since this is the period of time for which the RODS surveys were also carried out (RODS will be explained in the chapter “Passenger Data from RODS”). NETMIS data taken from the entire network for every day in November 2015 produced a tabulated comma separated file which was 187mb in size and contained 527,332 rows of data.

Since the mean dwell time is of interest, this summary statistic needs to be calculated for each location and direction combination for each station on the network. A Python script was developed to process the raw NETMIS data in order to calculate the mean dwell at each location-direction combination. However, the mean for each location-direction combination cannot simply be calculated for the entire month. This would yield a dataset that is too small and could not be correlated with passenger number data for results that are statistically significant. To overcome this challenge a statistical approach called resampling was employed.

Resampling time series based data, such as NETMIS data, involves calculating a summary statistic (in this case the mean) for a period of time at fixed intervals in the data. Taking one data label, Victoria Northbound, as an example. Raw NETMIS data with the dwell time is calculated in table 1.

Victoria Station Northbound Example Data		
Arrive	Depart	Dwell
06:00:00	06:00:30	00:00:30
06:01:10	06:01:25	00:00:15
06:01:50	06:02:30	00:00:40

06:03:00	06:03:25	00:00:25
06:03:55	06:04:45	00:00:50
06:05:00	06:05:30	00:00:30

Table 1 - Typical dwell time data

If this data is resampled at a 2 minute frequency, with the arrival time as the resample reference point and the mean calculated, the data will reduce and look like table 2.

Resampled Victoria Station Northbound Example Data	
Resample time	Dwell
06:02:00	00:00:28
06:04:00	00:00:38
06:06:00	00:00:30

Table 2 - Resampled dwell times

6.1.1.2 Delays on the Tube from CUPID

London Underground also gathers data on delays. Delays are defined as service affecting failures (SAFs) which last two minutes or more. A SAF will be anything that incurs inconvenience or delay to a customer. This includes escalator failures, lift failures, partial station closures, full station closures, platform closures, train delays and line suspensions. All SAFs of 2 minutes or more are recorded in a centralised database known as CUPID. The accessible version of this database contains SAFs from 2009 to present. Along with a description of the incident, the length of the delay, the delay duration, a basic root cause analysis and a calculation of lost customer hours is included. Each incident also contains data on location, time, train number (if applicable), line and direction (if applicable).

This large database on reliability will be included as part of the dwell time study to investigate any correlation between the number of SAFs on a line and the dwell time. The primary disadvantage of CUPID data is that it does not contain data for service affecting failures less than two minutes. Additionally there is likely human error introduced in the recording of these incidents, which is done remotely at a control centre shortly after the incident occurs. Statistics based on this data can be

calculated, such as the mean number of SAFs per year for each line and the mean duration of each SAF.

The dataset is labelled by a combination of Line and direction for allocation of the SAF rate. This breakdown can be seen in table 3.

LINE ID	LINE NAME	RODS LINE NAME	DIRECTION CODE	DIRECTION DESCRIPTION
0	Bakerloo	Bakerloo	0	Bakerloo Northbound
0	Bakerloo	Bakerloo	1	Bakerloo Southbound
2	Central	Central	0	Central Westbound
2	Central	Central	1	Central Eastbound
3	Victoria	Victoria	0	Victoria Northbound
3	Victoria	Victoria	1	Victoria Southbound
4	Metropolitan	Metropolitan	0	Metropolitan Northbound/Westbound
4	Metropolitan	Metropolitan	1	Metropolitan Southbound/Eastbound
5	Northern	Northern	0	Northern Northbound
5	Northern	Northern	1	Northern Southbound
6	Jubilee	Jubilee	0	Jubilee Northbound/Westbound
6	Jubilee	Jubilee	1	Jubilee Southbound/Eastbound
7	Piccadilly	Piccadilly	0	Piccadilly Eastbound
7	Piccadilly	Piccadilly	1	Piccadilly Westbound
8	District	District	0	District Westbound
8	District	District	1	District Eastbound
11	W&C	n/a	0	Waterloo and City WestBound
11	W&C	n/a	1	Waterloo and City EastBound
13	Circle Hammersmith & City	Hammersmith & C	0	Circle and H&C Westbound/Inner Rail
13	Circle Hammersmith & City	Hammersmith & C	1	Circle and H&C Eastbound/Outer Rail
14	SSL	n/a	0	SSL Northbound/Westbound
14	SSL	n/a	1	SSL Southbound/Eastbound

Table 3 - Labelling of line IDs and direction codes. Primary data gathered from within London Underground.

For the purposes of this study the Circle Hammersmith & City and District are to be excluded due to a mixture of rolling stock types on these lines. The W&C data will also be excluded because the line is only two stations which both act as termini - termini are excluded from this analysis.

Processing the number of SAFs on each line direction combination yields the following summary statistics in table. Line availability was calculated by assuming that unavailability is defined by any train delay that affects the service. The SAFs in table are only train delay SAFs and do not include the number of service affecting failures for other categories, such as station closures or escalator failures. This is because train delay SAFs are directly relevant to the service. The SAF data can be seen in table 4.

Line Ops	Direction	Number of Service Affecting Failures (over 2330 days)	Mean Delay Time (minutes)	Failure Rate (failures per day)	Total Downtime (days)	Line Availability
Bakerloo	NB	4,017	6.10	1.72	17.01	0.9928
Bakerloo	SB	3,246	6.16	1.39	13.88	0.9941
Central	EB	11,170	6.14	4.79	47.61	0.9800
Central	WB	9,703	5.56	4.16	37.49	0.9842
Jubilee	EB	7,452	5.28	3.20	27.31	0.9884
Jubilee	WB	9,941	4.67	4.27	32.27	0.9863
Metropolitan	NB	5,683	7.14	2.44	28.17	0.9881
Metropolitan	SB	7,539	6.81	3.24	35.66	0.9849
Northern	NB	8,133	5.02	3.49	28.34	0.9880
Northern	SB	7,849	5.22	3.37	28.45	0.9879
Piccadilly	WB	7,164	6.17	3.07	30.70	0.9870
Piccadilly	EB	6,266	6.24	2.69	27.14	0.9885
Victoria	NB	6,246	4.57	2.68	19.81	0.9916
Victoria	SB	5,026	4.94	2.16	17.23	0.9927

Table 4 - Data on service affecting failures on the London Underground. Primary data gathered by analysing internal London Underground CUPID data.

The “failure rate” is the statistic which will be used in this study. It could be argued that Line Availability should be used instead however upon inspection, a scatter plot in figure 4, it can be seen that the failure rate correlated extremely strongly with line availability, thus only the failure rate needs to be used.

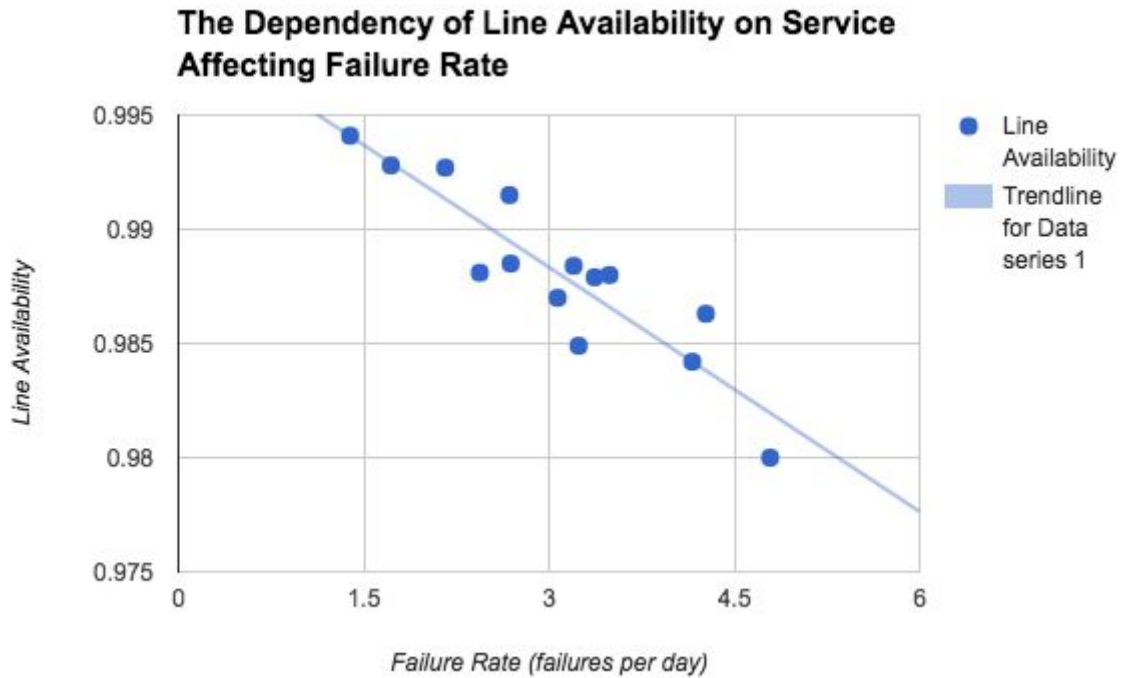


Figure 4 - The relationship between the service affecting failure rate and line availability for London Underground lines

Line availability thus intends to target whether the service is “good” or disrupted in some way. Figure 5 shows an example of good service across all lines as displayed to customers:

Bakerloo	Good service
Central	Good service
Circle	Good service
District	Good service
Hammersmith & City	Good service
Jubilee	Good service
Metropolitan	Good service
Northern	Good service
Piccadilly	Good service
Victoria	Good service
Waterloo & City	Good service
London Overground	Good service
TfL Rail	Good service
DLR	Good service

Figure 5 - Good service on all lines on the London Underground

Service affecting failures result in passengers receiving service updates which state minor delays, severe delays, line suspensions or line closures.

This study is concerned primarily with the “good service” aspect, that is, attempting to understand the factors which drive performance in the sub-2 minute region. Poor performance due to service affecting failures accounts for a relatively small percentage of overall system performance loss, given the high availability figures presented.

Having said this, service affecting failures still have a knock on effect on the normal running of the train service. A service that breaks down needs to recover. The frequency of failures could therefore have an effect, directly or indirectly, on the dwell time.

Failure rates at a system level, including failures that do not directly affect the train service, result in higher values as the failures also categories such as station closures, platform closures and line suspensions.

The system level failure rate is described for each line in table 5. This is the data which will be fed into the analysis.

Line	SAF Failure Rate (mean number of incidents per day on line)
Bakerloo	6.389
Central	14.294
Jubilee	12.624
Metropolitan	7.929
Northern	13.005
Piccadilly	9.607
Victoria	6.844

Table 5 - Train service affecting failure rates of London Underground lines

6.1.1.3 Passenger Data from RODS

In order to investigate the number of passengers on the network, and compare this data with dwell times, several options exist. It is possible to make assumptions about the number of passengers at each location due to population density in different boroughs, both at home and at work. This could provide a crude estimate of the number of people using London Underground services. The clear disadvantage of this approach is that precise estimates of the number of people boarding and alighting trains cannot be calculated.

A second approach to estimate passenger numbers is through the instrumentation of trains which measure the onboard mass. This is a preferred approach as data collection can be automated and calibrated to a good degree of accuracy. However there are not enough resources as part of this study to instrument trains to this degree and London Underground has only instrumented a minority of the network, with instrumentation not calibrated to measure passenger numbers but rather provide a rough indication of loading.

A third option involves manual observation and counting of platform and in-train CCTV footage. While the accuracy of this approach would be high, the time required to review enough footage would be too long to be practical. A good option for estimating passenger number would be to use Oyster Card entry data to stations. Oyster cards are used by most people across London, with the exception of travel card users. The electronic tap-in through barriers at stations provides data on the number of entries and exits from stations. However it is difficult to estimate precisely how these figures translate into platform boarders and alighters.

Fortunately, London Underground has commissioned for a number of years “Rolling Origin and Destination Surveys” (RODS). These surveys, which take place yearly each November across the network, involve a combination of counting of passengers boarding and alighting trains at each station along with an analysis of Oyster Card entry and exits count. The result is an estimate that can be provided network-wide for the number of boarders and alighters at each location. This data is broken down into 15 minute slots throughout the day so provides a high degree of granularity.

The disadvantages of this data is that it provides estimates and not exact values, and that the data are only captured during November. Since passenger numbers change throughout the year this means that the data are November-specific. This will not be a problem for this study however, since it

is possible to only use NETMIS data from November too. Additionally error which exists in the estimates should be balanced out by good estimated when enough data is used.

Looking at the data up close at a single location is revealing. Taking Holborn as an example, there are high fluctuations in the number of boarders and alighters throughout the weekday as can be seen in figure 6.

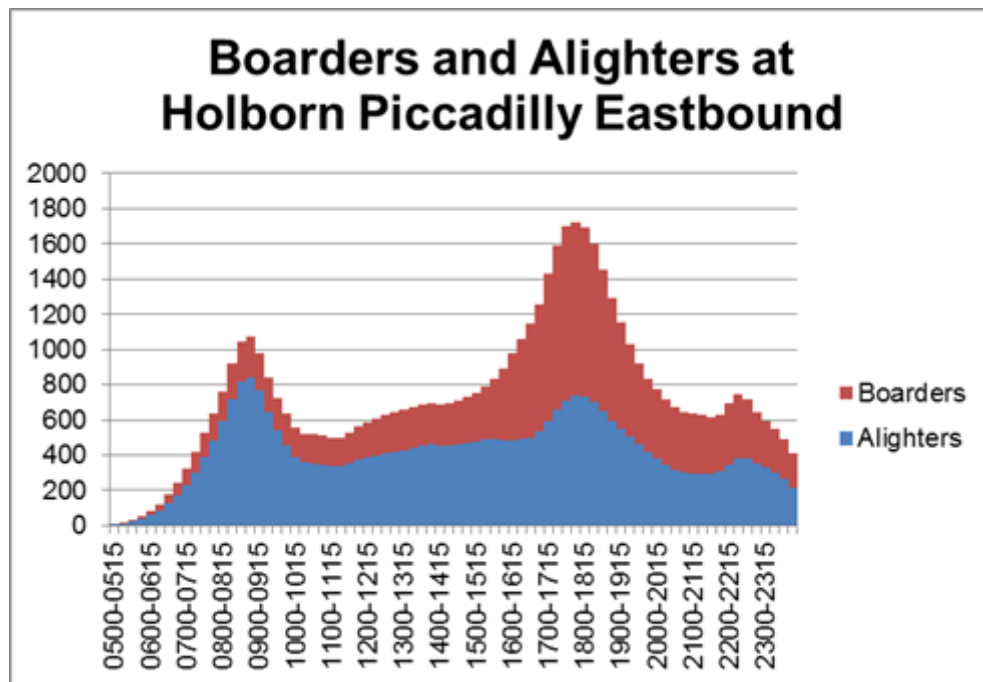


Figure 6 - Number of boarders and alighters on the London Underground at Holborn station. Primary data sourced from internal London Underground RODs data.

The mean dwell time for each 15 minute time period throughout the day is strongly dependent on the total number of boarders and alighters as can be seen in figure 7.

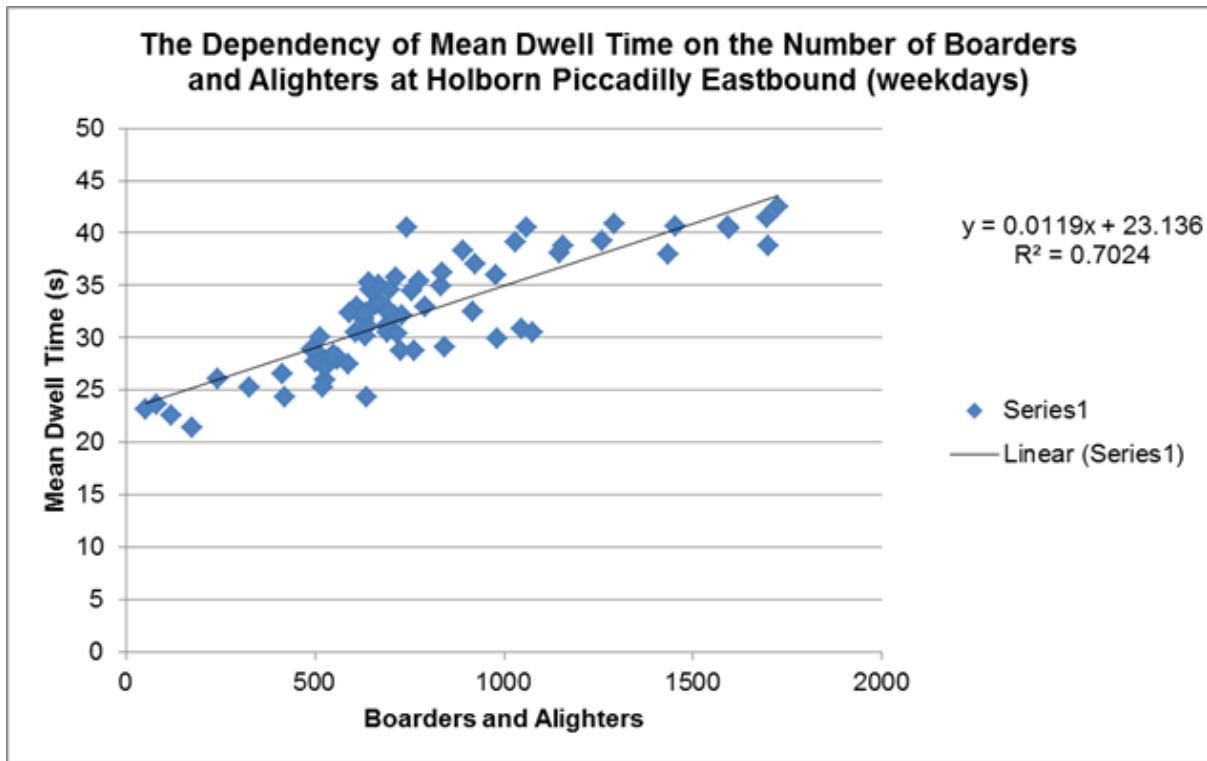


Figure 7 - The dependency of mean dwell time on the number of boarders and alighters

6.1.1.4 Rolling Stock Technical Specifications

Primary data from internal London Underground rolling stock technical specifications provides useful information on the internal layout of each train. Data on the number of seating spaces in each train, along with the floor space capacity of each train was captured. This was done for all lines except the District Line. This is because the District line at the time of this investigation runs two different rolling stock types, “S-stock” and “D-stock”, thus making it difficult to differentiate in the dwell time data which train is being represented. Therefore the District Line is excluded from this study.

A statistic for each line representing the number of trains on the line relative to the length of the line and the number of stations was calculated. This statistic will be referred to as train density and is calculated by the following equation.

$$TrainDensity = NTrainsOnLine / NStationsOnLine / LineLengthKM$$

The calculation for train density is presented in table 6.

Line	Length of Line (Track KM)	Number of Stations on Line	Line Density (Number of Stations per KM of Track)	Number of Trains in Service	Train Density (number of trains per station density)
Bakerloo	23.2	25	1.08	36	33.41
Piccadilly	71	53	0.75	86	115.21
Central	74	49	0.66	85	128.37
Northern	58	50	0.86	106	122.96
Jubilee	36.2	27	0.75	63	84.47
Victoria	21	16	0.76	47	61.69
Metropolitan	67	34	0.51	58	114.29

Table 6 - The calculation of train density statistic

The seating capacity of each train type for each line was recorded from the technical specifications. Additionally the standing capacity of each train type was recorded. This data is presented in table 7.

Line	Rolling Stock Type	Seating Capacity (seats per train - excluding tip-up seats)	Standing Capacity (m2)
Bakerloo	1972 Mkl and MkII Stock	268	116.6
Piccadilly	1973 Stock	228	114
Central	1992 Stock	272	155.02
Northern	1995 Stock	200	110.36
Jubilee	1996 Stock	234	145.92
Victoria	2009 Stock	252	153.2

Metropolitan	S8 Stock	306	174
--------------	----------	-----	-----

Table 7 - Rolling stock seating and standing capacities

6.1.2 Distribution Analysis Methodology

6.1.2.1 Parameter Estimation

Parameter estimation is the process of estimating a set of values that describe how a particular distribution is shaped given a set of observations. A Python script was developed by the author to automatically estimate the parameters of 84 different distributions against 30 different samples of dwell time data from across the London Underground Network. The methodology for goodness of fit testing and results are explained further on. These distributions which have been tested are listed in table 8 and the descriptions are taken from the SciPy website (SciPy 2017b).

#	Distribution	Description
1	alpha	An alpha continuous random variable.
2	anglit	An anglit continuous random variable.
3	arcsine	An arcsine continuous random variable.
4	beta	A beta continuous random variable.
5	betaprime	A beta prime continuous random variable.
6	bradford	A Bradford continuous random variable.
7	burr	A Burr (Type III) continuous random variable.
8	cauchy	A Burr (Type XII) continuous random variable.
9	chi	A Cauchy continuous random variable.
10	chi2	A chi continuous random variable.
11	cosine	A chi-squared continuous random variable.
12	dgamma	A cosine continuous random variable.
13	dweibull	A double gamma continuous random variable.
14	erlang	A double Weibull continuous random variable.
15	expon	An Erlang continuous random variable.
16	exponnorm	An exponential continuous random variable.
17	exponweib	An exponentially modified Normal continuous random variable.
18	exponpow	An exponentiated Weibull continuous random variable.

19	f	An exponential power continuous random variable.
20	fatiguelife	An F continuous random variable.
21	fisk	A fatigue-life (Birnbbaum-Saunders) continuous random variable.
22	foldcauchy	A Fisk continuous random variable.
23	foldnorm	A folded Cauchy continuous random variable.
24	genlogistic	A folded normal continuous random variable.
25	genpareto	A Frechet right (or Weibull minimum) continuous random variable.
26	gennorm	A Frechet left (or Weibull maximum) continuous random variable.
27	genexpon	A generalized logistic continuous random variable.
28	genextreme	A generalized normal continuous random variable.
29	gausshyper	A generalized Pareto continuous random variable.
30	gamma	A generalized exponential continuous random variable.
31	gengamma	A generalized extreme value continuous random variable.
32	genhalflogistic	A Gauss hypergeometric continuous random variable.
33	gilbrat	A gamma continuous random variable.
34	gompertz	A generalized gamma continuous random variable.
35	gumbel_r	A generalized half-logistic continuous random variable.
36	gumbel_l	A Gilbrat continuous random variable.
37	halfcauchy	A Gompertz (or truncated Gumbel) continuous random variable.
38	halflogistic	A right-skewed Gumbel continuous random variable.
39	halfnorm	A left-skewed Gumbel continuous random variable.
40	halfgennorm	A Half-Cauchy continuous random variable.
41	hypsecant	A half-logistic continuous random variable.
42	invgamma	A half-normal continuous random variable.
43	invgauss	The upper half of a generalized normal continuous random variable.
44	invweibull	A hyperbolic secant continuous random variable.
45	johnsonsb	An inverted gamma continuous random variable.
46	johnsonsu	An inverse Gaussian continuous random variable.
47	kstwobign	An inverted Weibull continuous random variable.
48	laplace	A Johnson SB continuous random variable.

49	levy	A Johnson SU continuous random variable.
50	levy_l	Kappa 4 parameter distribution.
51	logistic	Kappa 3 parameter distribution.
52	loggamma	General Kolmogorov-Smirnov one-sided test.
53	loglaplace	Kolmogorov-Smirnov two-sided test for large N.
54	lognorm	A Laplace continuous random variable.
55	lomax	A Levy continuous random variable.
56	maxwell	A left-skewed Levy continuous random variable.
57	mielke	A Levy-stable continuous random variable.
58	nakagami	A logistic (or Sech-squared) continuous random variable.
59	ncx2	A log gamma continuous random variable.
60	ncf	A log-Laplace continuous random variable.
61	nct	A lognormal continuous random variable.
62	norm	A Lomax (Pareto of the second kind) continuous random variable.
63	pareto	A Maxwell continuous random variable.
64	pearson3	A Mielke's Beta-Kappa continuous random variable.
65	powerlaw	A Nakagami continuous random variable.
66	powerlognorm	A non-central chi-squared continuous random variable.
67	powernorm	A non-central F distribution continuous random variable.
68	rdist	A non-central Student's T continuous random variable.
69	reciprocal	A normal continuous random variable.
70	rayleigh	A Pareto continuous random variable.
71	rice	A pearson type III continuous random variable.
72	recipinvgauss	A power-function continuous random variable.
73	semicircular	A power log-normal continuous random variable.
74	t	A power normal continuous random variable.
75	triang	An R-distributed continuous random variable.
76	truncexpon	A reciprocal continuous random variable.
77	truncnorm	A Rayleigh continuous random variable.
78	tukeylambda	A Rice continuous random variable.

79	uniform	A reciprocal inverse Gaussian continuous random variable.
80	vonmises	A semicircular continuous random variable.
81	vonmises	A skew-normal random variable.
82	wald	A Student's T continuous random variable.
83	frechet_r	A trapezoidal continuous random variable.
84	frechet_l	A triangular continuous random variable.

Table 8 - Distribution descriptions (SciPy 2017b)

There are several methods to estimate parameters for distributions based on a given set of sample data. For the purposes of this study the maximum likelihood estimation (MLE) method will be used. From the SciPy website:

“This fit is computed by maximizing a log-likelihood function, with penalty applied for samples outside of range of the distribution. The returned answer is not guaranteed to be the globally optimal MLE, it may only be locally optimal, or the optimization may fail altogether.” (SciPy 2014)

This MLE method has the advantage of being quite suitable for large datasets. As the sample size grows larger the parameter estimates converge to the correct value. The parameters for each distribution were first estimated using maximum likelihood estimation.

The estimated distribution was plotted against the raw data on a probability plot and a linear coefficient of determination (R-squared) value was calculated. The coefficient of determination provides an indication of how well the estimated distribution matches the sample data. Rank regression is an alternative method of estimating a distribution parameters from the observed data. It is also possible to manually plot data to a specific distribution by hand, however this is impractical for the purposes of this study when computational methods are available.

6.1.2.2 Scenarios to Test

30 different scenarios were established to gather data from. Since dwell time behaviour may be different at different locations and times of the day, these scenarios were selected in order to try and achieve a balanced set of samples. The criteria for selection was as follows:

- Eight lines were selected
- Two stations from each line were selected: one in “zone 1” and one in “zone 4” (with the exception of the Victoria Line for which the furthest from Central London station is in “zone 3”.

- For each station data was acquired in a single direction from the NETMIS database.
- Peak and off-peak data was acquired from the NETMIS database with peak data being defined as 07:00 to 09:00 and off-peak data being defined as 12:00 to 14:00

6.1.2.3 Analysing the Data

Once the MLE estimate for the distribution parameters is calculated for the observed data, the quality of the fit of the estimated distribution against the data needs to be scored. A probability plot of the observed data against the quantiles of the estimated distribution is then generated and a linear coefficient of determination (R^2) score is calculated. If the estimated distribution perfectly matches the observed data this score will be 1. Similarly a score closer to 0 represents a poor fit. This method thus provides a way of judging the quality of the fit for the estimated distribution against the observed data. SciPy was used to generate the probability plots and carry out the calculations (SciPy 2017a).

Since the calculations of these distributions is assumed to be statistically significant, as each calculation is based on hundreds of data samples, the mean R^2 scores will be directly compared with each other as a way of judging the most generalisable distribution.

6.1.3 Factor Analysis Methodology

6.1.3.1 Pre-Processing Data

In order to create a complete dataset a significant amount of data assimilation and reduction needs to be carried out. The goals that need to be achieved with pre-processing are:

1. Automation of the merging of data sources
2. Filtering the data to exclude data points that are not relevant to the study
3. A dataset large enough to have enough data points for statistically significant results
4. A dataset resampled at a low enough frequency to capture enough samples for calculation of a mean dwell with minimal noise and maximal signal

6.1.3.1.1 Matching NETMIS with RODS

A Python script was developed to merge the NETMIS dataset with the RODS dataset. A sample of this merged dataset can be found in appendix C. This process presented a number of challenges. While the NETMIS data is presented in a tabular fashion, the RODS data is not, and is instead presented in customised spreadsheets. The RODS data presents passenger numbers in the form of boarders and alighters for each location across the network, in 15 minute time bands throughout the day. For each row of NETMIS data, the appropriate RODS boarders and alighters statistic needed to be looked up.

This was also challenging to achieve as the RODS data did not use standard timestamp codes and the station names were in a different format to the NETMIS data. Additionally a number of data points could not be matched, due to either the time of the day being out of service hours (RODS only covers 5am to midnight), or a non-service location such as a depot being in the data leading to a lack of data on numbers of boarders and alighters. These non-matches were removed from the dataset. Overall 84% of the original NETMIS data was retained after matching with RODS. The final script took 40 minutes to complete on internal Transport for London computers.

6.1.3.1.2 Filtering Data Points

The next step with processing the data is to remove data points that are not relevant to the study. This is important as dwell times that do not reflect a normal service pattern will produce results that are not relevant. With the database in a raw format, there are a number of scenarios that will skew the distribution of data, these include:

- Trains stopping at terminus stations which have special long dwell times
- Engineering trains stopping for unusually long times
- Trains that need to de-train passengers from the service, for example if the train needs to enter the depot. These will have unusually long train times.
- Trains that are reversing and require a driver to change ends
- Complex junctions where a significant amount of operational signalling delay is introduced

These exclusions are normal parts of a train service. However this study focuses on through running stations in order to limit variability in the data.

A simple solution to filtering out unusual dwell times is to filter out dwell times over a certain duration. Since it is known from previous studies that the number of boarders and alighters correlates with the dwell times, it can be hypothesised that the optimum cut-off point for filtering the dwell times can be found by calculating the correlation between the number of boarders and alighters and the dwell time for a range of cut-off points. The results of this exercise are below in figure 8. Note that this is with raw data that has not been resampled.

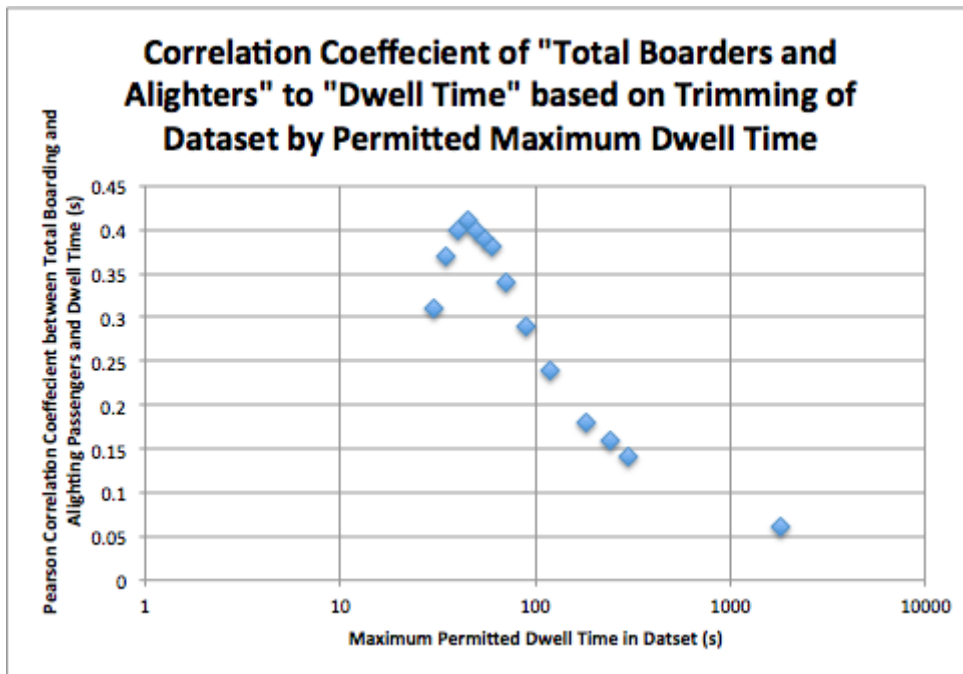


Figure 8 - Correlation coefficient of total boarders and alighters to dwell time

It was found that the Pearson correlation between the number of boarders and alighters and the dwell time peaks when the data is filtered for a cut-off point of 45 seconds. This suggests that boarding and alighting data does not generalise well to unusual dwell times, but does correlate well with normal-service dwell times of up to 45 seconds.

Another way of filtering out data is to explicitly select locations that would not yield data relevant to the goals of this study. Thus, stations which meet the following criteria will be excluded from this study:

- Termini (e.g. Uxbridge)
- Flat junctions (e.g. Baker Street)
- Reversing moves (e.g. Queen's Park and Rayners Lane)
- Stations with route in/out of depots (e.g. Northfields)

These station types would be subjected to special operating dwell times which could not easily be differentiated from regular service dwell times.

One final way of filtering the data is to visually inspect it. A script was developed to plot the distribution of dwell times at each location. Locations which demonstrated significant non-parametric activity were removed from the study as it could be that the non-parametric dwell

time behaviour is due to special or irregular operating procedures. Thus, reducing the usability of data for the creation of a generalisable model.

6.1.3.1.3 Resampling Data

The optimum resample period was not clear initially. Increasing the period size has the benefit of increasing the number of samples available for calculating a summary statistic, but the disadvantage comes from reducing the number of samples available in the dataset. Thus a balance exists between filtering out the noise without damaging the signal.

One method of discovering the optimum resample period for this study, i.e. maximising the signal while keeping the number of data points as high as possible, is to experiment with different resample periods and measure the correlation between boarders and alighters and dwell time. Since it is known that a relationship exists between these two variables, a stronger correlation should indicate a better dataset with an improved signal to noise ratio.

The resample period was tested from 2 minutes up to 240 minutes with the tests and results presented in table 9. The spearman correlation coefficient between the dwell time and the number of boarders and alighters was calculated for each resample period. The Spearman correlation coefficient measures the strength of monotonically increasing or decreasing relationships between two variables.

The Spearman correlation coefficient was used here because:

- There is no reason to believe that the dwell time statistics linearly change with influencing factors.
- Subsequent modelling of the dwell time can take into account non-linear factors which the Pearson coefficient may discount.

Resample Period (minutes)	Spearman Correlation Coefficient between Mean Dwell Time and Total Boarders and Alighters	Number of Rows in Resampled Dataset
2	0.45	479,606
5	0.4728	310,041

10	0.5113	177,242
15	0.5325	122,579
30	0.5674	63,818
45	0.5853	43,691
60	0.594	33,584
120	0.6	18,615
240	0.6117	10,526

Table 9 - Correlation between mean dwell time and total boarders of alighters for different dwell time resample periods.

The data in table is presented in figure 9.

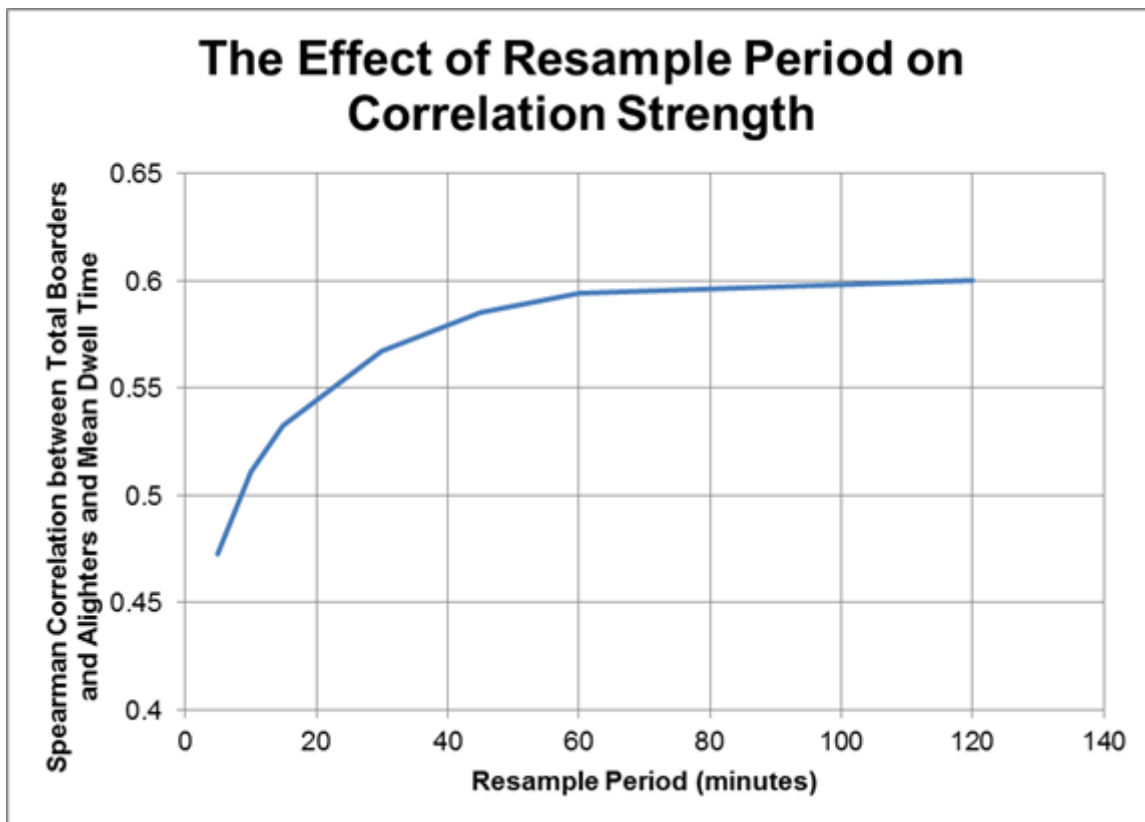


Figure 9 - The effect of resample period on correlation strength

The correlation rapidly increases initially as the resample period increases and then the rate of increase tapers off around 60 minutes.

Thus a 60 minute resample period will be used for this study. A sample of this dataset can be found in appendix D.

6.1.4 Regression Analysis Methodology

Regression analysis is a process for estimating the relationship between different variables. Many forms of regression analysis exist and several methods have been tested in this study.

A Python script was developed to carry out regression analysis using a number of different models. Models were used from the SK-Learn open-source Python library was used (scikit-learn 2017). The choice of regression algorithm to use will depend on a number of factors including complexity and linearity of the relationships. Ultimately the decision on which model to use will depend on the results of the validation which will be discussed at the end of this chapter.

A known problem with R-squared goodness of fit is that as the number of parameters increases the R-squared value increases due to over-fitting of the model. This is known as the curse of dimensionality (Wikipedia 2017).

6.1.4.1 Multiple Linear Regression Discussion

The simplest form of regression analysis, linear regression, results in the model:

$$y = b_1x_1 + b_2x_2 + b_3x_3 + \dots + c$$

Where y is the dependent variable in this instance mean dwell time, b_i is the constant associated with the factor x_i .

6.1.4.2 Decision Tree Regression Discussion

Decision trees have several advantages. The method does not require any data preparation such as converting the data into a standard normal distribution. Since decision trees use boolean logic to determine the structure of the tree, the method of modelling can be described as “white box”, i.e. it is possible to explain what is happening. By contrast methods such as neural network regression are “black box” as it is not easy to determine how the model has generated the fit.

There are also some disadvantages to using decision tree regression. The models created by decision tree learning algorithms can create overly complicated decision trees. This may lead to a model that does not generalise very well.

For example, it is possible to use a decision tree algorithm to fit the number of boarders and alighters to dwell time in the dataset, see figure 10.

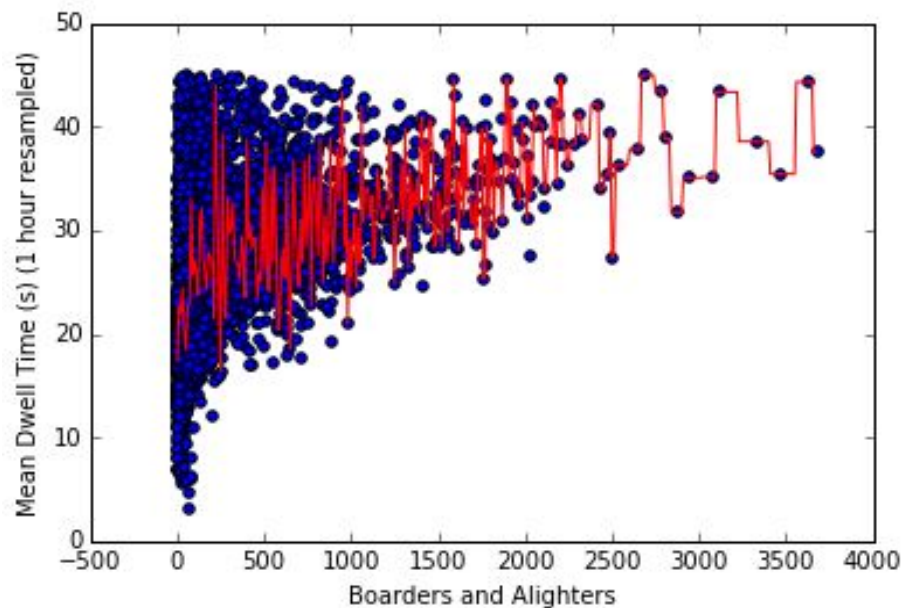


Figure 10 - Decision tree model overfitting

Figure 10 presents an R-squared value of 0.774. However it is clear that the algorithm is overfitting the model to the data, thus making it poor for generalization. This is a recognised problem with decision trees (SciKit-Learn 2016c).

A decision tree algorithm tries to find the best location to split. It does this by looking at a set number of features and deciding where to split from these. The maximum depth size of the tree was varied in order to find the maximum r-squared value. The remaining parameters for the model were based on the default values from the scikit-learn library. These parameters can be found on the scikit-learn decision tree web page (SciKit-Learn 2016c).

6.1.4.3 K-Nearest Neighbours Discussion

The K-nearest neighbour algorithm is a commonly used method for regression analysis. This method works by determining a regression based on the values of the k nearest data points.

There are two sub-methods which will be investigated: (1) uniform weights and (2) inverse distance weights. Along with these sub-methods, the key determinate of the performance of this model is based on the choice of the number for k . Thus, this will also be varied.

Figure 11 shows an example of the difference in performance between the uniform weights method (top) and inverse weights method (bottom).

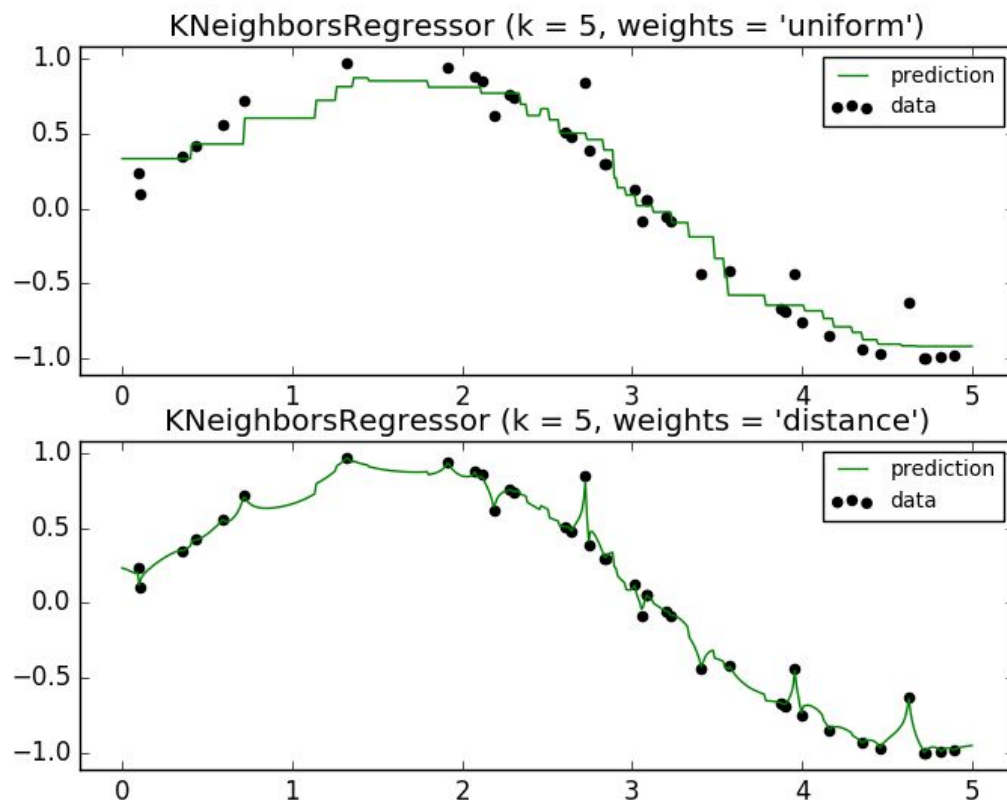


Figure 11 - Uniform versus inverse weights (SciKit-Learn 2016a)

6.1.4.4 Neural Network Discussion

The multi-layer perceptron (MLP) neural networks is a powerful tool for regression analysis. The MLP is essentially a series of regression models which are organised on top of one another. The neural network contains a number of hidden layers and hidden units within each layer. The purpose of these units is to model non-linear behaviour. Each of the units in the neural network has an activation function associated with it. There are a number of functions that can be used for the activation function such as the logistic sigmoid function and the rectified linear unit function. The sklearn algorithm for the MLP neural network attempts to optimise the squared loss in order to find the best model.

A MLP neural network machine learning regression model was applied to the data. A stochastic gradient descent optimizer ('Adam') was utilised. This is a modification to the standard stochastic gradient descent optimizer and has been shown to work well on large datasets with improved model fitting speed and validation accuracy (Kingma and Ba 2014). The rectified linear unit function was chosen. The number of units per layer was varied from 2 to 50.

6.1.4.5 Validation

The R^2 value provides an indication of the model fit. However this simply reveals how well the model fits the sample dataset. As discussed, and particularly with more complex non-parametric methods, it is quite possible to create models that produce high R^2 scores and suggesting that they are good models when they are, actually, 'overfitting'. An overfitted model is unlikely to generalise well to other data. Thus, in order to determine whether the model is actually generalisable, it needs to be validated against a second dataset.

One method of validating the sufficiency of a chosen model is to split the data into training and test sets. The training set is used for creating the regression model. The test set is used for testing the performance of the new regression model. The best performing model on the test set is then selected. For the purposes of this study 80% of the data will be partitioned for the training set and 20% will be partitioned for the test set based on guidance by Murphy ((Murphy 2012). The order of the dataset will be randomised before being partitioned. In order to evaluate the performance of the model on the validation set, the pearson correlation coefficient will be calculated between the true mean dwell times in the test set and the predicted mean dwell times from the model. This method can also be visualised.

6.2 Capacity Simulation Case Study

The Victoria Line makes a good candidate option for modelling capacity in this study because it is a simple straight line with few complexities. Recently the London Underground has made upgrades to this line in order to increase the capacity to from 24 to 36 trains per hour by making upgrades to Northern end of the branch (EveningStandard 2015). This was achieved by improving a section of track at the Northern end of the line, thereby increasing the run-in time to Walthamstow. This improvement in run-in time has unlocked the bottleneck preventing an increase in capacity and the whole line has been able to be run at 36 trains per hour as a result.

A stochastic discrete-event simulation will be constructed of the Victoria Line to assess and forecast capacity. Building this model will involve the following steps based on guidelines found in Modelling Transportation Systems book (insert reference)

6.2.1 Model Qualification and Assumptions

What is to be included in the model and what can be safely excluded.

1. Model verification: what does the logic of the model look like?
2. Model validation: test the model base case against real world behaviour?

This model will be built around each station on the Victoria Line Northbound service. Each station will be modelled in a chain and will contain a full speed run-in time, a dwell time and a run-out time. The inter-station journey time will be excluded from this model since journey time is not of interest, only capacity. Run in and run out times were modelled using Railway Engineering Simulator.

Improvements in capacity can be found by utilising modern signalling systems that allow trains to run closer together in through running sections. However this only holds true if there are not bigger constraints that exist at stations. For single platforms in a series-like railway design, the constraint on the system will be found around the run in time, the dwell time and the run out time from platforms. Multiple platforms could in theory be one technical solution to removing constraints at stations.

However, as stated before, it is not feasible to produce a discrete event simulation model that captures the mechanics behind modern signalling systems. Instead a macro-level model is used. Signalling system logic can be captured at a station level, when the interaction between trains on the running line needs to be captured a continuous simulation needs to be employed such as Railway Engineering Simulator.

Simulating run-in, dwell and run out times are a good method of predicting capacity. Note that in this context capacity is independent from journey time. Modelling the tunnel sections in-between stations is a necessary exclusion from the model because a simple discrete-event simulation (DES) model is unable to continuously update speeds of trains relative to each other. A continuous simulator with very small fixed timesteps if required for this dynamic behaviour, such as Railway Engineering Simulator (RES). Thus, including tunnel sections between stations in this DES would likely introduce a significant degree of error in the capacity calculation. The bottlenecks which determine the maximum available capacity are found at the stations.

The dwell time mean is calculated per location from the multiple linear regression model. A gamma distribution was assumed for the dwelltime with mean of 30s and standard deviation of 7.8s since this was shown to be most generalisable distribution across the range of stations and times tested earlier. This can be seen in figure 12.

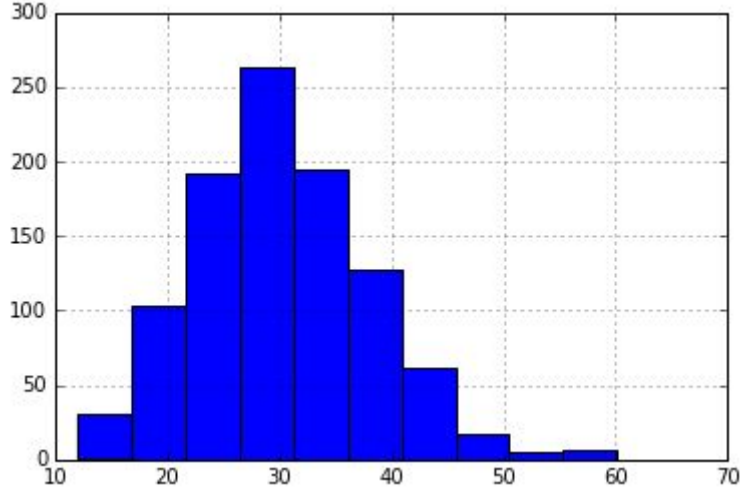


Figure 12 - Gamma distribution representing dwell time

The dwell standard deviation σ_{dwell} is assumed to be a constant 7.8s. Thus the variance is calculated by:

$$Var[Dwell] = \sigma_{dwell}^2 = 7.8^2 = 60.84$$

The Gamma distribution takes two parameters: shape k and scale θ .

Given the mean dwell μ_{dwell} and variance σ_{dwell}^2 these parameters can be calculated as follows:

$$k = \mu_{dwell}^2 / \sigma_{dwell}^2$$

$$\theta = \sigma_{dwell}^2 / \mu_{dwell}$$

Thus, each location on the Victoria Line samples dwell times from a unique gamma distribution.

Each station included in the model contains a RORIF (run-in and run-out time at full speed) and an associated number of passenger numbers. This data is presented below in table 10.

Station	RORIF (s)	Boarders and Alighters (morning peak)
Stockwell	48.4	3070
Vauxhall	50.8	1669
Pimlico	52.6	570
Victoria	44.6	3305
Green Park	46	1947
Oxford Circus	44.5	3677
Warren Street	55.1	952
Euston	46.5	1164
Kings Cross	50.9	1272
Highbury and Islington	50	888
Finsbury Park	48.8	821
Seven Sisters	51.9	459
Tottenham Hale	53	202
Blackhorse Road	52.3	100

Table 10 - RORIF and peak boarders and alighters for Victoria Line

The regression model for the mean dwell time is processed separately and loaded into the simulation. This is easily done using the pickle library in Python. Simply, the regression model is first developed and then “picked” (saved) as its own file. The file is then read into the Victoria Line simulation script and the regression model can be unpacked and used for predictive purposes - in this case calculating a mean dwell time to go into the Gamma distribution.

The dwell time sampling process is represented by figure 13.

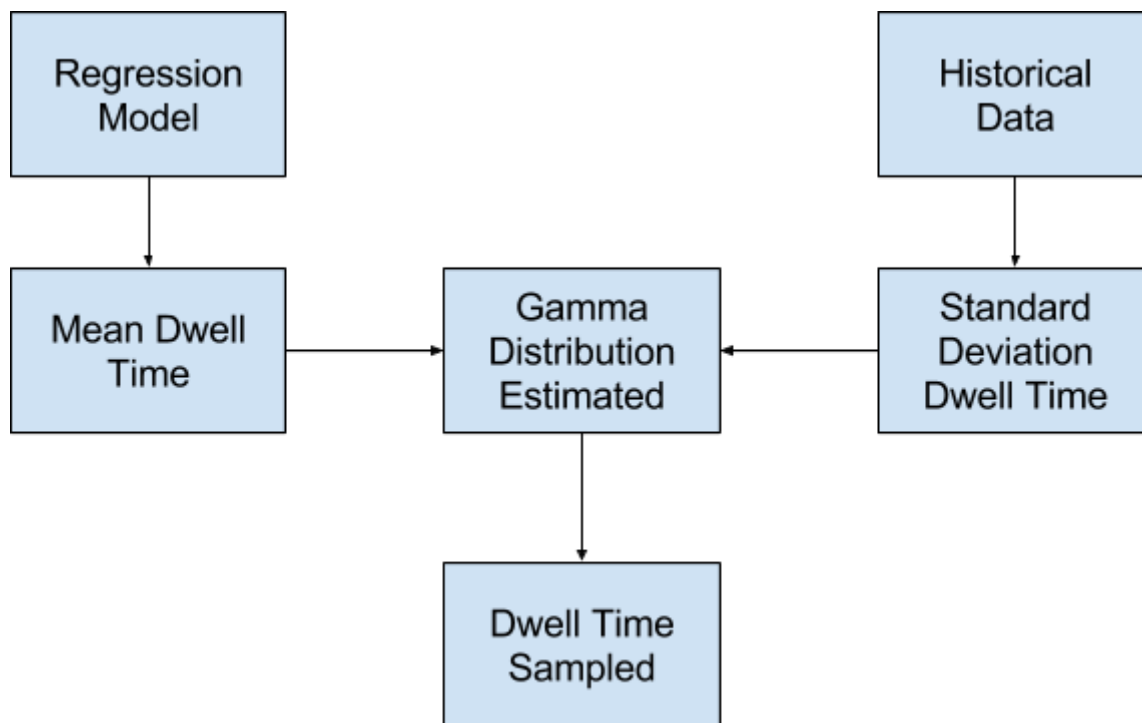


Figure 13 - Dwell time sampling process

The simulation time has been set to 1,000,000 seconds. This is the equivalent of 277.78 hours of continuous service. At 34 trains per hour this over 9000 trains in the simulation. This ensures that the summary statistics in the results are statistically significant.

The service affecting failure (SAF) rate is assumed to be constant in the simulation. In reality this is not likely to be the case, however it is difficult to forecast the change in SAF rate over the long term. Operationally London Underground should be able to improve quality and reduce the number of service affecting failures within their control, however increases in passenger numbers will likely lead to more customer related delays and ageing rolling stock could result in increased fleet failures, particularly towards the end of life as the hazard rate increases. Thus, the failure rate is assumed constant at 6.844.

In order to test the maximum capacity of the system, trains will be injected at a rate higher than what it is possible to achieve. While in practice this will not be the case, it is useful in order to test the maximum capacity of the system. Thus the trains will be generated at a rate of one every 60 seconds. Thus ensuring that a queue of trains are available to enter the system so that any spare capacity is instantly consumed.

6.2.2 System Diagram

The simplified design presented in figure 14 shows the simulation logic that is used to represent the run in and run out from the station.

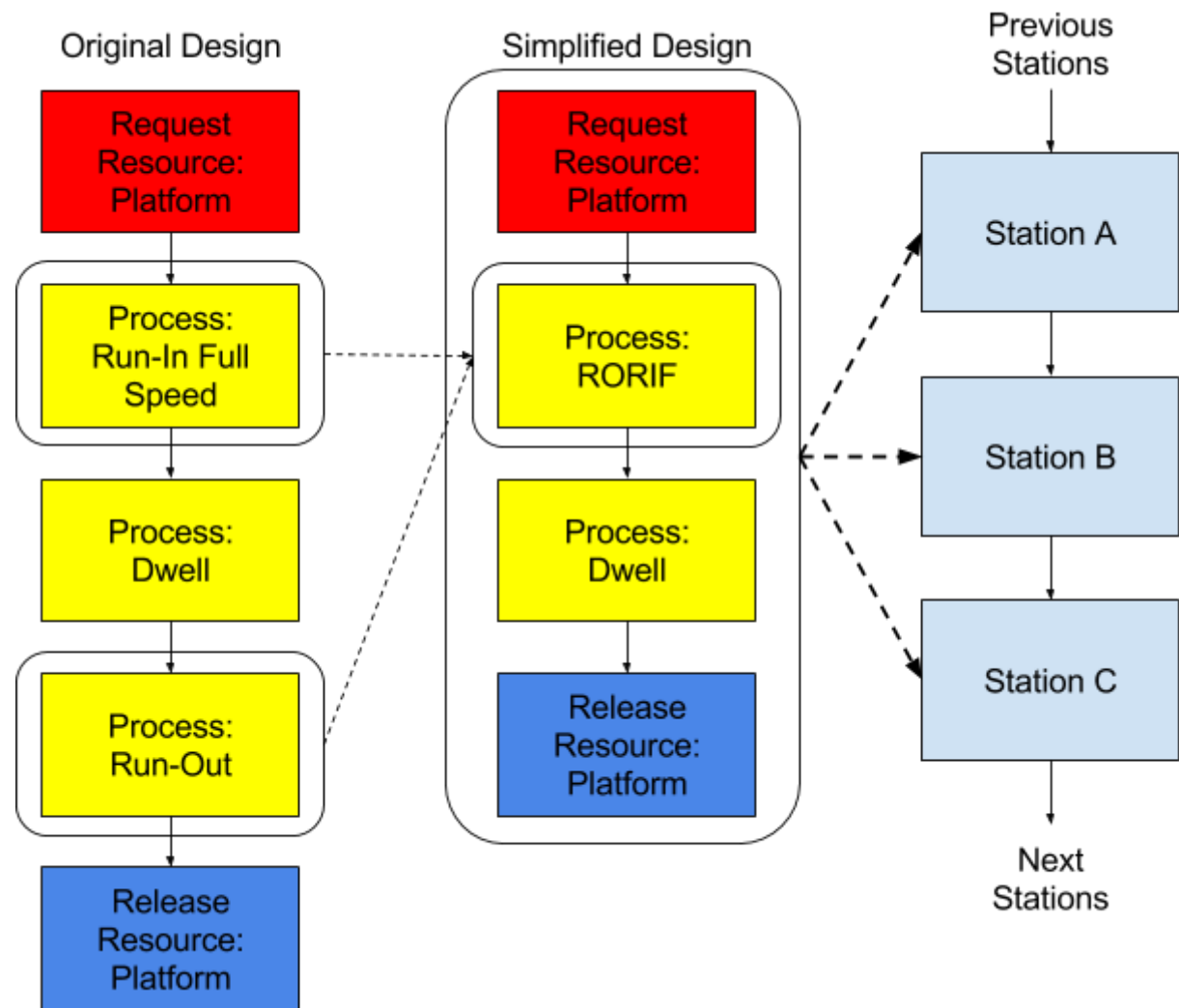


Figure 14 - Simulation logic

The platform resource has a capacity of one and controls the movement of trains in and out of the station. Resources are seized on a first come first serve basis. If the resource is not available then the request queues as shown in figure 15.

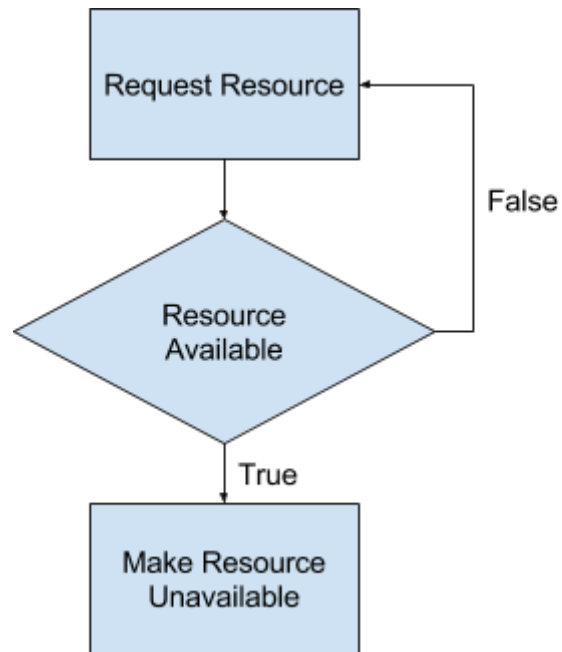


Figure 15 - Queuing logic for resource requests

Since the Victoria Line Northbound direction is a simple through-running line with no junctions used during normal service, a journey for the entire line (excluding termini) is represented in the simulation by the system in figure.

The whole system logic is shown in figure 16. Trains enter the system at the source, generated at 60 second intervals. Each station is represented by a set of sub-processes, as described previously in figure. This ensures that the flow of trains through the system is naturally limited.

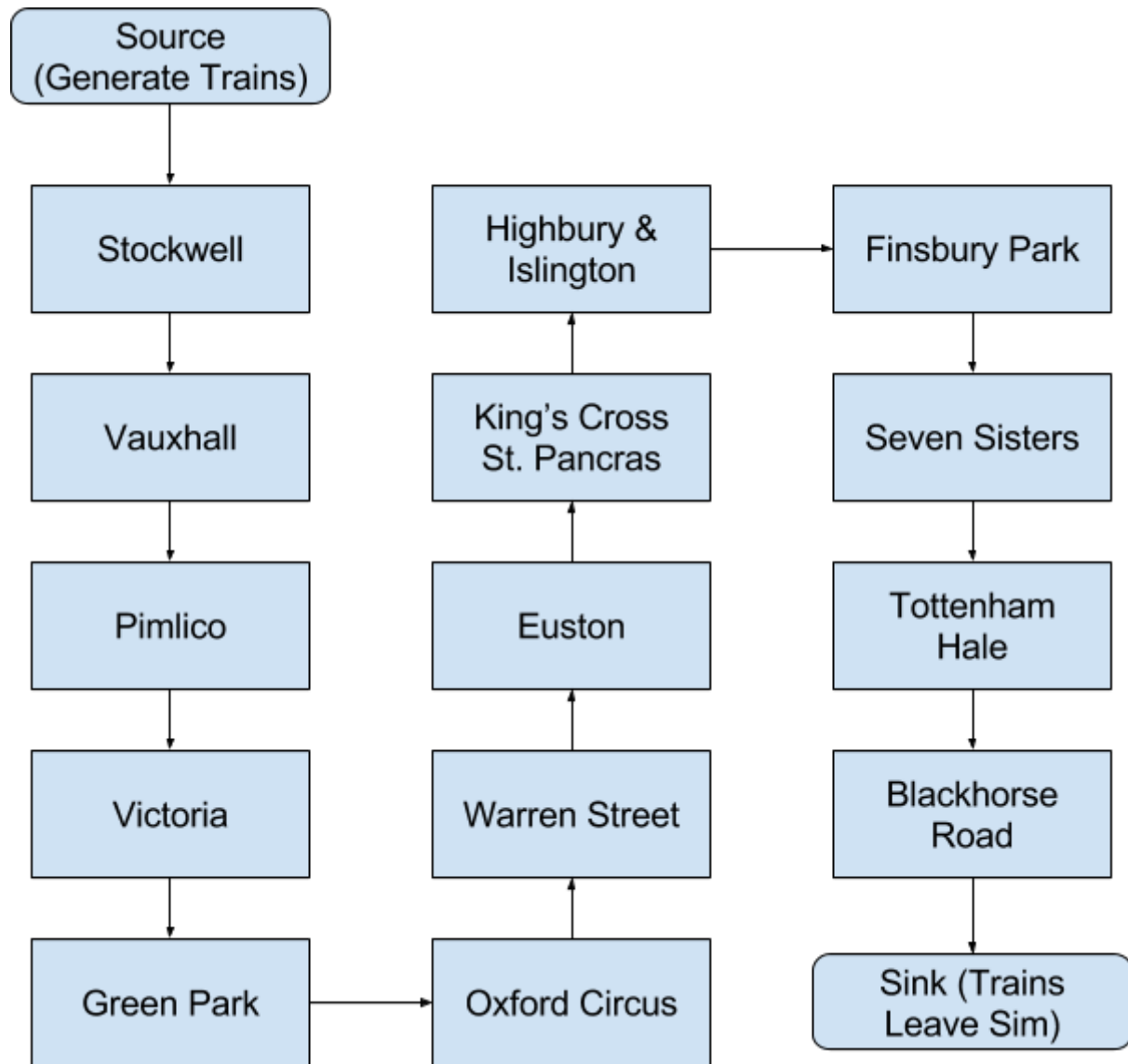


Figure 16 - Victoria Line system logic

The times at which trains leave the simulation are recorded and headways are calculated in the post-processing of the results.

6.2.3 Operational Recovery Time

Operational recovery can be described as the ability of the system to absorb delays - in other words the amount of *resilience* which the system has. In the London Underground, operational recovery is the amount of operational slack that exists in the process of running trains in and out from platforms. It is used to add resilience to a network and reduce the severity of the impact of service affecting failures. A 10 second recovery time at a station means that a train can take an extra 10 seconds to complete its run in, dwell and run out, without unduly affecting the timetable. Recovery can be thought of as scheduled extra dwell time, which is sometimes not used if the service needs to

catch up in the case of a delay, or else can be used to absorb variations in dwell time or run in and run out time.

It is difficult to be precise when stating what operational recovery time needs to be built into the railway system. Increasing recovery time leads to a reduced frequency of service, but the resilience and recoverability of the service improves. Quantifying this is not simple and no studies have been found which have successfully quantified the benefits and drawbacks. Instead, the amount of operational recovery built into the system is based on heuristic beliefs about the time. Generally a 10 second operational recovery time is built into the process of building London Underground timetables. The history of the 10 second heuristic and why it was chosen is not recorded in any literature.

It is possible to state mathematically based on simulation results what the operational recovery of the system is.

$$\text{OperationalRecovery} = \text{MeanMeasuredHeadway} - \text{MaxTargetHeadway}$$

Similarly given a fixed operational recovery it is possible to predict the maximum capacity of the system.

$$\text{MaxTargetHeadway} = \text{MeanMeasuredHeadway} - \text{OperationalRecovery}$$

Thus simulation presents the opportunity to frame recovery and the relationship to capacity in two lights: (1) the operational recovery can be varied to observe what TPH can be achieved, or (2) the target TPH can be varied to observe what operational recovery is needed to achieve the target TPH.

For the purposes of this simulation, operational recovery will be fixed at 8s and the mean headway achieved will be observed. 8s was chosen as an operational recovery heuristic because with 2015 population the results produced a mean capacity of greater than 36 trains per hour - which is what the service produces today.

6.2.4 London Population Growth

As a case study this discrete event simulation will seek to model the capacity of the Victoria Line and investigate the effects of population growth on that capacity.

Population forecasts in London up to 2050 are provided by the Greater London Authority (GLA 2015) and shown in table 11 and figure 17.

Year	Forecast London Population	Growth from 2015
2015	8,685,178.00	0%
2016	8,785,545.11	1.16%
2017	8,883,827.48	2.29%
2018	8,980,071.39	3.40%
2019	9,074,091.09	4.48%
2020	9,165,879.65	5.53%
2021	9,255,551.54	6.57%
2022	9,342,838.21	7.57%
2023	9,427,837.11	8.55%
2024	9,510,337.02	9.50%
2025	9,590,325.45	10.42%
2026	9,668,044.17	11.32%
2027	9,743,750.07	12.19%
2028	9,817,808.40	13.04%
2029	9,890,372.45	13.88%
2030	9,961,598.58	14.70%
2031	10,031,612.67	15.50%
2032	10,100,736.67	16.30%
2033	10,169,071.20	17.09%
2034	10,236,366.67	17.86%
2035	10,302,455.20	18.62%
2036	10,367,118.97	19.37%
2037	10,430,224.64	20.09%

2038	10,492,030.31	20.80%
2039	10,551,994.85	21.49%
2040	10,608,255.77	22.14%
2041	10,662,772.14	22.77%
2042	10,715,521.15	23.38%
2043	10,766,476.54	23.96%
2044	10,815,538.71	24.53%
2045	10,862,670.18	25.07%
2046	10,907,835.80	25.59%
2047	10,951,032.80	26.09%
2048	10,992,281.17	26.56%
2049	11,031,615.15	27.02%
2050	11,069,099.43	27.45%

Table 11 - London population growth

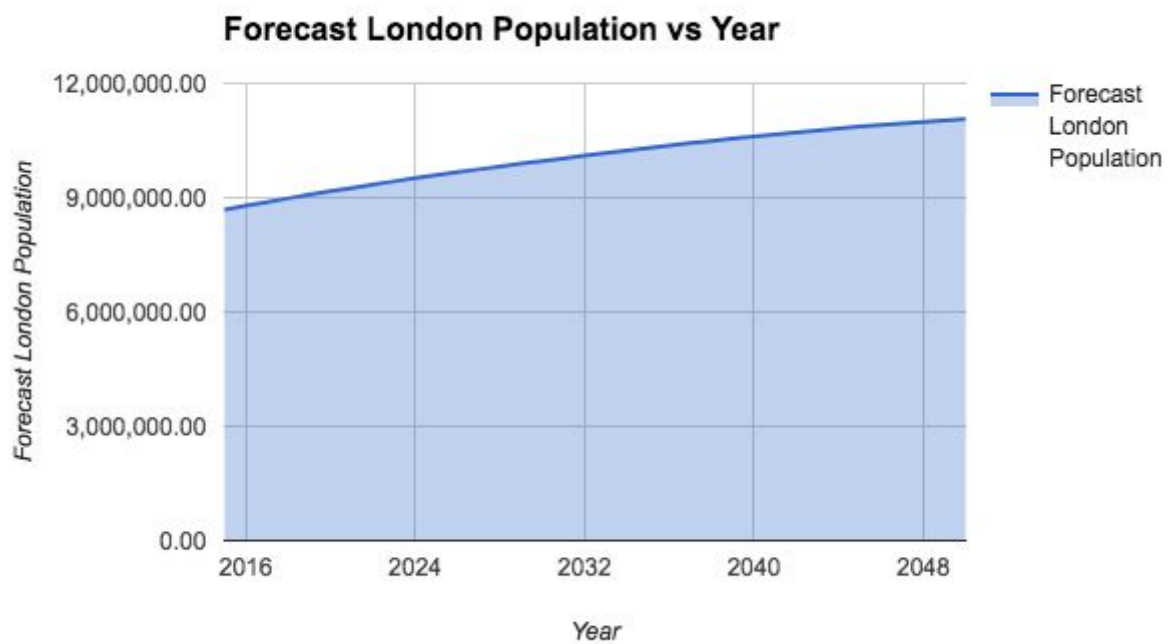


Figure 17 - London population growth

From 2015 to 2050 the population of London is forecast to increase from 8.685 million to 11.069 million which is a growth of 27%. The UN predicts that the percentage of the world's population living in cities is predicted to rise from 54.5% in 2016 to 60% in 2050 (UN 2016). London is forecast to become a “megacity”, defined as a city having a population of greater than 10 million inhabitants, by 2030.

This simulation will assume that the number of boarders and alighters at each station on the Victoria Line will increase linearly with the population growth in London. Thus, the simulation will be able to test and provide a capacity forecast for each year up to 2050. A population multiplier is included in the simulation, which for a linear relationship is set to 1. This population multiplier variable can be used to limit the effect of the increase in population on boarders and alighters. This could be useful if finer predictions could be made about which stations on the Victoria line will experience the uplift in boarders and alighters. Since these data do not exist the relationship is assumed to be linear.

The study will also assume a fixed recovery time of 8 seconds. The mean output headway will be calculated from the simulation results and a trains per hour figure will be based on this.

$$AchievableTrainsPerHour = MeanOutputHeadway/3600$$

7 Results

7.1 Dwell Time Modelling Results

7.1.1 Distribution Analysis Results

See appendix B for the data collected on distributions.

The mean R^2 score across the 30 scenarios for each distribution was calculated. The results can be seen in figure 18

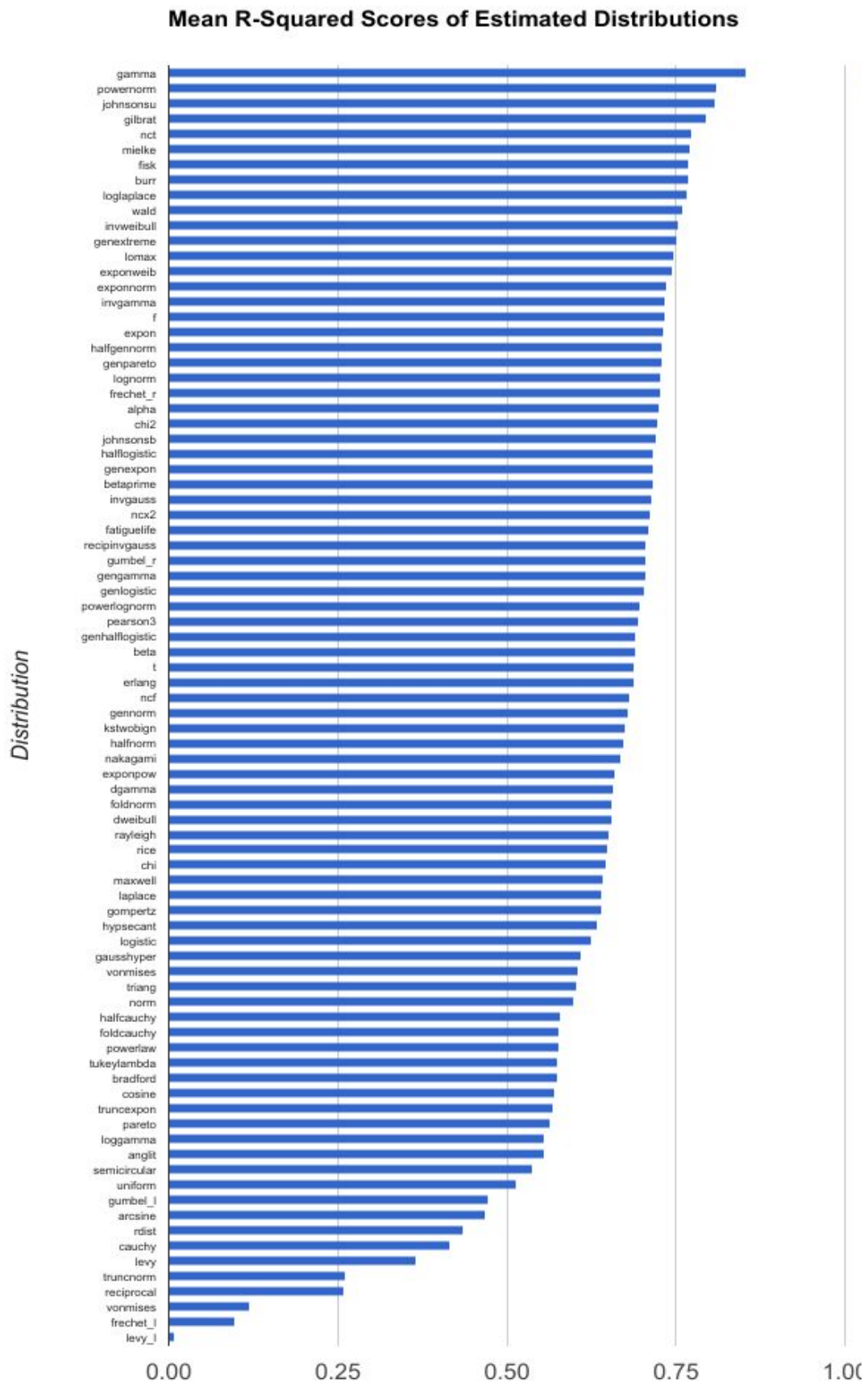
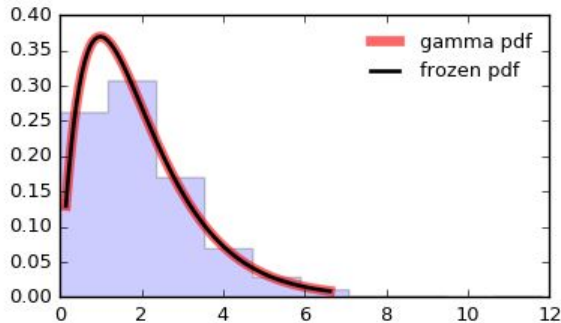
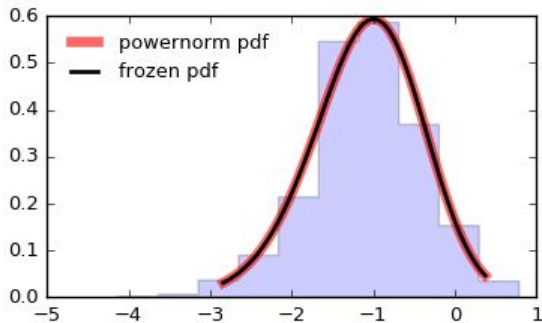


Figure 18 - Distributions tested for goodness of fit for the dwell time

The Gamma distribution was found to be, on average, the best estimate for the dwell time with a mean R^2 value of 0.85. This means that by using the Gamma distribution, on average it can be expected that 85% of the variation in the data can be captured by the distribution.

A complete list of each distribution tested along with the mean R^2 score can be found in appendix B.

The top five distributions with their mean R^2 scores are listed below in table 12.

Distribution with Reference to SciPy Website	Mean R^2 Score	Image of Distribution Taken from SciPy website. References found in first column.
Gamma	0.85	
Power normal	0.81	

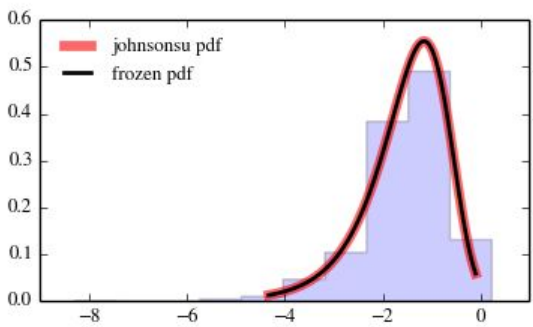
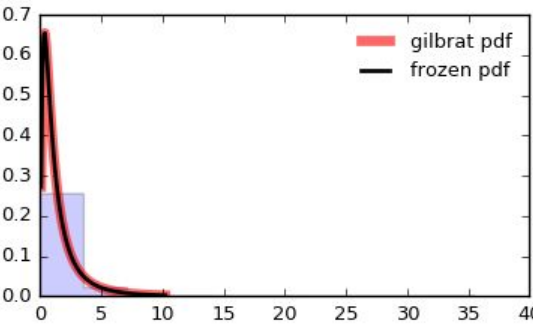
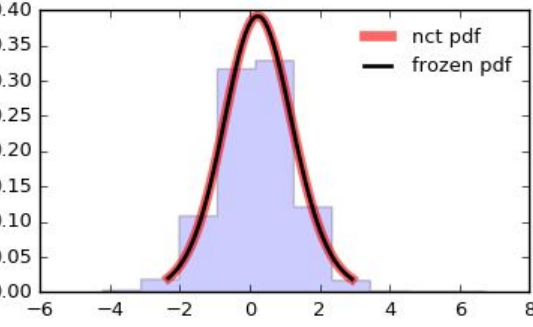
Johnson SU	0.81	 <p>A plot showing the Johnson SU distribution. The x-axis ranges from -8 to 0, and the y-axis ranges from 0.0 to 0.6. A blue histogram represents the data, and a red line represents the 'johnsonsu pdf'. A black line represents the 'frozen pdf'. The distribution is unimodal and slightly right-skewed, peaking around x = -1.</p>
Gilbrat	0.79	 <p>A plot showing the Gilbrat distribution. The x-axis ranges from 0 to 40, and the y-axis ranges from 0.0 to 0.7. A blue histogram represents the data, and a red line represents the 'gilbrat pdf'. A black line represents the 'frozen pdf'. The distribution is highly right-skewed, with a peak near x = 0.</p>
NCT (non-central students T distributon)	0.77	 <p>A plot showing the NCT (non-central students T distributon) distribution. The x-axis ranges from -6 to 8, and the y-axis ranges from 0.00 to 0.40. A blue histogram represents the data, and a red line represents the 'nct pdf'. A black line represents the 'frozen pdf'. The distribution is unimodal and symmetric, centered around x = 0.</p>

Table 12 - Top five distributions to use for the dwell time

7.1.2 Factor Analysis Results

7.1.2.1 Visualising the Data for Factor Analysis

Before carrying out any statistical analysis of the data it helps to plot the data visually. Dwell times of greater than 45 seconds were excluded from the analysis based on the peak correlation discovered at a cut-off point of 45 seconds, and presented in the methodology chapter on filtering data points. The dataset was resampled at a 60 minute frequency and the means for dwelltime and passenger number data were calculated for each resampled period. This resulted in a dataset containing N number of rows.

The following results show the distribution of each factor under test conditions in an attempt to describe the data. Scatter plots are included to show the relationship between each factor and the dwell time.

Figure 18 shows the distribution of mean dwell times in the dataset. The mean of the mean dwell times is 25.28s with a standard deviation of 5.19s. The distribution of the mean dwell times appears to be normally distributed.

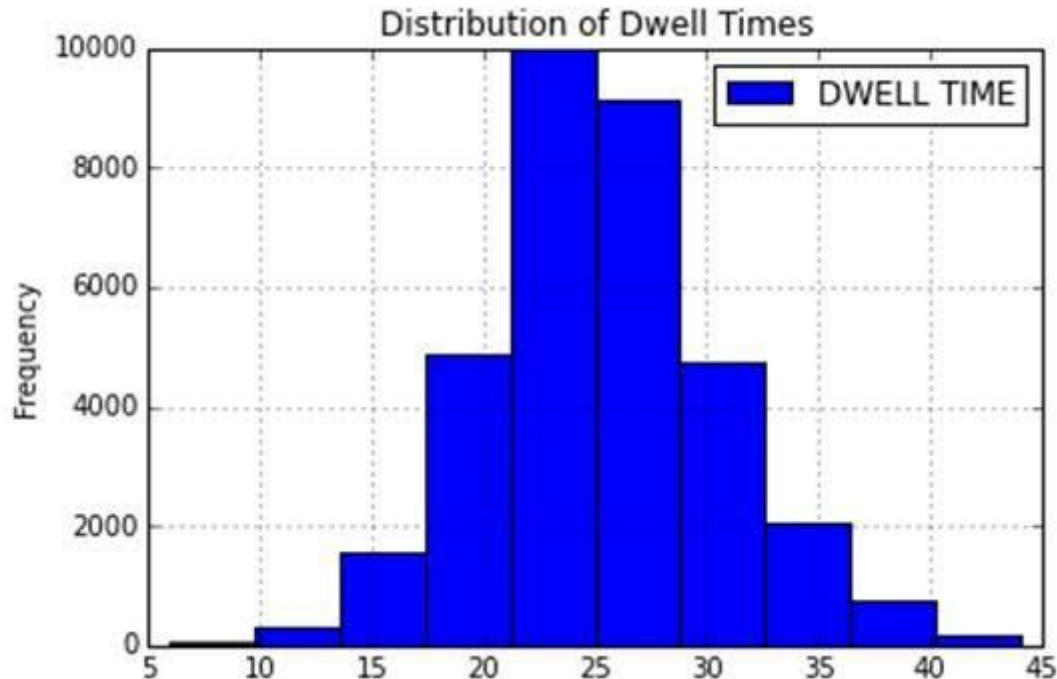


Figure 18 - The distribution of resampled mean dwell times in the dataset

The distribution of the number of boarders and alighters across the network appears is shown in figure 19. The mean number of boarders and alighters in the dataset is 336.83 with a standard deviation of 477.21. The data appears to be exponentially distributed.

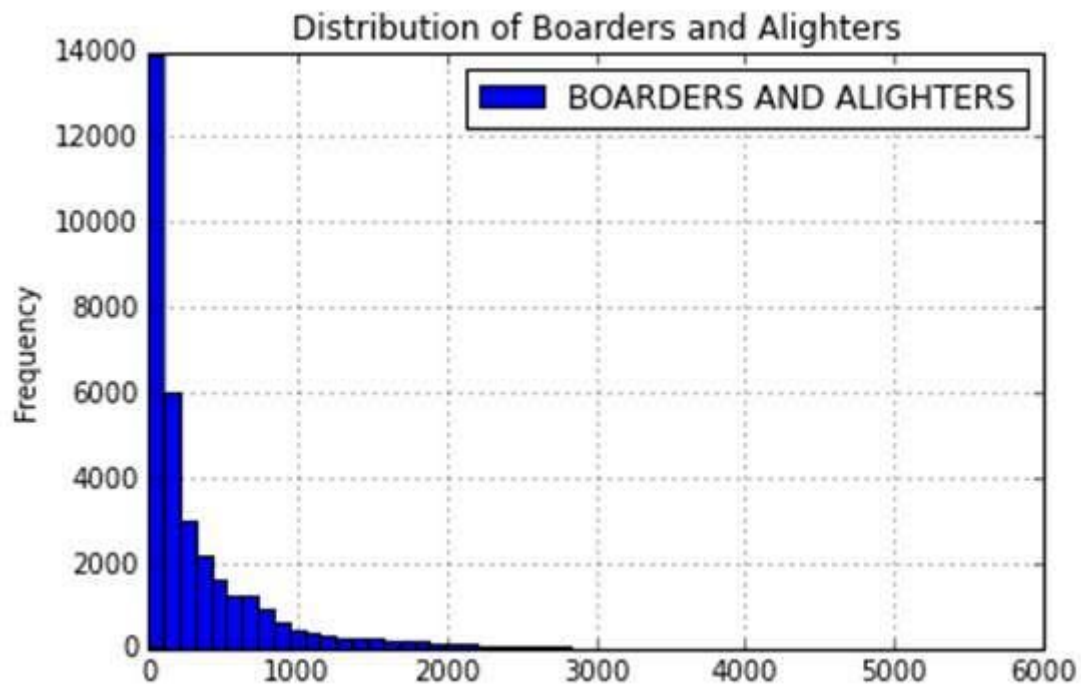


Figure 19 - Distribution of resampled mean boarders and alighters

Figure 20 shows the dependency of the resampled mean dwell times on the number of boarders and alighters. It is clear that there is a positive correlation here with increasing numbers of boarders and alighters correlating with increased dwell times. The Spearman correlation was calculated as 0.6 with a p-value of 0.

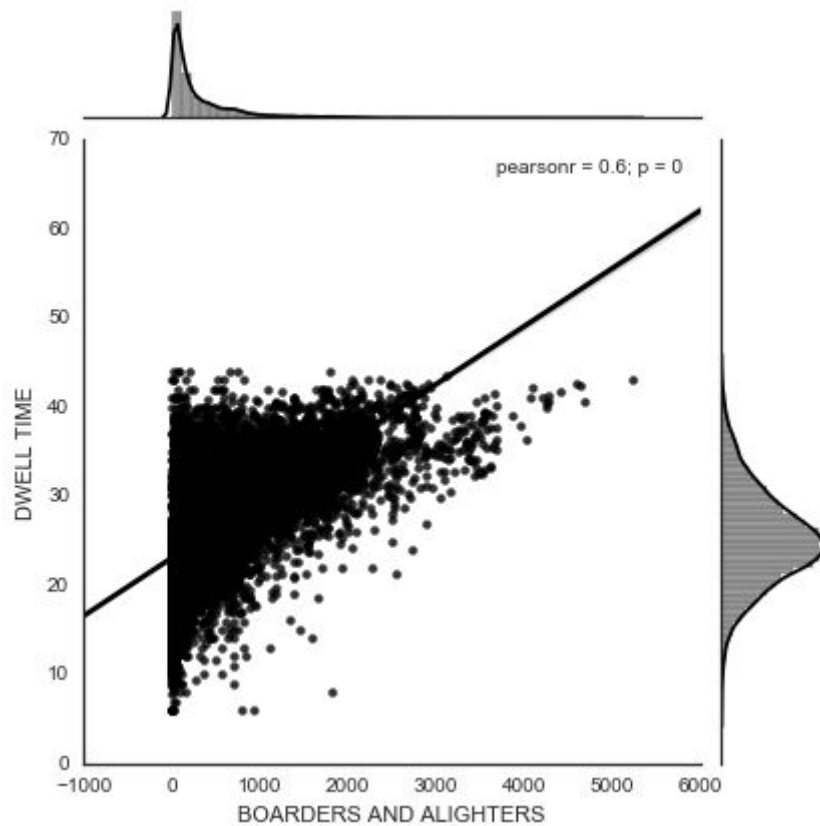


Figure 20 - The dependency of dwell time on total boarders and alighters

Figure 21 shows the distribution of the SAF rates in the dataset. Since SAF rates are assigned at a line level, there are six values for the data. The mean SAF rate is 10.83 failures per day for a line with a standard deviation of 2.79. It could be argued that the SAF rate follows a uniform distribution between the minimum value of 6.39 and 14.29.

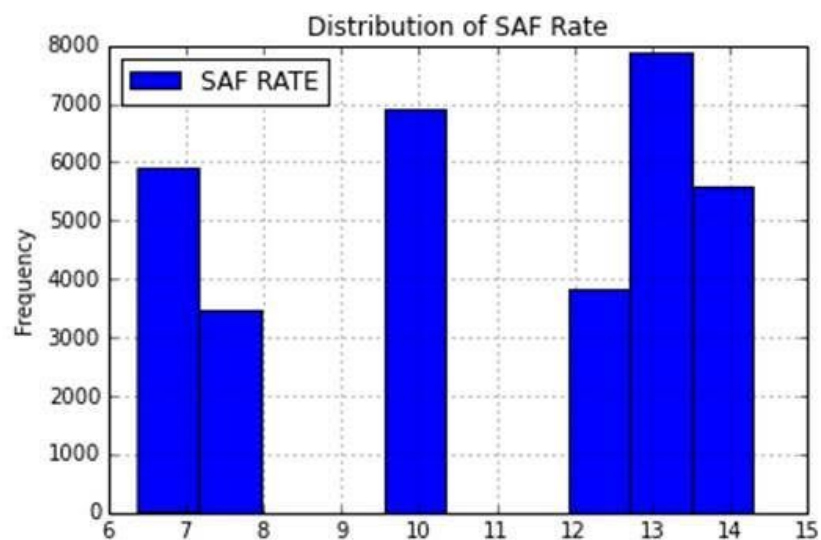


Figure 21 - The distribution of service affecting failure (SAF) rates

Figure 22 shows the dependency of mean dwell time on SAF rate. There is a positive correlation between the SAF rate and the dwell time with a pearson correlation coefficient of 0.27 and a p-value of 0.

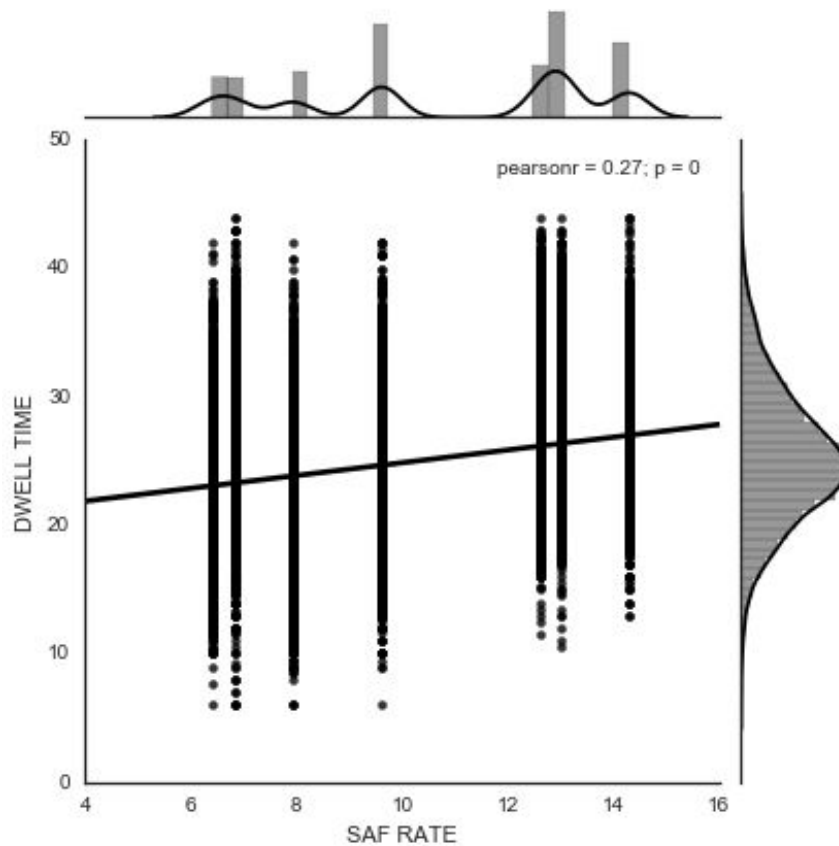


Figure 22 - The dependency of mean dwell time on SAF rate

The distribution of standing capacity across the different lines is shown in figure 23. The mean standing capacity is 133.4m² and the standard deviation is 33.21. The standing capacity of different lines does not appear to follow a particular distribution. The distribution is heavily skewed towards the minimum.

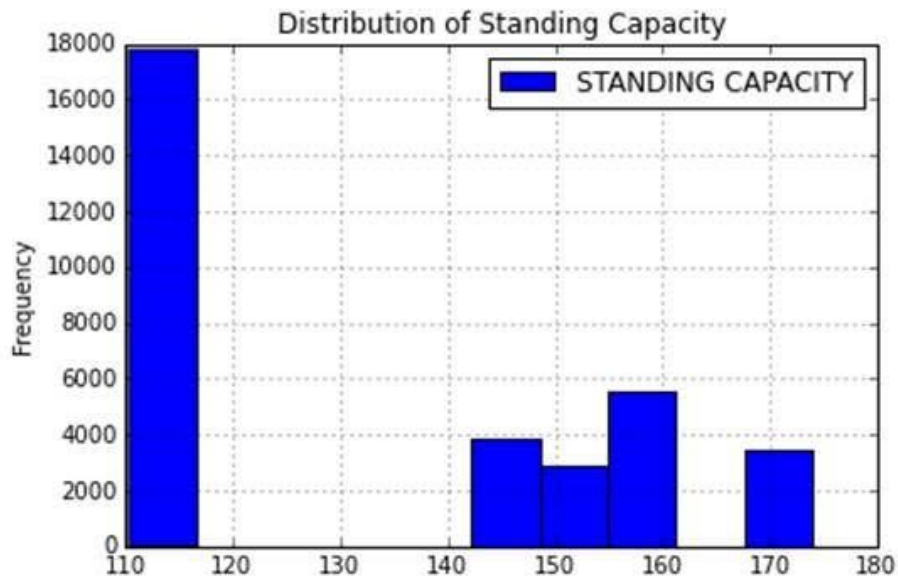


Figure 23 - The distribution of standing capacities of different rolling stock in the dataset

The dependency of dwell time on standing capacity is illustrated in figure 24. There is a small but statistically significant negative correlation here, with increased standing capacity correlating with a slight reduction in dwell time. The Pearson correlation coefficient was calculated at -0.027 with a p-value of 5.6e-7.

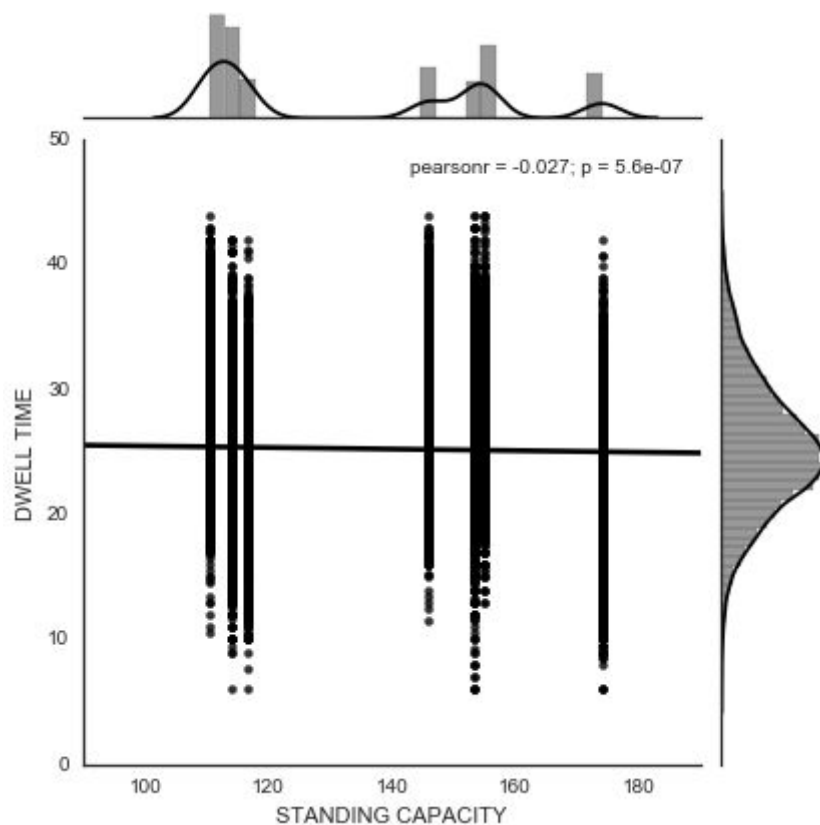


Figure 24 - The dependency of mean dwell time on standing capacity

The distribution of seating capacity across the seven lines can be seen in figure 25. The mean seating capacity is 243.13 with a standard deviation of 33.21.

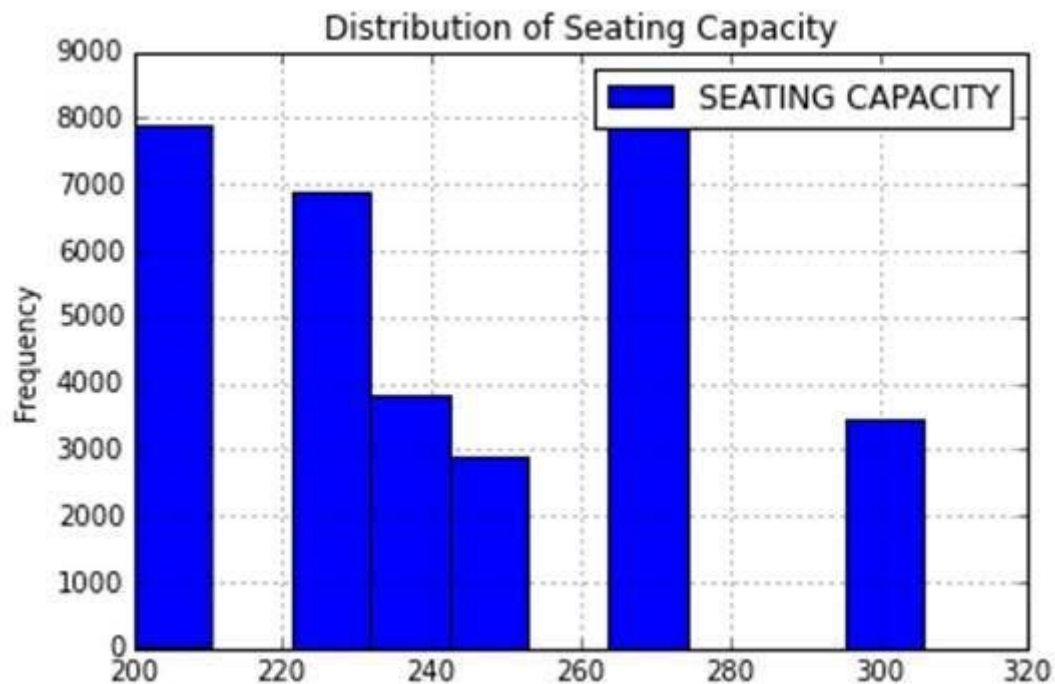


Figure 25 - The distribution of seating capacities of rolling stock in the dataset

The dependency of dwell time on seating capacity is illustrated in figure 26. There is a small but statistically significant negative correlation here, with increased seating capacity correlating with a slight reduction in dwell time. The Pearson correlation coefficient was calculated at -0.19 with a p-value of 8.1e-273.

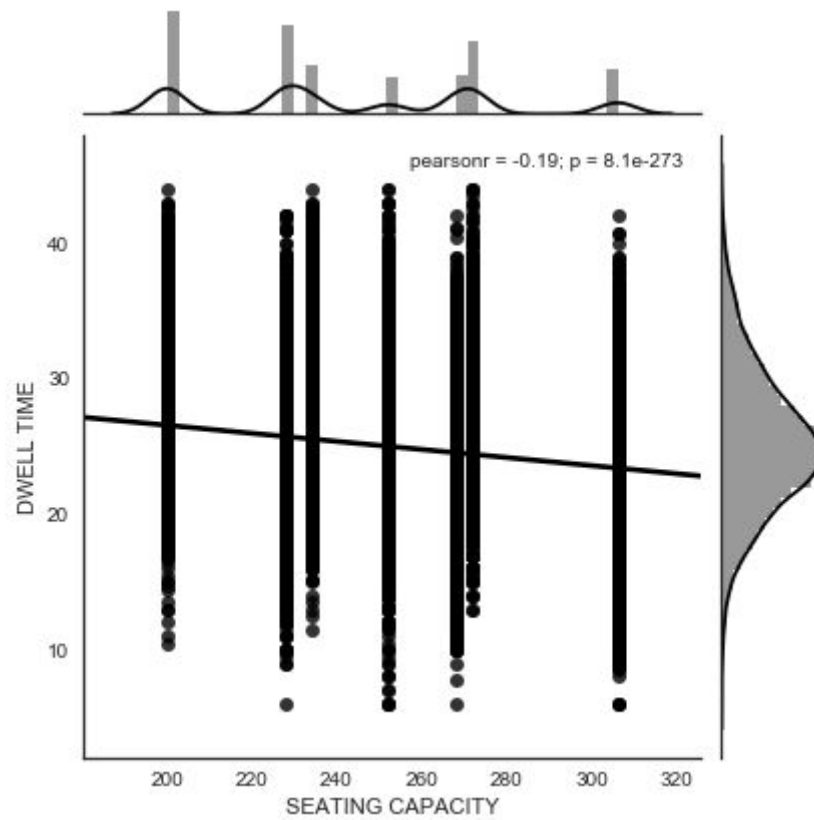


Figure 26 - The dependency of dwell time on seating capacity

The distribution of train density statistics in the dataset is presented in figure 27.

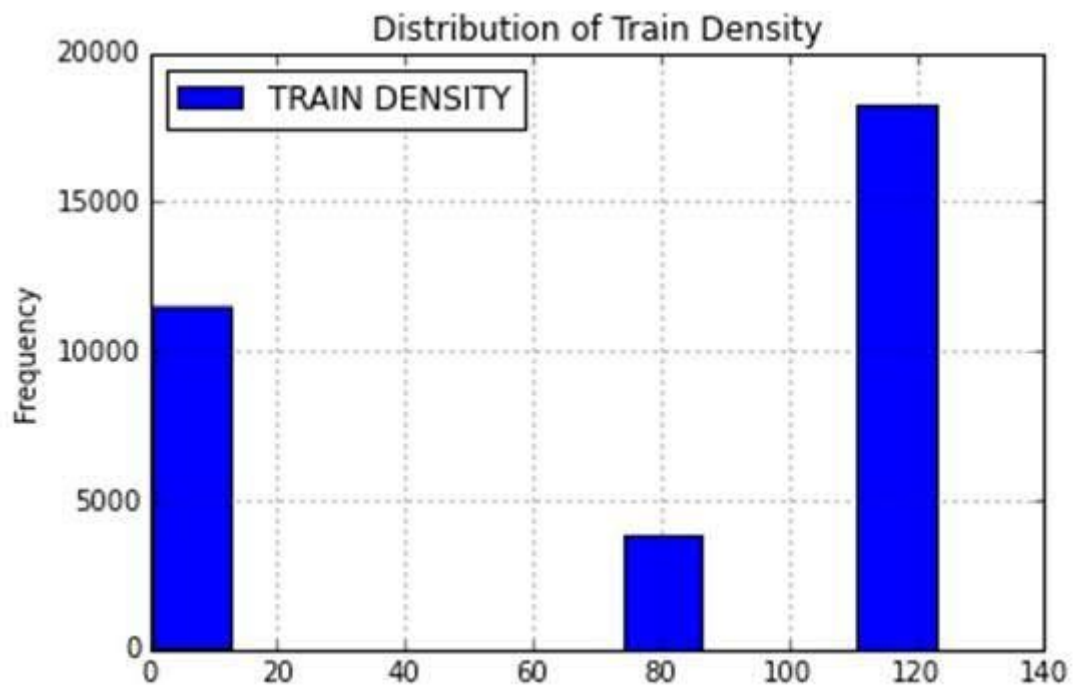


Figure 27 - The distribution of train density statistics

The dependency of dwell time on the train density statistic is illustrated in figure 28. There is a small but statistically significant negative correlation here, with increased train density correlating with a slight reduction in dwell time. The Pearson correlation coefficient was calculated at -0.0077 with a p-value of 7.7e-45.

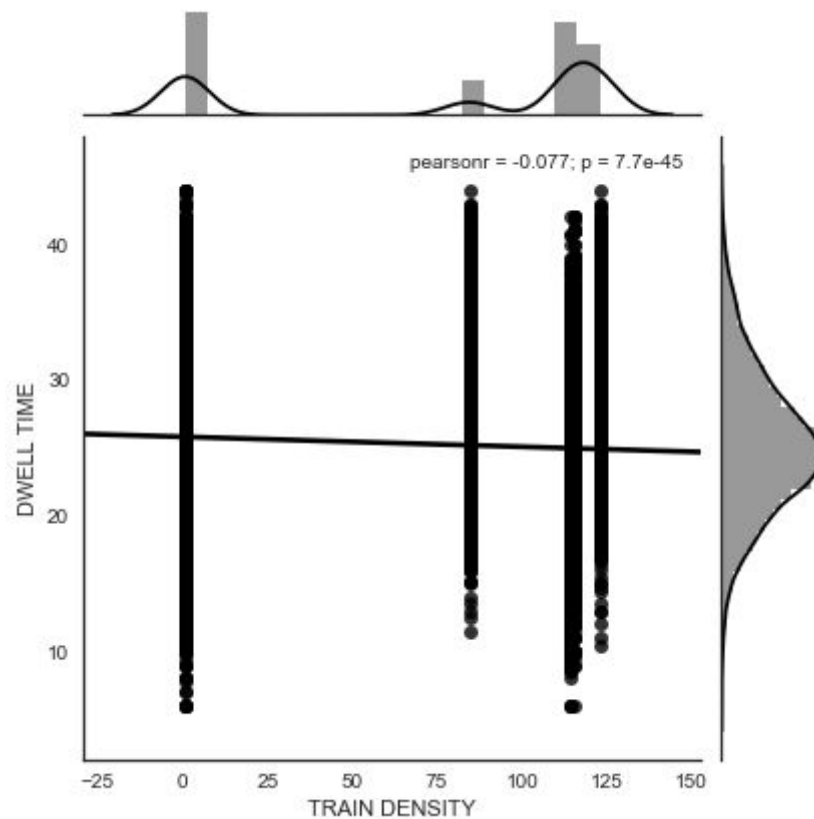


Figure 28 - The dependency of dwell time on train density

The Pearson correlation coefficient is a measure of the linear relationship between two variables. The Spearman correlation coefficient is a measure of the strength of any monotonic relationship between two variables. Both correlation metrics were calculated between each possible factor and the mean dwell times. All results are statistically significant to 99% confidence levels due to the large sample sized used.

7.1.2.2 Correlation Results for Factor Selection

Table 14 presents Spearman and Pearson correlation scores between tested factors and mean dwell time.

Factor	Pearson Correlation with Mean Dwell Time	Spearman Correlation with Mean Dwell Time
Mean Dwell Time	1	1
Mean Boarders and Alighters	0.597	0.602
Service Affecting Failure Rate for Line and Direction of Travel	0.267	0.252
Train Density on the Line	-0.077	-0.033
Standing Capacity of Train	-0.027	-0.081
Seating Capacity of Train	-0.191	-0.124

Table 14 - Pearson and spearman correlation coefficient results for different factors

7.1.3 Regression Analysis Results

The coefficient of determination, R^2 , is calculated for each model. This value provides a score for the proportion of variance in the dwell time that is explained by the model and the input variables. A number of regression models were tested.

7.1.3.1 Multiple Linear Regression

The multiple linear regression model was calculated as:

$$y = 0.0063x_1 + 0.4124x_2 - 0.0189x_3 + 0.0008x_4 - 0.0064x_5 + 23.646$$

Where:

y = Dwell Mean

x_1 = Total Boarders and Alighters

x_2 = SAF Rate

x_3 = Seating Capacity

x_4 = Standing Capacity

x_5 = Train Density

The R^2 value for this multiple linear regression fit was calculated to be 0.434 with the validation correlation score found to be 0.658.

7.1.3.2 Decision Tree Regression

The maximum depth was varied and can be seen plotted in figure 29 against the r-squared value. The data can also be found in table 15.

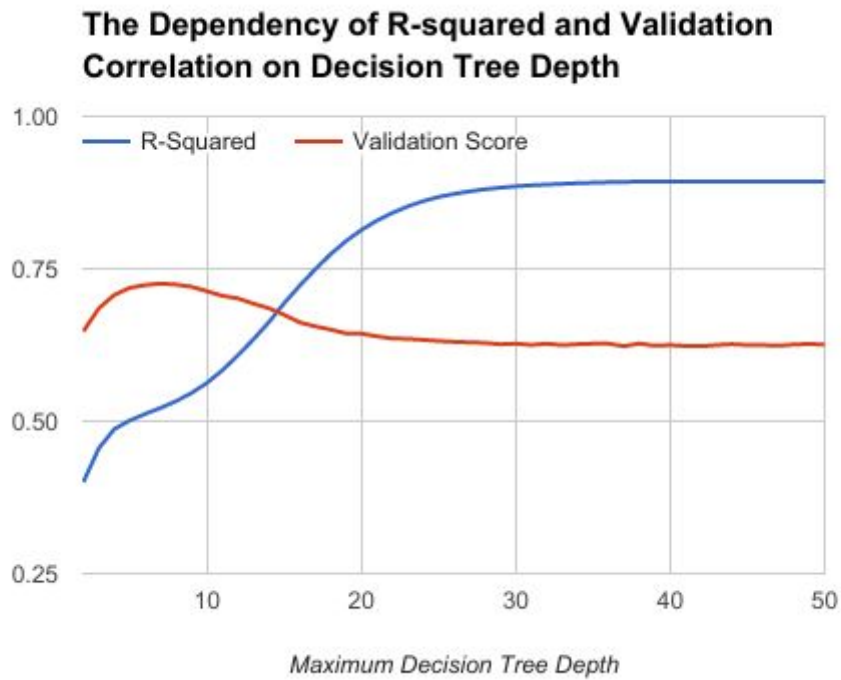


Figure 29 - The dependency of R-squared and validation correlation on decision tree depth

The R^2 value for the decision tree regression model was maximised at 0.89 with a tree depth of 30. However this resulted in a model that was overfitted as the validation score was lowest at this point with a correlation of 0.63.

The best model had a validation correlation coefficient of 0.73 between the test dwell means and the predicted dwell means. The decision tree depth for this model was 7. The R^2 value of this model was found to be 0.52.

Maximum Decision Tree Depth	R-Squared	Validation Score
2	0.40	0.65
3	0.46	0.69

4	0.49	0.71
5	0.50	0.72
6	0.51	0.72
7	0.52	0.73
8	0.53	0.72
9	0.55	0.72
10	0.56	0.71
11	0.58	0.71
12	0.61	0.70
13	0.63	0.69
14	0.66	0.69
15	0.69	0.67
16	0.72	0.66
17	0.75	0.66
18	0.77	0.65
19	0.80	0.64
20	0.81	0.64
21	0.83	0.64
22	0.84	0.64
23	0.85	0.64
24	0.86	0.63
25	0.87	0.63
26	0.87	0.63
27	0.88	0.63
28	0.88	0.63
29	0.88	0.63
30	0.89	0.63
31	0.89	0.62
32	0.89	0.63

33	0.89	0.62
34	0.89	0.63
35	0.89	0.63
36	0.89	0.63
37	0.89	0.62
38	0.89	0.63
39	0.89	0.62
40	0.89	0.62
41	0.89	0.62
42	0.89	0.62
43	0.89	0.62
44	0.89	0.63
45	0.89	0.62
46	0.89	0.62
47	0.89	0.62
48	0.89	0.63
49	0.89	0.63
50	0.89	0.63

Table 15 - Decision tree testing results

7.1.3.3 K-nearest Neighbours Regression

The default parameters for the model were used from SciKit learn library (SciKit-Learn 2016b). This resulted in a leaf size of 30 using the minkowski method with a power of 2. Uniform weights were tested as well as inverse distance weights. The number of neighbours was varied in order to find the optimal R^2 value and the results shown in figure 30 and table 16.

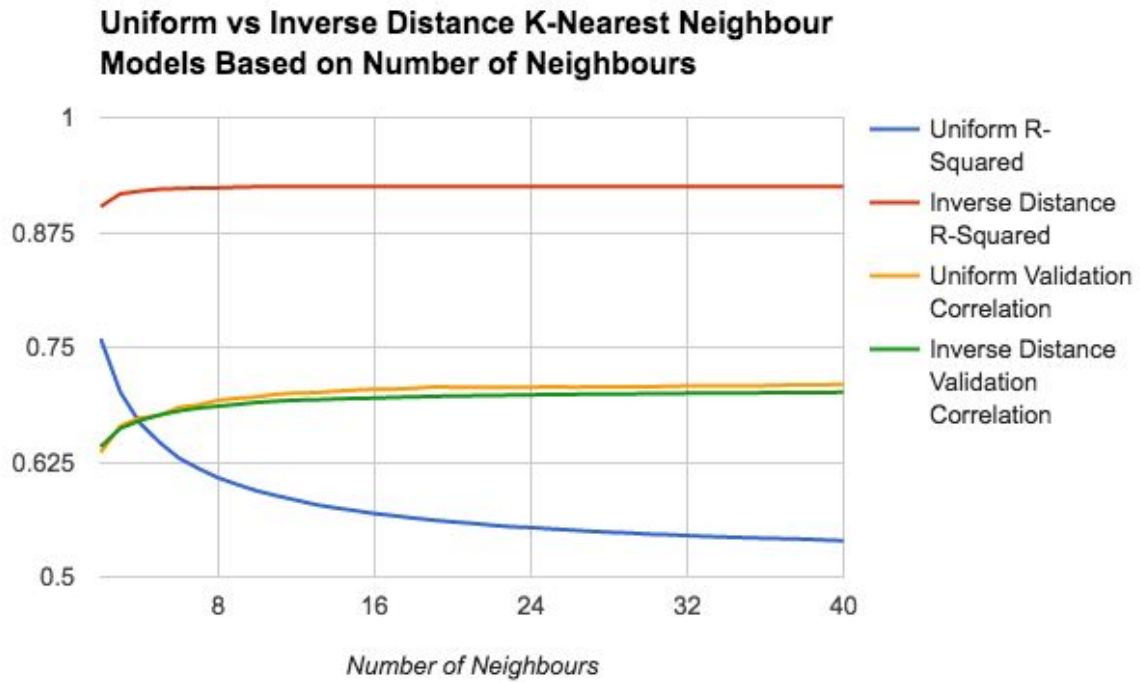


Figure 30 - Uniform vs inverse distance K-nearest neighbour models based on the number of neighbours

N_neighbours	R-squared		Validation Correlation	
	Uniform R-Squared	Inverse Distance R-Squared	Uniform Validation Correlation	Inverse Distance Validation Correlation
2	0.76	0.90	0.64	0.64
3	0.70	0.92	0.66	0.66
4	0.67	0.92	0.67	0.67
5	0.65	0.92	0.68	0.68
6	0.63	0.92	0.68	0.68
7	0.62	0.92	0.69	0.68
8	0.61	0.92	0.69	0.69
9	0.60	0.92	0.69	0.69
10	0.59	0.92	0.70	0.69
11	0.59	0.92	0.70	0.69
12	0.58	0.92	0.70	0.69

13	0.58	0.92	0.70	0.69
14	0.58	0.92	0.70	0.69
15	0.57	0.92	0.70	0.69
16	0.57	0.93	0.70	0.69
17	0.57	0.93	0.70	0.70
18	0.56	0.93	0.71	0.70
19	0.56	0.93	0.71	0.70
20	0.56	0.93	0.71	0.70
21	0.56	0.93	0.71	0.70
22	0.56	0.93	0.71	0.70
23	0.55	0.93	0.71	0.70
24	0.55	0.93	0.71	0.70
25	0.55	0.93	0.71	0.70
26	0.55	0.93	0.71	0.70
27	0.55	0.93	0.71	0.70
28	0.55	0.93	0.71	0.70
29	0.55	0.93	0.71	0.70
30	0.55	0.93	0.71	0.70
31	0.55	0.93	0.71	0.70
32	0.54	0.93	0.71	0.70
33	0.54	0.93	0.71	0.70
34	0.54	0.93	0.71	0.70
35	0.54	0.93	0.71	0.70
36	0.54	0.93	0.71	0.70
37	0.54	0.93	0.71	0.70
38	0.54	0.93	0.71	0.70
39	0.54	0.93	0.71	0.70
40	0.54	0.93	0.71	0.70

Table 16 - Uniform vs inverse distance K-nearest neighbour models based on the number of neighbours

The R^2 value for the decision tree regression model was maximised at 0.92 using the inverse distance weights method with 6 neighbours. However this has clearly produced a model that is overfitted to the data since the validation score is 0.68. This validation score improves with a greater number of neighbours and by changing the method to uniform weights. The uniform weights model produces the best validation correlation score of 0.71 with 18 neighbours in the model and with a R-squared value of 0.54.

7.1.3.4 Neural Network Regression

The number of hidden layers was varied from 10 to 200 and the results plotted below in figure 31. The results can also be seen in table 17.

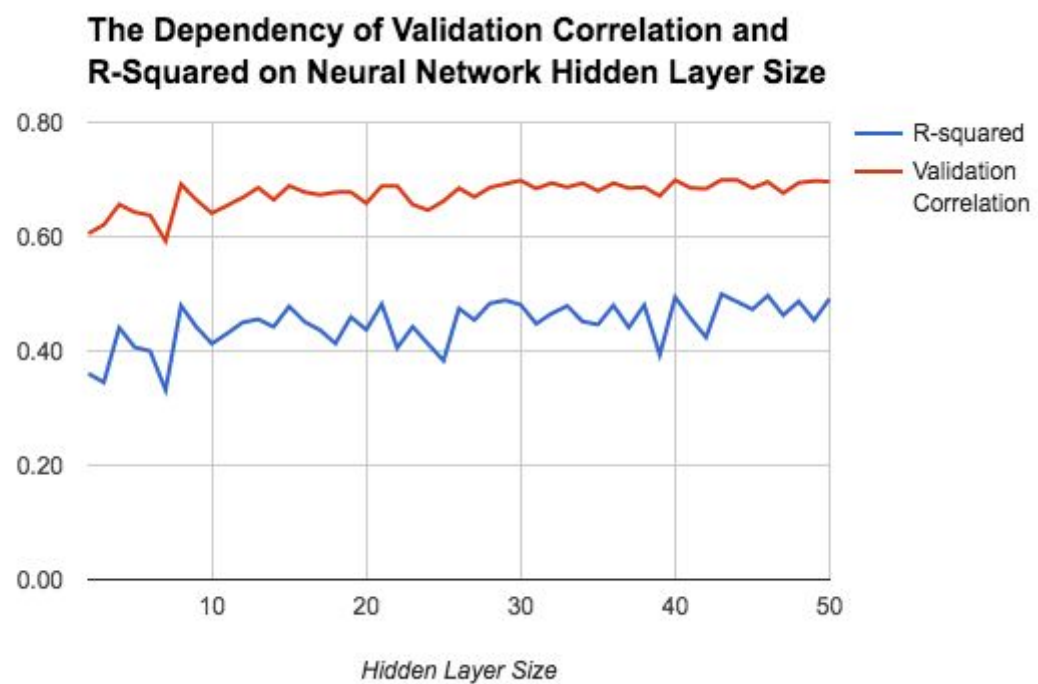


Figure 31 - The dependency of validation correlation and r-squared on neural network nidden layer size

Number of Units per Layer	R-squared	Validation Correlation
2	0.36	0.61
3	0.34	0.62
4	0.44	0.66
5	0.41	0.64
6	0.40	0.64
7	0.33	0.59
8	0.48	0.69
9	0.44	0.66
10	0.41	0.64
11	0.43	0.65
12	0.45	0.67
13	0.46	0.69
14	0.44	0.66
15	0.48	0.69
16	0.45	0.68
17	0.44	0.67
18	0.41	0.68
19	0.46	0.68
20	0.44	0.66
21	0.48	0.69
22	0.41	0.69
23	0.44	0.66
24	0.41	0.65
25	0.38	0.66
26	0.47	0.68
27	0.45	0.67
28	0.48	0.69
29	0.49	0.69

30	0.48	0.70
31	0.45	0.68
32	0.47	0.69
33	0.48	0.69
34	0.45	0.69
35	0.45	0.68
36	0.48	0.69
37	0.44	0.68
38	0.48	0.69
39	0.39	0.67
40	0.49	0.70
41	0.46	0.69
42	0.42	0.68
43	0.50	0.70
44	0.49	0.70
45	0.47	0.69
46	0.50	0.70
47	0.46	0.68
48	0.49	0.69
49	0.45	0.70
50	0.49	0.70

Table 17 - The dependency of validation correlation and r-squared on neural network hidden layer size

The neural network model validation correlation score is maximised at 0.7 at several points. The smallest hidden layer size where this score is maximised is found with 30 hidden layers. The R^2 value with this model was found to be 0.48.

7.1.3.5 Regression Results Summary

The R^2 score provides a measure of how well the model fits the training data. The validation correlation score provides a measure of how generalisable and thus valid the model is. A summary of regression models tested is presented in table 18.

Regression Model	R-squared Score (p = 0 for all results)	Best Validation Correlation Score (p = 0 for all results)
Multiple Linear Regression	0.43	0.66
K-Nearest Neighbours	0.56	0.71
Decision Tree	0.73	0.52
Neural Network	0.48	0.7

Table 18 - Summary results of regression models tested

7.2 Victoria Line Simulation Case Study Results

The mean headway at each station was calculated and an equivalent trains per hour (TPH) value calculated from this. Since each year up to 2050 was tested, an equivalent forecast for the capacity of the line was made. The results can be seen in figure 32.

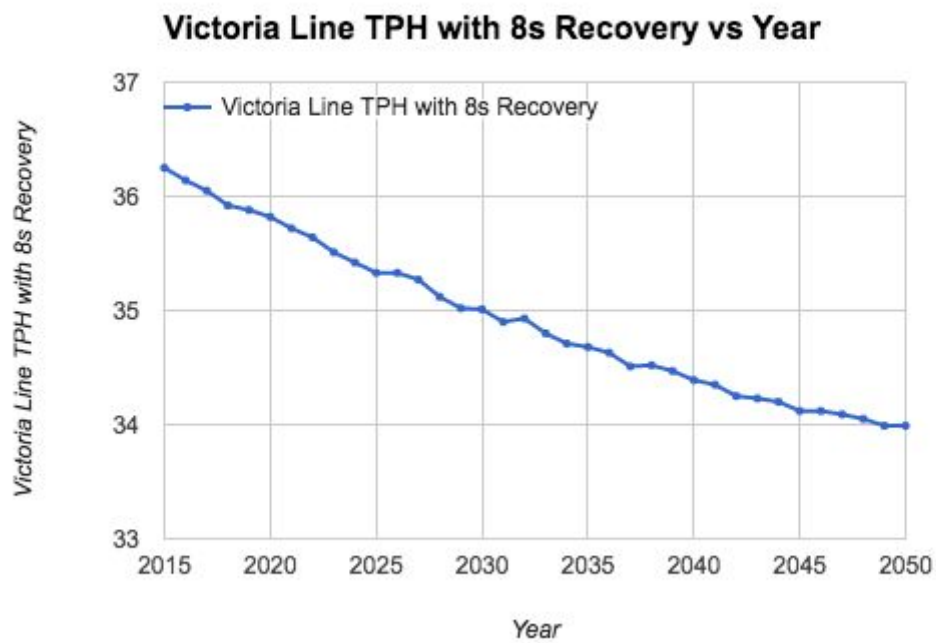


Figure 31 - Predicted Victoria Line achievable capacity as London population increases

The TPH results show a steady decrease in achievable capacity on the line as the population increases based on the assumption of a linear increase in boarding and alighting numbers with population.

With 2015 population levels and therefore 2015 numbers of boarders and alighters the breakdown of capacity at each station can be seen in figure 33.

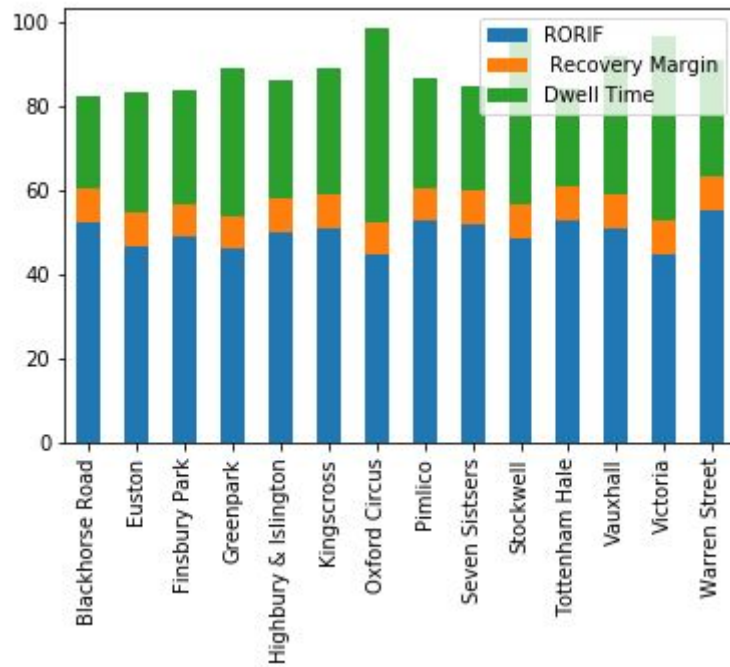


Figure 33 - Analysis of capacity constraints at each station on Victoria Line in 2015

The maximum RORIF + recovery margin + mean dwell time is 98.5s at Oxford Circus, 98.4s at Stockwell and 96.6 at Victoria. This indicates that Oxford Circus is the bottleneck site to limiting capacity to 98.5 second headways or 36.55 TPH.

With the 2050 population forecast and assuming that this translates linearly to an increase in passenger numbers at each station, the individual station results can be seen in figure 34.

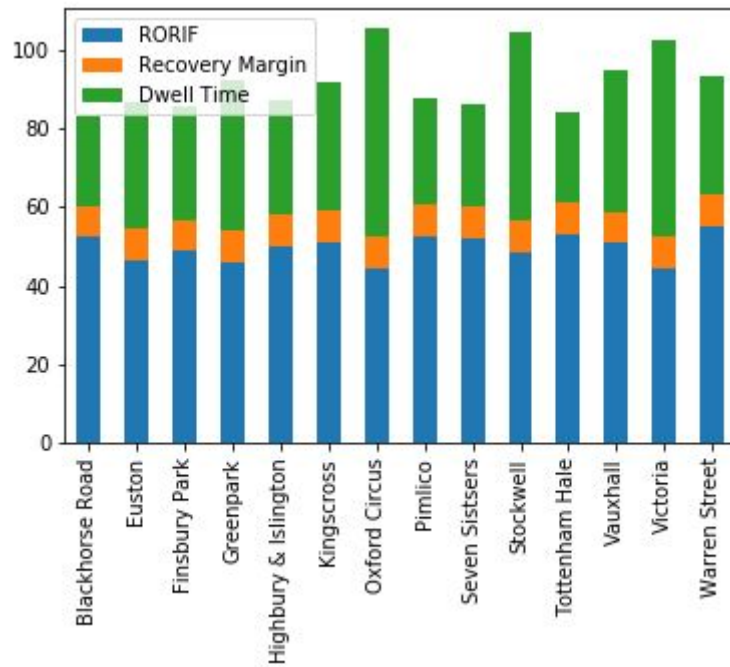


Figure 34 - Analysis of capacity constraints at each station on Victoria Line in 2050

Oxford Circus had a total limiting time of 105.5s, Stockwell had 104.4s and Victoria 102.6. This indicates that Oxford Circus provides the bottleneck limiting capacity to 105.5 second headways or 34.12 TPH.

8 Discussion

8.1 Dwell Time Modelling Analysis

8.1.1 Distribution Analysis

The top five distributions with the highest mean R^2 score were presented in the results section. It was found that the Gamma distribution had the highest mean R^2 value and is thus the most suitable distribution for estimating dwell times in simulations. The Gamma distribution is also suitable as it cannot produce negative values. This is important in the context of simulation, since the dwell time cannot be a negative value. Having a continuous distribution which presents the possibility of negative values would mean an invalid simulation.

The top five distributions, the Power Normal, Johnson SU and non-central Student's T distribution all present the possibility of negative numbers when sampling from them. Therefore these distributions should be discounted as possibilities for simulation.

If the Gamma distribution cannot be estimated or is not desired to be used for the primary dwell time distribution the next best choice would be the Gilbrat distribution. This scored highly with a mean R^2 value of 0.79 and operates in the region greater than zero.

It should be noted that the normal distribution was found to be an unsuitable choice with a mean R^2 score of 0.6. The lognormal distribution has a mean R^2 score of 0.73 suggesting it is more appropriate than the normal distribution. Due to its popularity there may be a temptation in the simulation of train systems to use the normal distribution for the dwell time. The results of this study showing a poor estimation for the dwell time relative to other distributions and the potential for negative number sample show that the normal distribution is not recommended for sampling the dwell time.

8.1.2 Factor Analysis

The factor analysis yielded positive results with a range of different factors showing a correlation with the mean dwell time. The mean number of boarders and alighters was by far the strongest predictor of mean dwell time. This makes sense from a simple crowding perspective; more crowding logically leads to an increased amount of time for the passengers to board or alight.

The second strongest factor was the SAF rate. This was an interesting result. As discussed in the methodology chapter, SAFs are only recorded in the data as delays of greater than two minutes. The dwell time data has been trimmed to exclude data points of greater than 45 seconds time. Thus it might at first glance seem to be illogical that the SAF rate of delays greater than two minutes positively correlates with the mean dwell times of data trimming dwells greater than 45 seconds. The answer to this is found in the knock on effect of SAFs. When a SAF occurs, the impact of the incident is usually not contained to a single location. System-level effects occur from train delays as the impact of the delay cascades down the line. Since one train is delayed, trains behind this one cannot advance. A concertina-like effect thus takes place and even after the delay is over the service will not recover instantaneously. Thus, due to recovery after an incident, an increase in the SAF rate on a line logically translates to an increase in the *mean* dwell time.

The train density statistic was shown to slightly negatively correlate with dwell time. This makes sense, since an increased density of trains on a line is likely to result in a more frequent service and thus a lower number of boarders and alighters. There may be a psychological effect of a greater train density, since passengers know that missing one train will result in another arriving shortly. This may dispose passengers to be less inclined to force their way onto a busy train, thus reducing the mean dwell time.

The standing capacity and seating capacity of the train was shown to negatively correlate with mean dwell time. This makes sense since having trains with a higher capacity means that there is more room for passengers to maneuver on and off the train. What was interesting was that seating capacity was shown to be a stronger determinant of lower dwell times than standing capacity. It is not clear why this is the case and could be due to increased flow rates associated with trains with a greater number of seats. However it is not possible to prove this mechanism without laboratory testing of passenger boarding and alighting behaviour.

8.1.3 Regression Analysis

Several regression models were tested for deriving the mean dwell time from the selected input factors.

The multiple linear model produced a relative strong model with a validation correlation score of 0.66. The main advantage of the this model is simplicity, it is possible to communicate the results in a straightforward equation. However, other models produced a higher validation score and thus from a technical standpoint should be favoured.

An inverse distance weighted K-nearest neighbours model delivered the highest R^2 value of 0.92. This value was maximised with 6 neighbours using the method of inverse distance weights. However as pointed out in the methodology chapter, the inverse distance weights method is more prone to overfitting the data, which may explain the high R^2 value. Comparing the inverse distance weighting results to the uniform weighting results it is clear that overfitting is occurring. While the R^2 value of 0.92 appears attractive in the inverse distance weights model, there is actually a small loss of validation performance when comparing the model to the uniform weights model. The highest performing K-nearest neighbours algorithm based on validation correlation was found to be with 18 neighbours and a validation correlation score of 0.71. This outperformed the multiple linear regression model.

This is one of the primary limitations of judging the performance of the regression model by the R^2 value alone. Non-parametric machine learning regression models, such as those tested here, are prone to overfitting. This produces high R^2 values for the sample data but does not necessarily mean the model generalises well. The validation correlation score thus takes precedence over the R^2 score when judging the validity of the models.

The decision tree models tested were the worst performers with the best model demonstrating a validation score of 0.52. It is thus not recommended that decision tree regression models are used for estimating the mean dwell time.

The best neural network model outperformed the linear regression model but underperformed the best K-nearest neighbours model. The validation correlation score for the neural network model was maximised at 0.7 with 30 hidden layers. The R^2 score for this model was 0.48.

The regression results show that it is possible to predict the mean dwell time from the given factors on the London Underground network with a good degree of accuracy. However it should be noted that it may not be possible to generalise this model to other networks which have different operating parameters and system designs. There are many factors which have not been investigated, for example, boarding and alighting from both sides of the train, the use of customer service assistants aiding the boarding and alighting process and the impact of different visual and audio aids. Metros around the world differ in the way they manage dwell times and this data has not been captured or considered in this study. For example some trains around the world utilise a system where passengers board from one side of the train and alight from another.

This analysis does provide strong evidence that it is possible to produce a generalisable model using a number of factors to predict the mean dwell time. The easiest way forward for railway operators would be to develop their own models specific for their network. The K-nearest neighbours regression model provides a strong starting point for future models to be built on.

8.2 Victoria Line Case Study Analysis

Rolling stock are generally designed for a 40 year life. The rolling stock on the Victoria Line was introduced in 2009, therefore upgraded trains are likely to not arrive until at least 2049. By this period if the number of boarders and alighters at each station increases linearly in proportion to the population change, the achievable TPH of the line may decrease from 36 trains per hour to 34 trains per hour.

The loss of capacity is purely due to the increases in the dwell time as a result of the assumed increased in the numbers of boarders and alighters. It should be noted that a significant limitation of these results is in this assumption. It is likely that a more complex distribution of increased passenger numbers takes place as the population of London increases.

The bottlenecks in the system were shown to be at Oxford Circus, Stockwell and Victoria. The constraint is primarily driven by a combination of the RORIF and the dwell time. Strategies to reduce the loss of capacity at these locations could be directed towards managing the dwell time or improving the run-in and run-out speeds.

A significantly limitation of these results is the assumption of a linear increase in the numbers of boarders and alighters with an increase in population. This relationship has not been validated and it is likely that the relationship is not that straightforward. For example, as passenger numbers increase, passengers may take different routes in order to avoid congested areas of the network. Additionally as technology and information flow improves, along with working practices enabling more flexible working times, the distribution of the number of boarders and alighters across the network could change.

An additional limitation of these results is the assumption that the full-speed run in time will always be utilised. In practice this is a worst case scenario, since the signalling system has the ability to dynamically modify train speeds. Thus it could be said that in practice the full-speed run in only occurs some of the time, and quicker run-in times with trains at slower speeds are likely to be utilised when trains need to be run closer together.

The results demonstrate however that a simple discrete event simulation framework can be employed to estimate capacity constraints on a line and identify bottlenecks. Employing the regression model for calculation of the mean dwell time enables the testing of various different inputs to the model.

9 Conclusions and Recommendations

The consideration of dwell times is important for the planning and optimisation of railway transport systems. This study has presented new evidence on the distribution which a dwell time follows for through running stations. This estimated distribution can be utilised in discrete-event simulation, for which this study presents a case study. This model and methodology is generalisable to other similar railway transport systems. Transport systems which differ significantly from the London Underground may require modelling of the dwell using data from those systems, however the methodology can remain consistent.

Dwell times on the London Underground network most commonly follow a gamma distribution. It is recommended that the gamma distribution is employed in simulations where the dwell time needs to be sampled from a continuous probability distribution.

A number of factors were tested and found to have a statistically significant correlation with the mean dwell time. The total number of boarders and alighters was found to have the strongest correlation with mean dwell time. The service affecting failure rate for the line positively correlates with the mean dwell time, likely due to the knock on effect of delays on the line causing mean dwell times to increase while the service recovers. The number of trains on the line relative to the number of stations and the line length, that is, the train density, was also found to negatively correlate with mean dwell times. This is hypothesised to be due to the psychological effect of having more frequent train services on dwell time. The seating capacity and standing capacity of the train negatively correlates with mean dwell times, with seating capacity being a stronger factor. Increasing train capacity may increase the flow of passengers on and off the train.

One significant limitation to this study is that the relationship, if any, between the factors and the variance of the dwell time has not been investigated. It could be the case that the factors identified and investigated positively or negatively correlate with the dwell time variance. For the purposes of this study, in particularly evident in the discrete event simulation of the Victoria Line, it has been assumed that variance of the dwell time remains constant. Future studies could investigate the effect of factors on dwelltime variance. This would be an important investigation as a greater variance in dwell times is likely to result in a less dependable service for customers.

Railway operators should develop their own dwell time models that are specific to their own network. There is scope for a more complex model which amalgamates data from different rail

networks in order to discover the true factors that influence dwell times and tell the rest of the story.

This study has demonstrated the method of testing several regression models and using the pearson correlation coefficient to test predicted values against a partitioned validation dataset. The results indicate that the K-nearest neighbours regression model is the best performing algorithm for forecasting mean dwell times. Neural network decision tree models were also tested. The decision tree models performed the worst and the neural network models performed well but not as well as the K-nearest neighbours model. A simple linear regression can also be employed with less power than K-nearest neighbours and neural networks models but with more power than the decision tree network. The advantage of using the simple linear regression approach is found in the simplicity of the model which makes communication and sharing straightforward. There are many different parameters for these models which were not investigated, as well as many other regression models, it is highly likely that a better solution exists for which regression model to utilise. Further work is required to investigate which regression model to employ with the appropriate parameters. This will also need to be tested on a different dataset and the choice of which model is best could change if the number of factors changes.

This study has demonstrated that it is possible to save a regression model and import into a discrete event simulation for the capacity analysis of a line. This is useful in the context of testing different parameters such as an increase in the number of boarders and alighters or a change in train factors such as seating or standing capacity. A number of factors which were identified by the literature were not investigated. A more comprehensive regression model would involve more factors, which would enable the optimisation of railway design at a system level.

The capacity of the Victoria Line was assessed assuming a linear relationship between the number of boarders and alighters and the population change of London forecast up to 2050. The multiple linear regression model established in this study was employed. This model was chosen for its simplicity, however in practice a K-nearest neighbours regression model may produce more accurate dwell time results. Given these assumptions the peak capacity constraint on the Victoria Line was shown to decrease from 36 to 34 trains per hour with an 8 second recovery margin built into the RORIF time. Mitigating this loss in capacity could involve reducing the dwell time or improving the RORIFs at the bottleneck locations. The primary limitation of this study is the assumption of the linear increase in boarders and alighters with the London population. In practice it is unlikely that such a linear relationship exists.

This study has effectively demonstrated how a factor analysis specific to one operators rail network can lead to a regression model that can be directly employed in a discrete event simulation in order to test parameters at a system level for a rail network. This simulation can be used for system-level design and forecasting of performance into the future. The simulation can also identify bottlenecks as a way of optimising the allocation of resources for improvement works.

New areas of investigation could seek to build on this dwell time modelling by expanding the dataset to include more factors and data from other metros and railways around the world. A larger dataset affords the possibility to generate more robust models of the dwell time. This will lead to greater accuracy in the modelling of dwell times, thus enabling precise assessment of future transport system capacity.

10 References

- Community of Metros and Imperial College London. 2013. "Dwell Time Recalibration." Imperial College London.
- . 2015. "Dwell Time Management." Imperial College London.
- Edwards Deming, W. 2002. *Out of the Crisis*.
- Engineer, Rail. 2013. "Squeezing More from the Tube." *Rail Engineer Web Site*. May 8.
<http://www.railengineer.uk/2013/05/08/squeezing-more-from-the-tube/>.
- EveningStandard. 2015. "Victoria Line Work Sets 'Gold Standard' with 36 Trains an Hour," August 14.
<http://www.standard.co.uk/news/transport/victoria-line-work-sets-gold-standard-with-36-trains-an-hour-a2633621.html>.
- GLA. 2015. "2015 Round Population Projections."
https://files.datapress.com/london/dataset/2015-round-population-projections/2016-10-21T14:23:54/long_term_trend_2015_round.xlsx.
- Goodwin, T. 2015. "Changing Behaviours on the London Underground: A Journey into Dwells." MSc, University College London.
- Grube, Pablo, Felipe Núñez, and Aldo Cipriano. 2011. "An Event-Driven Simulator for Multi-Line Metro Systems and Its Application to Santiago de Chile Metropolitan Rail Network." *Simulation Modelling Practice and Theory* 19 (1): 393–405.
- Hollocks, B. W. 2005. "Forty Years of Discrete-Event Simulation — a Personal Reflection." *The Journal of the Operational Research Society* 57 (December): 1383 to 1399.
- Holloway, C., T. Roan, and N. Tyler. 2013. "New Deep Tube Train: Design Features Affecting Boarding and Alighting of Passengers." University College London.
- Kass, A. H. 1998. "Methods to Calculate Capacity of Railways." Technical University of Denmark.
- Kingma, Diederik P., and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization."
<http://arxiv.org/abs/1412.6980>.
- Kontaxi, Evangelia, and Stefano Ricci. 2009. "Techniques and Methodologies for Railway Capacity Analysis: Comparative Studies and Integration Perspectives." presented at the 3rd International Seminar on Railway Operations Modelling and Analysis, Sapienza Università di Roma.
- Landex, Alex. 2012. "Reliability of Railway Operation." In *Artikler Fra Trafikdage Pa Aalborg Universitet*. Aalborg University.
- Li, Dewei, Winnie Daamen, and Rob M. P. Goverde. 2016. "Estimation of Train Dwell Time at Short Stops Based on Track Occupation Event Data: A Study at a Dutch Railway Station." *Journal of Advanced Transportation* 50 (5): 877–96.
- Lu, Quan, Maged Dessouky, and Robert C. Leachman. 2004. "Modeling Train Movements through Complex Rail Networks." *ACM Transactions on Modeling and Computer Simulation* 14 (1): 48–75.
- Motraghi, Adam, and Marin Varbanov Marinov. 2012. "Analysis of Urban Freight by Rail Using Event Based Simulation." *Simulation Modelling Practice and Theory* 25: 73–89.
- Mukkamala, R., S. Lakkoju, V. Kamineni, S. Kamisetty, A. Polu, and J. Creedon. 2008. "Improving Runway Capacity: An Integrated Approach Using Modeling, Simulation, and Analysis." Norfolk, VA 23529: Old Dominion University.
- Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. MIT Press.
- Nash, A., and D. Huerlimann. 2004. "Railroad Simulation Using OpenTrack." In , edited by J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, and S. Sone, 45. WITpress.
- Nunez, Felipe, F. Reyes, P. Grube, and A. Cipriano. 2010. "Simulating Railway and Metropolitan Rail Networks: From Planning to On-Line Control." *IEEE Intelligent Transportation Systems Magazine* 2 (4): 18–30.
- Ox, J., and E. Goldratt. 1986. *The Goal: A Process of Ongoing Improvement*. Croton-on-Hudson: North

- River Press.
- Paolucci, M., and R. Pesenti. 1999. "An Object-Oriented Approach to Discrete-Event Simulation Applied to Underground Railway Systems." *Simulation* 72 (6): 372–83.
- SciKit-Learn. 2016a. "Nearest Neighbors Regression — Scikit-Learn 0.18.1 Documentation." http://scikit-learn.org/stable/auto_examples/neighbors/plot_regression.html.
- . 2016b. "sklearn.neighbors.KNeighborsRegressor — Scikit-Learn 0.18.1 Documentation." <http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html>.
- . 2016c. "sklearn.tree.DecisionTreeRegressor — Scikit-Learn 0.18.1 Documentation." <http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html#sklearn.tree.DecisionTreeRegressor>.
- scikit-learn. 2017. "Scikit-Learn: Machine Learning in Python — Scikit-Learn 0.18.1 Documentation." *Scikit-Learn Website*. <http://scikit-learn.org/>.
- SciPy. 2014. "SciPy.stats.rv_continuous.fit — SciPy v0.14.0 Reference Guide." May 11. https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.rv_continuous.fit.html.
- . 2017a. "SciPy.stats.probplot — SciPy v0.19.0 Reference Guide." March 9. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.probplot.html>.
- . 2017b. "Statistical Functions (scipy.stats) — SciPy v0.19.0 Reference Guide." March 9. <https://docs.scipy.org/doc/scipy/reference/stats.html>.
- Siefer, T. 2008. "Simulation." *Railway Timetable & Traffic* 1: 155 to 169.
- TfL. 2015. "TfL Mayor's Budget 2016/2017." Transport for London.
- . 2016. "LU Delivers Mayor's Target of 30% Reduction in Delays." *Transport for London Web Site*. January 28. <https://tfl.gov.uk/info-for/media/press-releases/2016/january/lu-delivers-mayor-s-target-of-30-reduction-in-delays>.
- Tortorella, Michael. 2015. *Reliability, Maintainability, and Supportability*.
- UN. 2016. "The World's Cities in 2016." UN. http://www.un.org/en/development/desa/population/publications/pdf/urbanization/the_worlds_cities_in_2016_data_booklet.pdf.
- Wikipedia. 2017. "Curse of Dimensionality - Wikipedia." March 12. https://en.wikipedia.org/wiki/Curse_of_dimensionality.
- Wong, H., and A. Key. 2014. "Tapir Phase 1 Report." Transport for London, Strategy & Service Development.

11 Appendices

11.1 Appendix A - Code Used for Processing

Written in Python 3.

Also available for download at github: <https://github.com/harrymunro/MSc-Code>.

11.1.1 Script for Merging raw NETMIS and RODS datasets

```
"""
```

Created on Wed Dec 14 11:48:35 2016

This script pulls in raw NETMIS data and RODS data.

It then does the following:

- Filters out unnecessary stations from a station exclusions list.
- Calculates dwell times.
- Optionally filters dwell times.
- Excludes unwanted lines.
- Matches passenger numbers with stations and time of day.
- Adds SAF rate for each line.
- Adds train density for each line.
- Adds standing capacity for each line.
- Adds seating capacity for each line.
- Calculates availability of each line.
- Populates line names.
- Saves the new dataset.

```
"""
```

```
import pandas as pd
```

```
import numpy as np
```

```
import time
```

```
start_time = time.time()
```

```
netmis_file = 'November 2015 NETMIS Data Dates Version.csv'
```

```
#df = pd.read_csv(netmis_file, usecols = ['TIMESTAMP',
```

```

#'ACTUAL DEPARTURE TIME', 'SUTOR CODE', 'LINE ID', 'DIRECTION CODE'],
#nrows = 100000) # with nrows filter

df = pd.read_csv(netmis_file, usecols = ['TIMESTAMP',
'ACTUAL DEPARTURE TIME', 'SUTOR CODE', 'LINE ID', 'DIRECTION CODE']) # with nrows filter

original = len(df)

# read in the station exclusions list
exclusions_file = 'station_exclusions.txt'
exclusions_list = pd.read_table(exclusions_file, index_col = 'Sutor')
# convert to dictionary
exclusions_list = exclusions_list.to_dict()['Include']
# map the exclusions list to sutor codes
df['INCLUDE'] = df['SUTOR CODE'].map(exclusions_list)
# delete rows containing "n"
df = df[df['INCLUDE'].str.contains('n') == False]

# drop rows where there is no dwell time data
df = df.dropna()

# convert timestamp data
df['TIMESTAMP'] = pd.to_datetime(df['TIMESTAMP'], format = '%d/%m/%Y %H:%M:%S')
df['ACTUAL DEPARTURE TIME'] = pd.to_datetime(df['ACTUAL DEPARTURE TIME'], format =
'%d/%m/%Y %H:%M:%S')

# calculating dwell times - need to convert timestamp data first
df['DWELL TIME'] = df['ACTUAL DEPARTURE TIME'] - df['TIMESTAMP']
df['SUTOR DIRECTION LINE'] = df['SUTOR CODE'] + df['DIRECTION CODE'].map(str) + df['LINE
ID'].map(str)

# Cleaning up again
df = df[df['DWELL TIME'] < pd.Timedelta('00:02:00')] # less than
df = df[df['DWELL TIME'] > pd.Timedelta('00:00:05')] # greater than

```

```

# deleting lines where we do not have RODs data
df = df[df['LINE ID'] != 11]
df = df[df['LINE ID'] != 14]

#names = list(df.columns.values)
col_list = ['TIMESTAMP', 'DWELL TIME', 'SUTOR CODE', 'LINE ID', 'DIRECTION CODE', 'SUTOR
DIRECTION LINE']
df = df[col_list]

# convert dwell times to integers
df['DWELL TIME'] = df['DWELL TIME'].dt.seconds

df = df.reset_index() # resets the index from

# now import the Alighters data
alighters_filename = 'Alighters.csv'
alighters_df = pd.read_csv(alighters_filename, skiprows = 2)
column_headers = list(alighters_df.columns.values)

# field which column timestamp falls into
alighters = [] # empty list to subsequently merge with dataframe
errors = []
match_errors = []
resampled_column_headers = column_headers[16:]
for row in range(len(df)):
    sample_time = df.TIMESTAMP[row]
    h = sample_time.hour
    m = sample_time.minute

# convert to strings
if h < 10:
    h = '0' + str(h)
else:
    h = str(h)

```

```

if m < 10:
    m = '0' + str(m)
else:
    m = str(m)

match = []
for n in resampled_column_headers:
    x = int(n[7:9]) # this is to remove the error of int(m[7:9]) equalling 0
    if x == 0:
        x = 59
    if n[0:2] == h and int(m) > int(n[2:4]) and int(m) <= x:
        match.append(n) # match contains the column heading for the match

if len(match) > 1:
    print('WARNING, MULTIPLE MATCHES FOUND')

if len(match) == 0:
    match_errors.append([sample_time, row])

sample_s_d_l = df['SUTOR DIRECTION LINE'][row]
# test if sample_s_d_l exists in the alighters database
try:
    location = alighters_df[alighters_df['Sutor Direction Line'] == sample_s_d_l].index[0]
    alighters.append(alighters_df[match[0]][location])
except IndexError:
    errors.append([sample_s_d_l])
    alighters.append(np.nan)

df['ALIGHTERS'] = alighters

# Now join the boarders data to dataset
boarders_filename = 'Boarders.csv'
boarders_df = pd.read_csv(boarders_filename, skiprows = 2)
column_headers_boarders = list(boarders_df.columns.values)

```

```

# field which column timestamp falls into
boarders = [] # empty list to subsequently merge with dataframe
resampled_column_headers = column_headers_boarders[17:]
for row in range(len(df)):
    sample_time = df.TIMESTAMP[row]
    h = sample_time.hour
    m = sample_time.minute

    # convert to strings
    if h < 10:
        h = '0' + str(h)
    else:
        h = str(h)

    if m < 10:
        m = '0' + str(m)
    else:
        m = str(m)

    match = []
    for n in resampled_column_headers:
        x = int(n[7:9]) # this is to remove the error of int(m[7:9]) equalling 0
        if x == 0:
            x = 59
        if n[0:2] == h and int(m) > int(n[2:4]) and int(m) <= x:
            match.append(n) # match contains the column heading for the match

    if len(match) > 1:
        print('WARNING, MULTIPLE MATCHES FOUND')

    if len(match) == 0:
        match_errors.append([sample_time, row])

    sample_s_d_l = df['SUTOR DIRECTION LINE'][row]
    # test if sample_s_d_l exists in the alighters database

```



```

#x = alighters_df['Sutor Direction Line'] == sample_s_d_l
try:
    location = borders_df[borders_df['Sutor Direction Line'] == sample_s_d_l].index[0]
    borders.append(borders_df[match[0]][location])
except IndexError:
    errors.append([sample_s_d_l])
    borders.append(np.nan)

df['BOARDS'] = borders

df = df.dropna()

# calculate borders and alighters
df['BOARDS AND ALIGHTERS'] = df['BOARDS'] + df['ALIGHTERS']

# populate with line names
line_names = {0:'Bakerloo', 2:'Central', 3:'Victoria', 4:'Metropolitan', 5:'Northern', 6:'Jubilee',
7:'Piccadilly', 8:'District', 13:'Circle Hammersmith & City'}
df['LINE NAME'] = df['LINE ID'].map(line_names)

# populate with failure data - safs per day per line
saf_rate = {'Bakerloo':6.3886121, 'Central': 14.29395018, 'Victoria': 6.844128113879, 'Metropolitan':
7.92918149466192, 'Northern': 13.0053380782918, 'Jubilee': 12.6241992882562, 'Piccadilly':
9.60747330960854, 'District': 9.30747330960854, 'Circle Hammersmith & City': 10.5996441281139}
df['SAF RATE'] = df['LINE NAME'].map(saf_rate)

# populate with train density (how many trains we have per station/km)
train_density = {'Bakerloo': 1.077586207, 'Central': 0.662162162, 'Victoria': 0.761904762,
'Metropolitan': 114.2941176, 'Northern': 122.96, 'Jubilee': 84.46666667, 'Piccadilly': 115.2075472}
df['TRAIN DENSITY'] = df['LINE NAME'].map(train_density)

# populate with standing capacity per train
standing_capacity = {'Bakerloo': 116.6, 'Central': 155.02, 'Victoria': 153.2, 'Metropolitan': 174,
'Northern': 110.36, 'Jubilee': 145.92, 'Piccadilly': 114}
df['STANDING CAPACITY'] = df['LINE NAME'].map(standing_capacity)

```

```

# populate with seating capacity per train
seating_capacity = {'Bakerloo': 268, 'Central': 272, 'Victoria': 252, 'Metropolitan': 306, 'Northern':
200, 'Jubilee': 234, 'Piccadilly': 228}
df['SEATING CAPACITY'] = df['LINE NAME'].map(seating_capacity)

# populate with line-direction availability
df['LINE DIRECTION'] = df['LINE NAME'] + df['DIRECTION CODE'].map(str)
availability = {'Bakerloo0': 0.992752, 'Bakerloo1': 0.9941, 'Central0': 0.9842, 'Central1':0.98,
'Victoria0': 0.9916, 'Victoria1': 0.9927, 'Metropolitan0': 0.9881, 'Metropolitan1': 0.9849, 'Northern0':
0.988, 'Northern1': 0.9879, 'Jubilee0': 0.9863, 'Jubilee1': 0.9884, 'Piccadilly0': 0.9885, 'Piccadilly1':
0.987}
df['LINE DIRECTION AVAILABILITY'] = df['LINE DIRECTION'].map(availability)

# Drop the district, circle and H&S
df = df.dropna()

# Data statistics
print(original - len(df))
print(len(df)/original)
print(time.time() - start_time)

df.to_csv('Populated NETMIS (including days) with RODS.csv')

```

11.1.1.2 Script to Resample the Merged Dataset

"""

Created on Wed Dec 14 21:15:48 2016

This code takes in a NETMIS-ish file and resamples by unique location.

"""

```

# script is currently working to return a dataframe with SUTOR CODES as columns, indexed by a
resampled timestamp

```

```

# takes in some NETMIS data
# deletes stations in sutor code list
# deletes rows where timestamp data is not available
# calculates dwell times

import pandas as pd
from datetime import datetime
import numpy as np
import time
start_time = time.time()

input_csv = 'Populated NETMIS (including days) with RODS.csv' ### MUST BE IN TIME FORMAT
yyy/mm/dd hh:mm:ss
print('read csv at time %d' % (time.time() - start_time))
# import dataset
df = pd.read_csv(input_csv)

df = df.drop('Unnamed: 0', 1)
df = df.drop('index', 1)

# calculate boarders and alighters
df['BOARDERS AND ALIGHTERS'] = df['BOARDERS'] + df['ALIGHTERS']

# convert timestamp data
df['TIMESTAMP'] = pd.to_datetime(df['TIMESTAMP'], format = '%Y-%m-%d %H:%M:%S')
df = df.sort(columns = 'TIMESTAMP')

# Trimming dwell outliers
df = df[df['DWELL TIME'] < 45] # optional

# Split into each sutor group
grouped = df.groupby(df['SUTOR DIRECTION LINE'])

# new dataframe

```

```

df2 = pd.DataFrame()

df['BOARDERS TO ALIGHTERS DIFFERENCE'] = df['BOARDERS'] - df['ALIGHTERS']
del df['ALIGHTERS']
del df['BOARDERS']
del df['LINE DIRECTION AVAILABILITY']

print('pruned dataframe at time %d' % (time.time() - start_time))

# Resample for each SUTOR group --- delete groups with too few data points
for group in df['SUTOR DIRECTION LINE'].unique(): # creates a list of tuples
    print('processing unique location %s at time %d' % (str(group), (time.time() - start_time)))
    sample = grouped.get_group(group)
    if len(sample) > 0.25 / len(df['SUTOR DIRECTION LINE'].unique()) * len(df): # only proceed for
"good" data based on size of dataframe and number of unique locations
        sample = sample.set_index('TIMESTAMP') # good
        #sample = sample.sort()
        sample = sample.resample('60T', how = 'mean') # mean dwell
        #sample = sample.resample('60T', how = {'DWELL TIME': np.var, 'BOARDERS AND ALIGHTERS':
np.mean, 'SAF RATE': np.mean, 'TRAIN DENSITY': np.mean, 'STANDING CAPACITY': np.mean,
'SEATING CAPACITY': np.mean}) # variance dwell
        sample['SUTOR DIRECTION LINE'] = group
        #df2 = pd.concat([sample, df2], axis = 1)
        #df2 = df2.rename(columns={'DWELL TIME': group})
        df2 = df2.append(sample)
        df2 = df2.dropna()# remove NAN rows

# optional: clean up dwell times
#df2 = df2[df2['DWELL TIME'] < 45] # less than
# write to csv
print('writing new csv at time %d' % (time.time() - start_time))
df2.to_csv('resampled_by_unique_location.csv')

```

11.1.3 Script to Test Goodness of Fit for Multiple Distributions

Take in a an input (e.g. a list of dwell times)

Tests against a number of distributions for goodness of fit

```
#import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import statsmodels.api as sm
```

```
import scipy.stats as stats
```

```
import math
```

```
import pandas as pd
```

```
# import data
```

```
f_name = "dwell.txt"
```

```
x = np.loadtxt(f_name)
```

```
y = np.log(x)
```

```
# mean and variance of data
```

```
var = np.var(x)
```

```
mean = np.mean(x)
```

```
# natural logarithms
```

```
var_log = np.var(y)
```

```
mean_log = np.mean(y)
```

```
print "Variance = %r" % var
```

```
print "Mean = %r" % mean
```

```
print "Variance log = %r" % var_log
```

```
print "Mean log = %r" % mean_log
```

```
##### lognormal paramaters #####
```

```
# mean and variance of distribution
```

```
lognormal_mean = math.exp(mean_log + 0.5 * var_log)
```

```
lognormal_var = math.exp(2 * mean_log + var_log) * (math.exp(var_log) - 1)
```

```

print lognormal_mean
print lognormal_var

# calculate shape parameter (complex way of re-arriving at var_log and mean_log)
mu = math.log(lognormal_mean/math.sqrt(1+(lognormal_var/lognormal_mean**2)))
sigma = math.sqrt(math.log(1+(lognormal_var/lognormal_mean**2)))

# plot
fig = plt.figure()
ax = fig.add_subplot(111)

# lognormal
lognormal_param = stats.lognorm.fit(x) # maximum likelihood fit

# weibull
weibull_param = stats.weibull_min.fit(x)

# beta
beta_param = stats.beta.fit(x)

# chi-squared
chi2_param = stats.chi2.fit(x)

# normal
norm_param = stats.norm.fit(x)

# log gamma
pareto_param = stats.pareto.fit(x)

# to get r-squared value need to do probplot[1][2] then square it

# function to automatically calculate r squared values
def get_r2_dist_fit(distribution, data):
    param = distribution.fit(data)
    probplot = stats.probplot(x, dist = distribution, sparams = param, fit = True, plot = ax)

```

```

print "R-squared value for %s is: %r\n" % (str(distribution), probplot[1][2]**2)
dist_name = str(distribution)
dist_name = dist_name[32:len(dist_name)-25]
return (dist_name, param, probplot[1][2]**2)

# all scipy distributions
DISTRIBUTIONS = [stats.alpha, stats anglit, stats arcsine, stats.beta, stats.betaprime, stats.bradford,
stats.burr, stats.cauchy, stats.chi, stats.chi2, stats.cosine, stats.dgamma, stats.dweibull, stats.erlang,
stats.expon, stats.exponnorm, stats.exponweib, stats.exponpow, stats.f, stats.fatiguelife, stats.fisk,
stats.foldcauchy, stats.foldnorm, stats.genlogistic, stats.genpareto, stats.gennorm, stats.genexpon,
stats.genextreme, stats.gausshyper, stats.gamma, stats.gengamma, stats.genhalflogistic,
stats.gilbrat, stats.gompertz, stats.gumbel_r, stats.gumbel_l, stats.halfcauchy, stats.halflogistic,
stats.halfnorm, stats.halfgennorm, stats.hypsecant, stats.invgamma, stats.invgauss, stats.invweibull,
stats.johnsonsb, stats.johnsonsu, stats.kstwobign, stats.laplace, stats.levy, stats.levy_l,
stats.levy_stable, stats.logistic, stats.loggamma, stats.loglaplace, stats.lognorm, stats.lomax,
stats.maxwell, stats.mielke, stats.nakagami, stats.ncx2, stats.ncf, stats.nct, stats.norm, stats.pareto,
stats.pearson3, stats.powerlaw, stats.powerlognorm, stats.povernorm, stats.rdist, stats.reciprocal,
stats.rayleigh, stats.rice, stats.recipinvgauss, stats.semicircular, stats.t, stats.triang, stats.truncexpon,
stats.truncnorm, stats.tukeylambda, stats.uniform, stats.vonmises, stats.vonmises_line, stats.wald,
stats.weibull_min, stats.weibull_max, stats.wrapcauchy]

results = []

for distribution in DISTRIBUTIONS:
    try:
        results.append(get_r2_dist_fit(distribution, x))
    except Exception:
        results.append((str(distribution)[32:len(str(distribution))-25], "Failed to fit", "n/a"))
        pass

df = pd.DataFrame(results)
df.to_csv("results.csv")
print "Distribution fit results successfully saved to csv file."

```

11.1.4 Regression Models Scripting

"""

Created on Sun Apr 16 13:13:29 2017

- Take in a dataset
- Partition into training and test data
- Fit regression model to training set
- Create predictions for test set
- Calculate pearson correlation coefficient

"""

```
import numpy as np
```

```
import pandas as pd
```

```
import scipy.stats as stats
```

```
filename = 'resampled_by_unique_location.csv'
```

```
df = pd.read_csv(filename)
```

```
df = df.sample(frac = 1) # randomise order of dataset
```

```
df_train = df.head(n = int(len(df) * 0.8))
```

```
df_test = df.tail(n = int(len(df) * 0.2))
```

```
# training time
```

```
y_train = np.array(df_train['DWEELL TIME'])
```

```
x1_train = df_train['BOARDERS AND ALIGHTERS']
```

```
x2_train = df_train['SAF RATE']
```

```
x3_train = df_train['SEATING CAPACITY']
```

```
x4_train = df_train['STANDING CAPACITY']
```

```
x5_train = df_train['TRAIN DENSITY']
```

```
X_train = np.array([x1_train, x2_train, x3_train, x4_train, x5_train])
```

```
X_train = X_train.T
```

```
# testing time
```

```
y_test = np.array(df_test['DWEELL TIME'])
```



```

x1_test = df_test['BOARDERS AND ALIGHTERS']
x2_test = df_test['SAF RATE']
x3_test = df_test['SEATING CAPACITY']
x4_test = df_test['STANDING CAPACITY']
x5_test = df_test['TRAIN DENSITY']

X_test = np.array([x1_test, x2_test, x3_test, x4_test, x5_test])
X_test = X_test.T

# linear regression model
from sklearn import linear_model
reg = linear_model.LinearRegression()
reg.fit(X_train, y_train)
score_ordinary_linear = reg.score(X_train, y_train)

predictions = []
for row in X_test:
    pred = reg.predict(row)
    predictions.append(pred[0])

y = []
for n in y_test:
    y.append(n)

pearson_corr_linear_regression = stats.pearsonr(y, predictions)

# decision tree

print("\nDECISION TREE RESULTS BELOW")
from sklearn import tree
reg = tree.DecisionTreeRegressor()
tree_scores = []
tree_correlations = []
for n in range(2, 51, 1):
    reg = tree.DecisionTreeRegressor(max_depth = n)

```

```

reg = reg.fit(X_train, y_train)
tree_score = reg.score(X_train, y_train)
predictions = []
for row in X_test:
    pred = reg.predict(row)
    predictions.append(pred[0])
pearson_corr_decision_tree = stats.pearsonr(y_test, predictions)
tree_scores.append(tree_score)
tree_correlations.append(pearson_corr_decision_tree)
print("\nR-SQUARED SCORES")
print(tree_scores)
print("\nCORRELATION SCORES")
print(tree_correlations)

# K Nearest Neighbours

from sklearn.neighbors import KNeighborsRegressor
neigh = KNeighborsRegressor()

knn_scores = []
knn_correlations = []
for n in range(2, 41):
    neigh = KNeighborsRegressor(n_neighbors = n, weights = 'uniform')
    reg = neigh.fit(X_train, y_train)
    s = reg.score(X_train, y_train)
    knn_scores.append(s)
    predictions = []
    for row in X_test:
        pred = reg.predict(row)
        predictions.append(pred[0])
    pearson_corr_knn = stats.pearsonr(y_test, predictions)
    knn_correlations.append(pearson_corr_knn)

#for score in knn_correlations:
    # print(score[0])

```

```

# neural network
from sklearn import neural_network
nn_scores = []
nn_correlations = []
for n in range(1, 51):
    reg = neural_network.MLPRegressor(hidden_layer_sizes = n)
    reg = reg.fit(X_train, y_train)
    nn_score = reg.score(X_train, y_train)
    nn_scores.append(nn_score)
    predictions = []
    for row in X_test:
        pred = reg.predict(row)
        predictions.append(pred[0])
    pearson_corr_nn = stats.pearsonr(y_test, predictions)
    nn_correlations.append(pearson_corr_nn)

```

11.1.1.5 Victoria Line Discrete Event Simulation

"""

Created on Thu Feb 9 10:14:48 2017

Simulation of the Victoria Line Northbound.

Macro level simulation of capacity.

Excludes inter-station run times.

Victoria0 = Victoria Northbound

Victoria1 = Victoria Southbound

Median standard deviation for all stations = 7.8.

"""

```
import simpy
```

```

from sklearn import linear_model
from sklearn.externals import joblib
import numpy as np

# import linear regression model
reg = joblib.load('linear_regression_model.pkl')

# simulation time
SIM_TIME = 1000000

# Parameters
population_multiplier = 1
SAF_RATE = 6.84
SEATING_CAPACITY = 252
STANDING_CAPACITY = 153.2
TRAIN_DENSITY = 0.76

# target TPH
target_tph = 36
recovery_margin = 8 # fixed recovery if interested in predicting TPH

# RORIFs and boarders and alighters for each station
stockwell = [48.4 + recovery_margin, 3070]
vauxhall = [50.8 + recovery_margin, 1669]
pimlico = [52.6 + recovery_margin, 570]
victoria = [44.6 + recovery_margin, 3305]
greenpark = [46.0 + recovery_margin, 1947]
oxfordcircus = [44.5 + recovery_margin, 3677]
warrenstreet = [55.1 + recovery_margin, 952]
euston = [46.5 + recovery_margin, 1164]
kingscross = [50.9 + recovery_margin, 1272]
highburyislington = [50.0 + recovery_margin, 888]
finsburypark = [48.8 + recovery_margin, 821]
sevensisters = [51.9 + recovery_margin, 459]
tottenhamhale = [53.0 + recovery_margin, 202]

```

```
blackhorseroad = [52.3 + recovery_margin, 100]
```

```
# headway analysis
```

```
def headway_analysis(times):
```

```
    times1 = times[:-1]
```

```
    times2 = times[1:]
```

```
    headways = []
```

```
    for n in range(len(times) - 1):
```

```
        headways.append(times2[n] - times1[n])
```

```
    return headways
```

```
def dwell(boarders_and_alighters):
```

```
    people = boarders_and_alighters * population_multiplier
```

```
    X = [people, SAF_RATE, SEATING_CAPACITY, STANDING_CAPACITY, TRAIN_DENSITY]
```

```
    dwell_mean = reg.predict(X)
```

```
    dwell_std = 7.8
```

```
    dwell_var = dwell_std**2
```

```
    # shape and scale paramaters for gamma distribution
```

```
    shape = dwell_mean**2 / dwell_var
```

```
    scale = dwell_var / dwell_mean
```

```
    t = np.random.gamma(shape, scale = scale)
```

```
    return t
```

```
def exponential_interval(inter):
```

```
    t = np.random.exponential(inter)
```

```
    return t
```

```
def source(env, stockwell_res, vauxhall_res, pimlico_res,
```

```
    victoria_res, greenpark_res, oxfordcircus_res,
```

```
    warrenstreet_res, euston_res, kingscross_res, highburyislington_res,
```

```
    finsburypark_res, sevensisters_res, tottenhamhale_res,
```

```
    blackhorseroad_res):
```

```
while True:
```

```
    env.process(route(env, stockwell_res, vauxhall_res, pimlico_res,
```

```

        victoria_res, greenpark_res, oxfordcircus_res,
        warrenstreet_res, euston_res, kingscross_res, highburyislington_res,
        finsburypark_res, sevensisters_res, tottenhamhale_res,
        blackhorseroad_res))

    yield env.timeout(exponential_interval(60))

def route(env, stockwell_res, vauxhall_res, pimlico_res,
        victoria_res, greenpark_res, oxfordcircus_res,
        warrenstreet_res, euston_res, kingscross_res, highburyislington_res,
        finsburypark_res, sevensisters_res, tottenhamhale_res,
        blackhorseroad_res):

    with stockwell_res.request() as req:
        yield req
        yield env.timeout(stockwell[0]) # RORIF (run in run out full speed)
        yield env.timeout(dwell(stockwell[1])) # dwell time
    with vauxhall_res.request() as req:
        yield req
        yield env.timeout(vauxhall[0]) # RORIF (run in run out full speed)
        yield env.timeout(dwell(vauxhall[1])) # dwell time
    with pimlico_res.request() as req:
        yield req
        yield env.timeout(pimlico[0]) # RORIF (run in run out full speed)
        yield env.timeout(dwell(pimlico[1])) # dwell time
    with victoria_res.request() as req:
        yield req
        yield env.timeout(victoria[0]) # RORIF (run in run out full speed)
        yield env.timeout(dwell(victoria[1])) # dwell time
    with greenpark_res.request() as req:
        yield req
        yield env.timeout(greenpark[0]) # RORIF (run in run out full speed)
        yield env.timeout(dwell(greenpark[1])) # dwell time
    with oxfordcircus_res.request() as req:
        yield req

```

```

    yield env.timeout(oxfordcircus[0]) # RORIF (run in run out full speed)
    yield env.timeout(dwell(oxfordcircus[1])) # dwell time
with warrenstreet_res.request() as req:
    yield req
    yield env.timeout(warrenstreet[0]) # RORIF (run in run out full speed)
    yield env.timeout(dwell(warrenstreet[1])) # dwell time
with euston_res.request() as req:
    yield req
    yield env.timeout(euston[0]) # RORIF (run in run out full speed)
    yield env.timeout(dwell(euston[1])) # dwell time
with kingscross_res.request() as req:
    yield req
    yield env.timeout(kingscross[0]) # RORIF (run in run out full speed)
    yield env.timeout(dwell(kingscross[1])) # dwell time
with highburyislington_res.request() as req:
    yield req
    yield env.timeout(highburyislington[0]) # RORIF (run in run out full speed)
    yield env.timeout(dwell(highburyislington[1])) # dwell time
with finsburypark_res.request() as req:
    yield req
    yield env.timeout(finsburypark[0]) # RORIF (run in run out full speed)
    yield env.timeout(dwell(finsburypark[1])) # dwell time
with sevensisters_res.request() as req:
    yield req
    yield env.timeout(sevensisters[0]) # RORIF (run in run out full speed)
    yield env.timeout(dwell(sevensisters[1])) # dwell time
with tottenhamhale_res.request() as req:
    yield req
    yield env.timeout(tottenhamhale[0]) # RORIF (run in run out full speed)
    yield env.timeout(dwell(tottenhamhale[1])) # dwell time
with blackhorseroad_res.request() as req:
    yield req
    yield env.timeout(blackhorseroad[0]) # RORIF (run in run out full speed)
    yield env.timeout(dwell(blackhorseroad[1])) # dwell time
times.append(env.now)

```

```

results = []
queue_times = {'stockwell':[], 'vauxhall':[], 'pimlico':[], 'victoria':[], 'greenpark':[], 'oxfordcircus':[],
'warrenstreet':[],
                'euston':[], 'kingscross':[], 'highburyislington':[], 'finsburypark':[], 'sevensisters':[],
'tottenhamhale':[], 'blackhorseroad':[]}

station_names = list(queue_times.keys())

population_multipliers = [1, 1.011556137, 1.022872241, 1.033953638, 1.044778943, 1.055347358,
1.065672062, 1.075722134, 1.085508796, 1.095007727, 1.10421749, 1.113165922, 1.1218826,
1.130409578, 1.138764508, 1.146965391, 1.155026722, 1.162985568, 1.170853517, 1.178601829,
1.186211175, 1.193656477, 1.200922381, 1.208038605, 1.214942843, 1.221420652, 1.227697595,
1.233771047, 1.239637984, 1.245286938, 1.250713593, 1.255913903, 1.260887549, 1.265636833,
1.270165695, 1.274481586]

year = 2015

for m in population_multipliers:
    population_multiplier = m

    env = simpy.Environment()

    # Resources
    stockwell_resource = simpy.Resource(env, capacity = 1)
    vauxhall_resource = simpy.Resource(env, capacity = 1)
    pimlico_resource = simpy.Resource(env, capacity = 1)
    victoria_resource = simpy.Resource(env, capacity = 1)
    greenpark_resource = simpy.Resource(env, capacity = 1)
    oxfordcircus_resource = simpy.Resource(env, capacity = 1)
    warrenstreet_resource = simpy.Resource(env, capacity = 1)
    euston_resource = simpy.Resource(env, capacity = 1)
    kingscross_resource = simpy.Resource(env, capacity = 1)
    highburyislington_resource = simpy.Resource(env, capacity = 1)
    finsburypark_resource = simpy.Resource(env, capacity = 1)

```



```

sevensisters_resource = simpy.Resource(env, capacity = 1)
tottenhamhale_resource = simpy.Resource(env, capacity = 1)
blackhorseroad_resource = simpy.Resource(env, capacity = 1)

times = []

env.process(source(env, stockwell_resource, vauxhall_resource, pimlico_resource,
    victoria_resource, greenpark_resource, oxfordcircus_resource,
    warrenstreet_resource, euston_resource, kingscross_resource, highburyislington_resource,
    finsburypark_resource, sevensisters_resource, tottenhamhale_resource,
    blackhorseroad_resource))

env.run(until = SIM_TIME)

headways = headway_analysis(times)

mean_headway = np.mean(headways)
achieved_tph = 3600.0 / mean_headway
target_headway = 3600.0 / target_tph
system_recovery = target_headway - mean_headway
results.append([year, achieved_tph])
year += 1

```

11.2 Appendix B - Distribution Analysis Results

Distribution	Mean R-Squared Score
alpha	0.73
anglit	0.55
arcsine	0.47
beta	0.69
betaprime	0.72
bradford	0.58
burr	0.77
cauchy	0.42

chi	0.65
chi2	0.72
cosine	0.57
dgamma	0.66
dweibull	0.66
erlang	0.69
expon	0.73
exponnorm	0.74
exponpow	0.66
exponweib	0.75
f	0.73
fatiguelife	0.71
fisk	0.77
foldcauchy	0.58
foldnorm	0.66
frechet_l	0.10
frechet_r	0.73
gamma	0.85
gausshyper	0.61
genexpon	0.72
genextreme	0.75
gengamma	0.71
genhalflogistic	0.69
genlogistic	0.70
gennorm	0.68
genpareto	0.73
gilbrat	0.79
gompertz	0.64
gumbel_l	0.47
gumbel_r	0.71

halfcauchy	0.58
halfgennorm	0.73
halflogistic	0.72
halfnorm	0.67
hypsecant	0.63
invgamma	0.73
invgauss	0.71
invweibull	0.75
johnsonsb	0.72
johnsonsu	0.81
kstwobign	0.67
laplace	0.64
levy	0.37
levy_l	0.01
loggamma	0.56
logistic	0.62
loglaplace	0.77
lognorm	0.73
lomax	0.75
maxwell	0.64
mielke	0.77
nakagami	0.67
ncf	0.68
nct	0.77
ncx2	0.71
norm	0.60
pareto	0.57
pearson3	0.70
powerlaw	0.58
powerlognorm	0.70

powernorm	0.81
rayleigh	0.65
rdist	0.44
recipinvgauss	0.71
reciprocal	0.26
rice	0.65
semicircular	0.54
t	0.69
triang	0.60
truncexpon	0.57
truncnorm	0.26
tukeylambda	0.58
uniform	0.51
vonmises	0.60
vonmises	0.12
wald	0.76

Table 19 - Distribution Analysis Results

11.3 Appendix C - Sample of Merged Dataset

Table 20 shows a 30 row sample of the merged NETMIS and RODS dataset.

TIMESTA MP	DWELL TIME	SUTOR DIRECTION LINE	BOARDERS ALIGHTERS	AND	SAF RATE	TRAIN DENSITY	STANDING CAPACITY	SEATING CAPACITY
11/30/15 5:23	11	KNT00	0		6.38861 21	1.07758620 7	116.6	268
11/30/15 5:22	19	CLW05	40		13.0053 3808	122.96	110.36	200
11/30/15 5:22	79	NGW06	104		12.6241 9929	84.4666666 7	145.92	234
11/30/15 5:22	21	SWC16	14		12.6241 9929	84.4666666 7	145.92	234
11/30/15 5:23	17	CWD04	0		7.92918 1495	114.294117 6	174	306

11/30/15				13.0053			
5:22	42	SWM05	17	3808	122.96	110.36	200
11/30/15				9.60747	115.207547		
5:22	26	ECM17	2	331	2	114	228
11/30/15				9.60747	115.207547		
5:22	15	OAK17	22	331	2	114	228
11/30/15				9.60747	115.207547		
5:23	22	HNE07	30	331	2	114	228
11/30/15				7.92918	114.294117		
5:23	20	ETE14	12	1495	6	174	306
11/30/15				13.0053			
5:23	22	TBE05	22	3808	122.96	110.36	200
11/30/15				13.0053			
5:23	13	BUR05	0	3808	122.96	110.36	200
11/30/15				9.60747	115.207547		
5:23	18	HNC17	2	331	2	114	228
11/30/15				14.2939	0.66216216		
5:23	21	GHL02	58	5018	2	155.02	272
11/30/15				9.60747	115.207547		
5:24	22	ALP07	33	331	2	114	228
11/30/15				13.0053			
5:23	26	TBY05	58	3808	122.96	110.36	200
11/30/15				6.84412	0.76190476		
5:24	19	FPK13	162	8114	2	153.2	252
11/30/15				7.92918	114.294117		
5:24	19	ICK14	6	1495	6	174	306
11/30/15				12.6241	84.4666666		
5:24	60	SJW16	7	9929	7	145.92	234
11/30/15				12.6241	84.4666666		
5:24	34	CNT06	178	9929	7	145.92	234
11/30/15				13.0053			
5:25	35	CLW05	40	3808	122.96	110.36	200
11/30/15				13.0053			
5:25	21	BAL05	21	3808	122.96	110.36	200
11/30/15				7.92918	114.294117		
5:25	21	FRD04	9	1495	6	174	306
11/30/15				13.0053			
5:54	28	CPS05	55	3808	122.96	110.36	200
11/30/15				6.38861	1.07758620		
5:54	27	HSD00	4	21	7	116.6	268

11/30/15				6.84412	0.76190476		
5:54	13	STK13	50	8114	2	153.2	252
11/30/15				6.84412	0.76190476		
5:54	65	SVS13	350	8114	2	153.2	252
11/30/15				13.0053			
5:54	25	WLO05	55	3808	122.96	110.36	200
11/30/15				12.6241	84.4666666		
5:54	29	WLO06	167	9929	7	145.92	234
11/30/15				13.0053			
5:54	17	MCR15	10	3808	122.96	110.36	200

Table 20 - Sample of merged NETMIS and RODs Datasets

11.4 Appendix D - Sample of Resampled Dataset

Table 21 shows a 30 row sample of the dataset used after resampling. The actual dataset used contains 33,529 rows of data.

TIMESTAMP	DWELL TIME	BOARDERS AND ALIGHTERS	SAF RATE	TRAIN DENSITY	STANDIN G CAPACITY	SEATING CAPACIT Y	BOARDERS TO ALIGHTERS DIFFERENCE	SUTOR DIRECTION LINE
03/11/15 05:00	26.77777778	15.88888889	9.60747331	115.2075472	114	228	12.11111111	HNW17
03/11/15 06:00	34.625	47.5	9.60747331	115.2075472	114	228	14	HNW17
03/11/15 07:00	32.9	47.4	9.60747331	115.2075472	114	228	-5.8	HNW17
03/11/15 08:00	26.75	46.91666667	9.60747331	115.2075472	114	228	-18.41666667	HNW17
03/11/15 09:00	26.71428571	35.28571429	9.60747331	115.2075472	114	228	-18.85714286	HNW17
03/11/15 10:00	30.2	32.8	9.60747331	115.2075472	114	228	-9.2	HNW17
03/11/15 11:00	26.9	32.2	9.60747331	115.2075472	114	228	-11.8	HNW17
03/11/15 12:00	29.45454545	48.72727273	9.60747331	115.2075472	114	228	-13.09090909	HNW17
03/11/15 13:00	29.45454545	77.63636364	9.60747331	115.2075472	114	228	-3.45454545	HNW17

03/11/15								
14:00	32.90909091	84.27272727	9.60747331	115.2075472	114	228	-18.45454545	HNW17
03/11/15								
15:00	35	84.66666667	9.60747331	115.2075472	114	228	-29.33333333	HNW17
03/11/15								
16:00	30.33333333	101.25	9.60747331	115.2075472	114	228	-70.91666667	HNW17
03/11/15								
17:00	31.72727273	154.6363636	9.60747331	115.2075472	114	228	-131.1818182	HNW17
03/11/15								
18:00	31.08333333	191.6666667	9.60747331	115.2075472	114	228	-171.3333333	HNW17
03/11/15								
19:00	28	139.1818182	9.60747331	115.2075472	114	228	-121.5454545	HNW17
03/11/15								
20:00	26.81818182	85.18181818	9.60747331	115.2075472	114	228	-72.09090909	HNW17
03/11/15								
21:00	29.36363636	71	9.60747331	115.2075472	114	228	-57.18181818	HNW17
03/11/15								
22:00	25.75	66.58333333	9.60747331	115.2075472	114	228	-46.75	HNW17
03/11/15								
23:00	25.91666667	49.75	9.60747331	115.2075472	114	228	-38.25	HNW17
04/11/15								
00:00	18.5	22.16666667	9.60747331	115.2075472	114	228	-19.16666667	HNW17
04/11/15								
01:00	15	9	9.60747331	115.2075472	114	228	-7	HNW17
04/11/15								
05:00	27.625	15.25	9.60747331	115.2075472	114	228	11.75	HNW17
04/11/15								
06:00	26.8	46.8	9.60747331	115.2075472	114	228	14.4	HNW17
04/11/15								
07:00	32.92307692	47.07692308	9.60747331	115.2075472	114	228	-7.692307692	HNW17
04/11/15								
08:00	30.18181818	47.54545455	9.60747331	115.2075472	114	228	-18.27272727	HNW17
04/11/15								
09:00	28.91666667	34.66666667	9.60747331	115.2075472	114	228	-18	HNW17
04/11/15								
10:00	25.76923077	32.46153846	9.60747331	115.2075472	114	228	-9.384615385	HNW17

04/11/15 11:00	31.33333333	32.16666667	9.60747331	115.2075472	114	228	-11.5	HNW17
04/11/15 12:00	32.54545455	46.36363636	9.60747331	115.2075472	114	228	-13.63636364	HNW17
04/11/15 13:00	28.7	77.5	9.60747331	115.2075472	114	228	-3.7	HNW17

Table 21 - Sample of resampled data