Mathematics of Last-Mile Transportation

Scott James Carnie-Bronca

November 23, 2022

Thesis submitted for the degree of
Master of Philosophy
in
Applied Mathematics
at The University of Adelaide
Faculty of Sciences, Engineering & Technology
School of Mathematical Sciences

THE UNIVERSITY
of ADELAIDE

Contents

List of Tables

List of Figures

Signed Statement

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

This work was supported with supercomputing resources provided by the Phoenix HPC service at the University of Adelaide.

Signed: Date: .....

Acknowledgements

I would firstly like to acknowledge and thank my supervisors for their support over the last few years. Their advice and feedback has been invaluable for directing my research, and their experience has made my studies interesting and enlightening throughout. I would especially like to thank Dr. Lewis Mitchell, who provided continuous and thorough advice for all aspects of my project, including the theoretical components, implementation and final presentations. To the other lecturers, academics and fellow students at the School of Mathematical Sciences, I thank you for providing inspiration and motivation through classes, lectures and conversations.

Finally, I would also like to thank my family for their continued support and encouragement, especially my parents Corrie and Robert and my sister Erin. Without your help this project could never have been possible.

Abstract

Last-mile transportation is a rapidly growing component of modern urban transportation, consisting of short trips between transportation hubs and destinations during the first and last sections of trips. A majority of studies in the field of last-mile transportation focus primarily on the well-established ride sharing systems such as bicycle rental. However, more recently electric scooters (or "e-scooters") have increased in popularity as a type of ride-sharing system allowing for rapid traversal of city streets and areas unsuited to bicycle traffic, particularly footpaths. These e-scooter systems differ significantly from traditional bike-sharing systems in that vehicles can be left in any location across the service area, rather than being restricted to specific dropoff locations. This makes modelling a system more challenging as it is no longer a fixed network, rather the locations where vehicles are deployed are non-stationary; they can change over time. This behaviour is more similar to free-floating car-sharing systems, however the use of e-scooters for short trips in urban areas and traversal of pedestrian areas causes different usage behaviours. The deployment of vehicles across the service area of an e-scooter ride-sharing system is paramount to the effective operation of the system to allow customers to take trips between their desired endpoints.

In this thesis, we investigate the patterns in e-scooter usage across the central business district of Adelaide, Australia, using data from two e-scooter ride-sharing system operators. We develop an algorithm for optimising the scooter deployment to maximise revenue generated in an e-scooter ride-sharing system. In Chapter 2 we introduce the data and perform an exploratory data analysis to show the high-level trends in behaviour. In Chapter 3 we develop a sophisticated model for estimating the spatiotemporal demand for vehicles in an e-scooter ride sharing system. These demand estimates then provide the foundations of the models used for simulating such an e-scooter system, shown in Chapter 4. Finally, these simulations then allow for testing the effectiveness of vehicle deployments across the service area, which we use to optimise the deployment of vehicles using simulated annealing in Chapter 5.

Chapter 1

Introduction

The *last-mile problem* is a well-known challenge in both logistics and transportation. In the field of logistics, the last-mile problem generally consists of the delivery of items from a transportation hub to a customer. This problem has been studied extensively, with a survey performed by Boysen et al. (2020) identifying key issues in last-mile delivery; an increasing demand due to population growth and urbanisation, sustainability, cost, and time pressure. In order to operate a successful delivery system, these aspects must be considered.

An analagous problem exists in transportation. Last-mile transportation is a rapidly growing component of modern urban transportation, consisting of short trips between transportation hubs and destinations during the first and last sections of trips. For example, when commuting to a workplace a passenger may take public transportation to a station or hub, located some distance from their workplace. By introducing an efficient and effective last-mile transportation system, a commuter can more quickly travel between a transportation hub and their destination. This may increase the usage of existing public transportation options by supplementing their service area, hence reducing the economic and environmental costs of travel (Shaheen & Chan (2016) and Lesh (2013)).

Last-mile transportation systems can take many forms. In this work, we focus on ride-sharing systems, which can generally be split into two distinct categories:

1. *Fixed-station ride sharing*: Ride-sharing systems where vehicles are located at several fixed locations, for example cars left in a carpark or bicycles left in a station on a street.

2. *Free-floating ride sharing*: Ride-sharing systems where vehicles can be located at any location in a fixed service area, for example bicycle or electric scooter systems without stations.

Fixed-station systems have been the subject of numerous studies in the last few decades. Pantuso (2022) propose a stochastic program for optimising both the relocation of vehicles and a location-based pricing system in a station-based car sharing system,

to maximise the expected profit for a ride-sharing operator. This stochastic program uses the earlier work by Bierlaire & Sharif Azadeh (2016), in which the authors propose an optimisation framework with integrated discrete choice modelling, based on demand for the choices, that ensures linearity of the problem, allowing for exact solutions to be obtained. A similar study by Wang (2019) proposes a general plan for last-mile transportation operators with an objective of minimising the waiting time of customers using a mixed integer program approach. Both a short-term solution, which depends on estimated demand for public transit, and a long-term solution depending on scheduled demand are implemented, which both provide improvements over existing strategies. Shu et al. (2013) propose an alternative, network flow model-based solution to the problem of allocating vehicles to stations in a bicycle sharing system. However, all of these approaches assume that there are a (relatively small) number of pre-determined stations, that vehicles are then moved between. The initial placement of these stations has also been studied, for example by Chen et al. (2019) in the case of electric bicycle sharing. Several authors also provide methods for deploying charging stations for electric vehicles including electric cars used for last-mile transportation, such as Andrenacci et al. (2016) and Gong et al. (2019).

Free-floating ride sharing systems require different analysis however, as there is no longer a finite number of locations where vehicles can be deployed. This category of last-mile transportation has been the target of many recent studies, ranging from the deployment of cars, bikes and e-scooters to the battery replacement and recharging of vehicles. Earlier research primarily concerns the deployment and usage of electric cars in free-floating car-sharing systems. Weikl & Bogenberger (2015) suggest a model for deployment and rebalancing consisting of two relocation types: macroscopic relocations between different, pre-defined zones of the service area, analagous to the station-based approach, in addition to intra-zone relocations of individual vehicles within the zones. A study by Schmöller et al. (2015) further investigates the demand and usage patterns of electric cars in the cities of Munich and Berlin in Germany. They find that the usage patterns tend to concentrate on a small number of hotspots of demand, causing an imbalance between the demand and supply of vehicles. More recently, Cui et al. (2022) investigate a novel model for charging vehicles in systems where, despite their free-floating nature, vehicles tend to form clusters in popular destinations. This characteristic can be used to efficiently charge a large number of vehicles in a single area.

Electric scooters, also known as e-scooters, are a recent form of free-floating dockless ride-sharing systems typically used in urban environments. This type of system allows for traversal of areas unsuited to other forms of last-mile transportation such as cars and bicycles, whilst being faster than walking. Several studies have investigated the societal effects of introducing an e-scooter ride-sharing system for last-mile transportation. An article by Moreau et al. (2020) discusses the environmental effects such as reduced pollution from transportation to inform a life-cycle assessment of an e-scooter system, applied to a system in the city of Brussels. They find that e-scooters are generally less

environmentally-efficient than alternative modes of transportation, however are likely to become more efficient as technology develops. Buehler et al. (2021) investigate the public perception of e-scooters, and commuters' likelihood of using such a system, before and after the introduction of an e-scooter ride sharing system. One finding was that approximately 30% of users utilised the e-scooter systems for the purpose of travelling to and from parking lots and public transportation, indicating that a large proportion of e-scooter commuters use such a system for last-mile transportation. A similar study conducted by Sanders et al. (2020) consisted of a study of university staff in Arizona found that e-scooters are considered as a more convenient method of travel compared to other methods such as walking, despite concerns of safety when using e-scooters in an urban environment. Baek et al. (2021) also found that e-scooters are a viable alternative to other last-mile transportation options by providing a flexible and fast mode of transportation.

Due to the recent introduction of electric scooters, few studies have investigated optimal strategies for operating an e-scooter ride sharing system. He & Shin (2020) propose a novel framework for predicting the flow distributions in e-scooter ride sharing systems, allowing for forecasting of e-scooter flows even after the service area of the system is modified. Ciociola et al. (2020) create a demand model using open data of e-scooter systems to estimate demand through both time and space, and compare the performance of different strategies when considering the number of vehicles to deploy and charging regimes. However, due to the reliance on heavily aggregated and anonymised data, the accuracy of the models are limited.

In this thesis, we obtain data from two ride-sharing companies that operated in the city of Adelaide, to allow for an in-depth investigation into the patterns of demand and usage of e-scooters. We then use these demand estimates in a stochastic optimisation framework to determine the deployment of e-scooters that maximises the expected revenue for an operator of an e-scooter system. While we solely study the behaviour of e-scooters, similar procedures could be used to estimate the demand for any mode of transportation that involves a free-floating ride-sharing system.

The structure of the remainder of the thesis is as follows. Chapter 2 discusses the e-scooter data that forms the basis of our analysis, including the high-level usage trends and the rate of battery decay of vehicles. In Chapter 3 we derive a model for estimating the demand for e-scooters. We assume that hires follow a Poisson process with a rate that varies depending on time and location, conditioned on vehicles being available in the area. Due to the presence of areas in the service area that contain very few recorded trips, we implement a Bayesian framework to more accurately reflect the demand in outlying areas. We then use these demand estimates to drive the hire process in the simulation models we construct in Chapter 4. These simulations allow us to investigate the dynamics of the e-scooter system, and assess the efficiency of particular operating strategies. Finally, we use the results from the simulations to estimate the expected revenue for a range of vehicle deployments across the service area, allowing us to optimise the deployment of vehicles

in Chapter 5. Due to the high dimensionality of the problem of vehicle deployment, we implement the simulated annealing algorithm. We find that the optimal deployments for a 6-hour time window generally follow the trends of the estimated demand, with vehicles deployed primarily along major roads and busy intersections along with public transportation hubs. However, the deployments are more even across the service area than the demand estimates, with areas with relatively low – but non-zero – demand still recieving vehicles.

All code used to perform the demand estimation, simulation and optimisation processes is available at `https://github.com/ScottCarnieBronca/escooter_mobility`.

Chapter 2

Data Analysis

In this chapter, we perform an exploratory data analysis of the mobility data, collected from e-scooter ride sharing operators, that is used in the remainder of this thesis.

2.1 Introduction

We obtained data from two e-scooter ride-sharing companies, RIDE (RIDE (2020)) and Beam (Beam (2022*a*)), which operated in Adelaide during 2018 to 2020 and 2020 to the time of writing in 2022 respectively. The RIDE data, spanning November 2018 to July 2019, consists of two datasets: Trip Data, containing information on every trip that occured in the RIDE system, and Snapshot Data, containing information about all scooters in the system at regular times. The Beam data contains Trip and Snapshot Data similar to that present in the RIDE data, however also contains a Status Data set which shows all *status* changes for all scooters in the system. This data is available for the entire operation of the system, starting in September 2020, however we only use the data up to October 2021.

In this chapter, we discuss and investigate all of these datasets. Section 2.2 highlights the service areas for e-scooter systems in the Adelaide Central Business District (CBD). Section 2.3 shows the process for the RIDE data sets. Section 2.4 shows the cleaning and analysis performed on the Beam data sets. In Section 2.5 we investigate the battery usage of scooters. Finally, in Section 2.6 we implement an algorithm to infer missing rows from the data sets that occur from unrecorded movement of scooters in the system.

2.2 E-scooter Service Areas in Adelaide

In the Adelaide CBD, e-scooter operation is restricted to a predefined area. This area has changed several times over the last few years as e-scooter systems have become more established. Here, we present the areas where e-scooters were allowed during the operation

of the RIDE system in 2019, and the area where e-scooters are currently allowed, for the Beam system.

Figure 2.1 shows the service area as of July 2019, which coincides with the data obtained from RIDE. This service area comprises of the Adelaide CBD, along with a portion of the surrounding parklands in the North and North-East of the CBD. There are some exclusions in the area; notably, Rundle Mall has an e-scooter ban, and Hindley Street in the West of the CBD has a ban on Friday and Saturday nights. Key transportation locations include the Adelaide Railway Station, located on North Terrace to the west of King William Road, which is included in the service area, and bus routes along King William Road, Grenfell Street and Currie Street. These locations are in the northern part of the CBD, where a majority of workplaces and traffic concentrate.

Figure 2.2 shows the e-scooter service area in the City of Adelaide council area in July 2022, which has expanded considerably from the 2019 area. As well as the CBD, some Southern parklands are included, along with North Adelaide and its surrounding parklands. While North Adelaide is less urbanised than the CBD, its inclusion in the operating areas allows for trips to occur over a longer distance, as commuters may travel between North Adelaide and the Adelaide CBD. We see the same exclusions as in 2019.

E-scooter ride-sharing operators typically allow their vehicles to travel to anywhere within the areas created by the City of Adelaide and other local governments. In the case of the Beam system, several nearby councils such as the City of Norwood, Payneham and St Peters in the East (City of Norwood, Payneham & St Peters (2021)) and the City of West Torrens in the West (City of West Torrens (2022)) allow for limited operation of e-scooters. However, we focus on the e-scooter usage across the Adelaide CBD, North Adelaide and the surrounding parklands as more data is available, and we expect the usage to be higher in the more urbanised area.

2.3 RIDE Data Analysis

2.3.1 RIDE Trip Data

This dataset contains 32,098 trips recorded between the 2nd of November 2018 and the 10th of July 2019. The important variables in the dataset are listed in Table 2.1. In this section, we investigate patterns in these variables and describe the cleaning process undertaken in order to remove missing and incorrect values.

For 6 trips in the dataset, at least one of these variables has a missing value. Due to the very low number of missing values, we remove these entries from the data.

The first variables in the RIDE trip data we investigate are the pickup and dropoff locations. These are stored as latitude and longitude for each end of the trip, as recorded by the GPS units built into the scooters. We expect these locations to occur entirely in and around Adelaide, primarily in the service area described in Section 2.2. However,

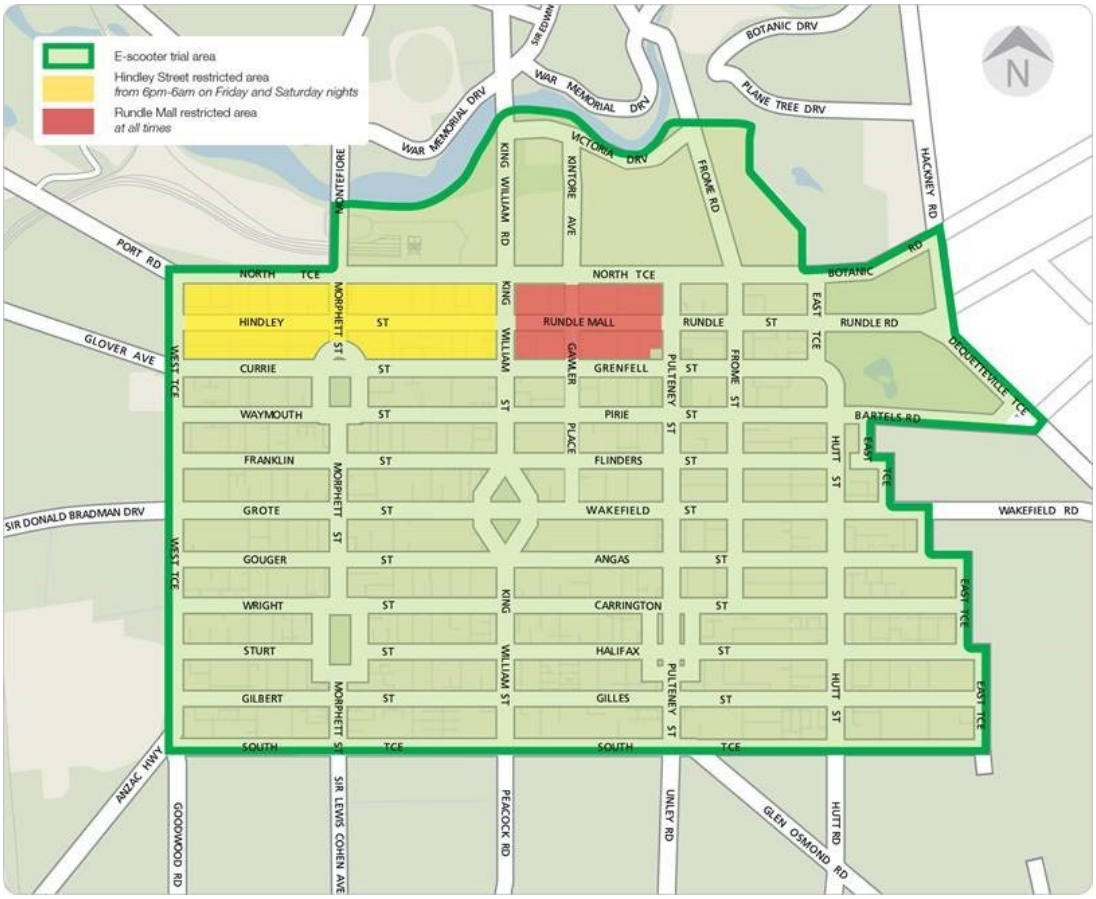


Figure 2.1: Map of the service area of e-scooter systems in Adelaide, July 2019. This area applied to the RIDE system. The southern edge of the service area, along South Terrace, is 2.7km in length, and the western edge of the service area, along West Terrace, is 1.6km in length. (City of Adelaide (2019))

Figure 2.2: Map of the service area of e-scooter systems within the City of Adelaide, July 2022. This area applies to the operation of the Beam system. The length of the southern edge of the service area, along Greenhill Road, is 2.7km, while the distance between the southernmost and northernmost edges, Greenhill Road and Fitzroy Terrace respectively, is 4.7km. (City of Adelaide (2022))

Variable Name	Type	Notes
pickupTs	character	Time of the start of the trip
dropoffTs	character	Time of the end of the trip
vehicle	character	ID of vehicle
pickupGeo0	numeric	Longitude of scooter pickup location
pickupGeo1	numeric	Latitude of scooter pickup location
dropoffGeo0	numeric	Longitude of scooter dropoff location
dropoffGeo1	numeric	Latitude of scooter dropoff location
totalsduration	numeric	Duration of the trip, in seconds
totalsdistance	numeric	Distance of the trip recorded by the scooter, in metres
totalsamount	numeric	Cost of the trip, in AUD cents

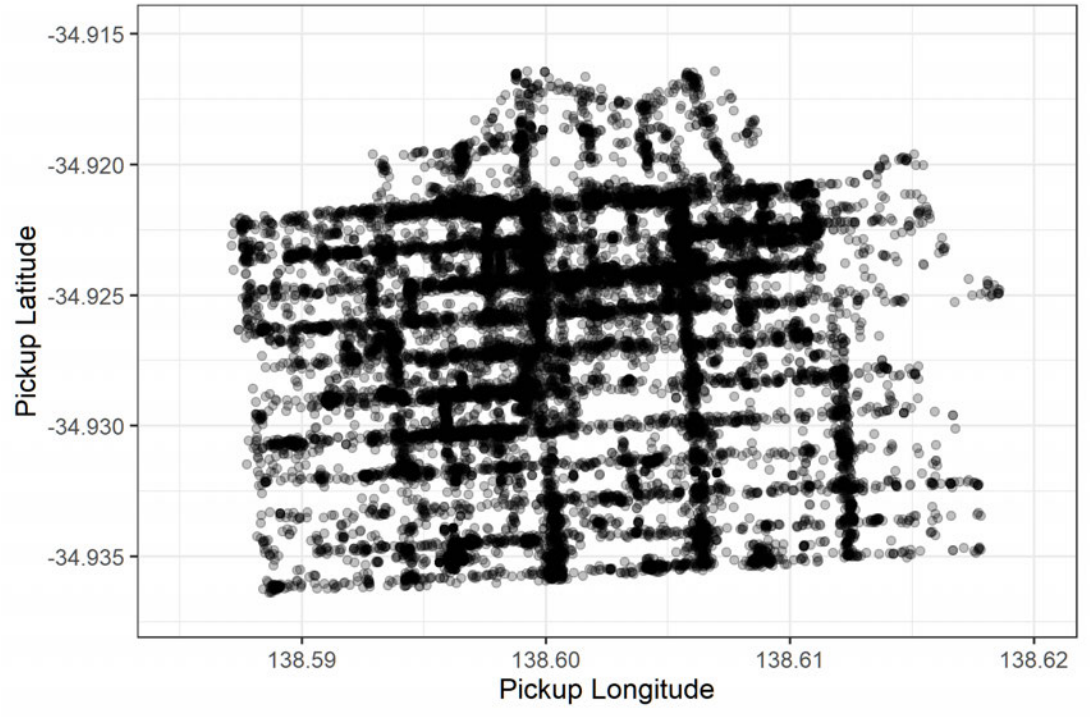
Table 2.1: List of key variables included in the Trip Data from the RIDE system.

7742 trips (24.1% of the trips) contain an endpoint that is outside the service area. As we are interested in demand in the service area, we remove these trips from the data set. Figure 2.3 shows all remaining pickup and dropoff locations after removing trips that occur outside of the service area.

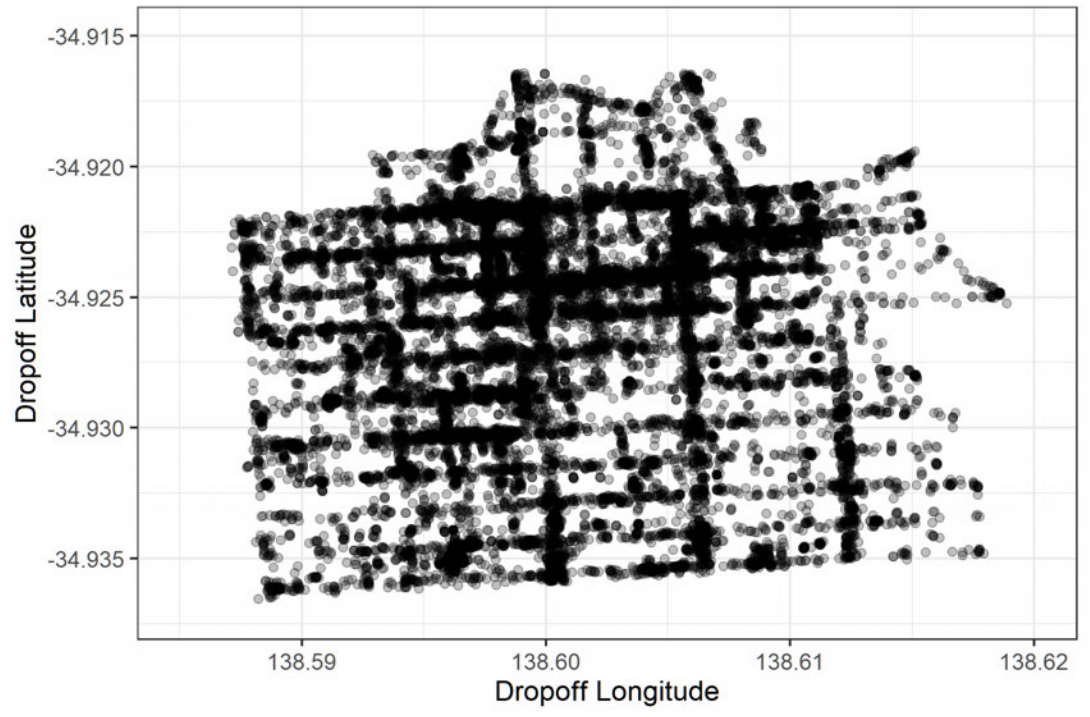
The pickup and dropoff locations follow the streets in the Adelaide CBD fairly well, with a relatively small number of trips taking place off of the streets. This is likely due to inaccuracy in the scooters' GPS units. There also appears to be a significant skew towards the northern part of the city, shown at the top of the images, in both hires and dropoffs. This coincides with the location of key transportation in the Adelaide CBD, including the Adelaide Railway Station and a large number of bus routes.

Another feature of the location data is that there is a large number of trips that start and end in one location in the south-west area of the city. This is the location of the RIDE warehouse, and so trips here are likely test trips, or RIDE moving scooters across the city for balancing – where the operator moves vehicles to areas of high demand – or maintenance purposes. This is further supported by the demand in surrounding areas being much lower. Due to this, we remove the 2180 trips that start or end in a 50×50m area around the RIDE warehouse.

The next variables we investigate are the pickup and dropoff times. These are stored as character vectors in local time. We convert these to `R` date-time objects using the `lubridate` package, which handles daylight savings time automatically (Grolemund & Wickham (2011)). To analyse these variables, we plot the cumulative number of trips taken over time in Figure 2.4. After the filtering performed above, the data now spans February to July 2019. However, only one trip occurs prior to the 15th of April, on the 4th of February. Since this is likely to be a test trip, we remove it from the data set. The hire rate appears to be approximately constant during and after May, after a short ramping period from mid-April to mid-May. This indicates that, when estimating demand, we do



(a) E-scooter pickup locations recorded in the RIDE data.



(b) E-scooter dropoff locations recorded in the RIDE data.

Figure 2.3: Pickup and dropoff locations recorded in the RIDE trip data set. These locations are filtered to those that occur in the service area. The locations primarily occur along streets, especially towards the north of the city (top of the plots).

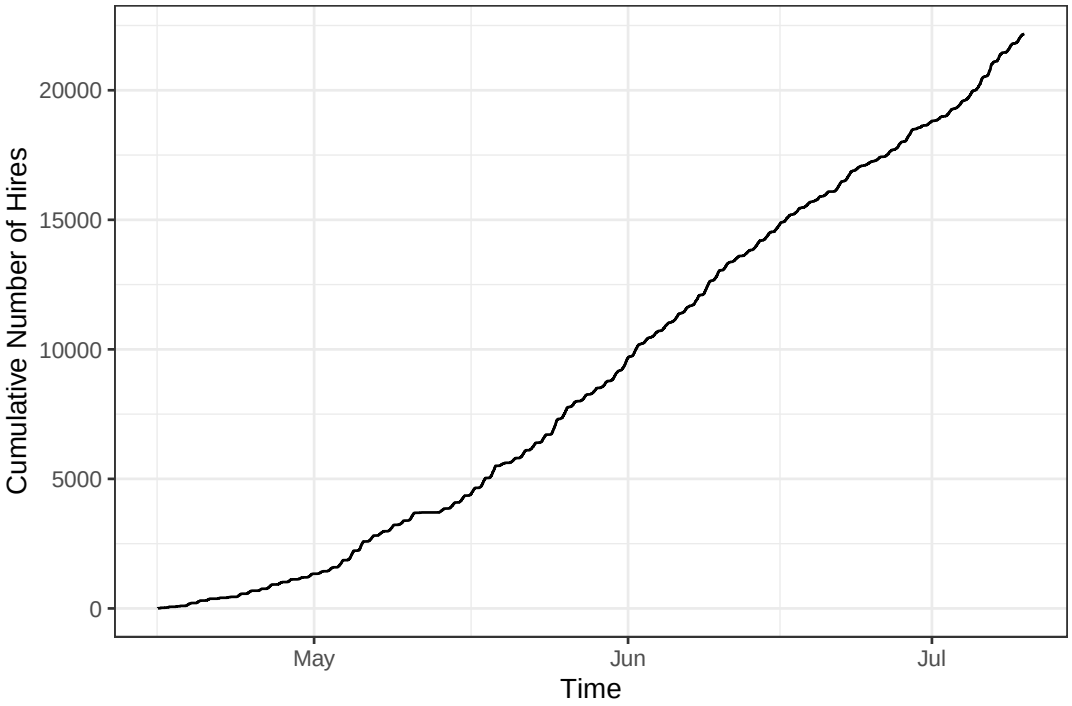


Figure 2.4: Cumulative number of hires over time in the RIDE trip data, which covers February to July 2019. Only one trip occurs prior to the 15th of April, 2019, and is likely a test trip.

not need to account for long-term demand variation in the data.

We further analyse the hire times by plotting the frequency of hires through time, split by time of day and day of week. In Figure 2.5 we plot the total number of hires for each half-hour segment of the week. As shown, for all days of the week the demand appears to be lowest between midnight and 7am, before increasing to a peak at around 3pm, then decreasing again. We do see some differences between the day of the week however; in Friday and Saturday afternoons and evenings the number of hires is much higher than the average, and Saturday and Sunday have higher than average numbers of hires during the early morning. This indicates that usage patterns for e-scooters change based on both the time of day and the day of the week.

The final component of the RIDE Trip Data we analyse is the trip metrics in the data; the trip duration and distance recorded by the scooter for each trip. While we can accurately calculate the duration of a trip from the start and end times, the distance cannot be calculated from only the endpoints. Figure 2.6 shows the distributions of these two values over all trips. As shown, the trip duration is positively skewed with a peak of around 450 seconds, or 7.5 minutes. The trip durations in the data vary from 2 to 1.249×10^7 seconds – about 144 days – however only 1.2% of the trips take longer than

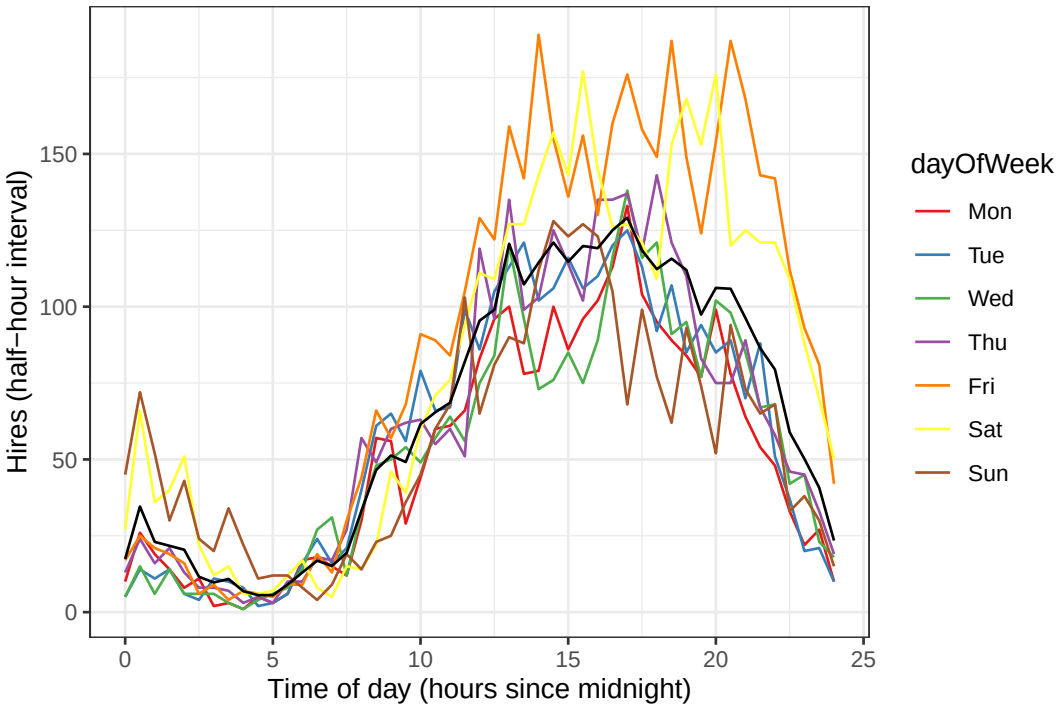


Figure 2.5: Number of hires that occur in the RIDE trip data for each time of day and day of the week. The black line is the average number of hires for that time of day, over all days of the week. Note that some days of the week may have higher numbers of trips due to the data not representing an integer number of weeks.

two hours (7200 seconds). Due to the very low frequency of the extremely long trips, it is likely that they are errors.

The trip distance distribution has several peculiar features. Firstly, the distribution is bimodal with a peak at 0m and 750m. The peak at zero likely corresponds to either test trips, or instances where an e-scooter customer hires a vehicle then ends the trip without actually moving. The peak at 750m then corresponds to the average distance travelled using e-scooters in the RIDE system. However, many of the trip distances are negative. As we expect distances measured by e-scooters to be non-negative, this indicates that the trip distance measurements from RIDE scooters are unreliable. We were unable to determine the cause of these negative measurements, which comprise 4.6% of the trips in the data. As we do not use these measurements in further analysis, we do not remove these trips.

2.3.2 RIDE Snapshot Data

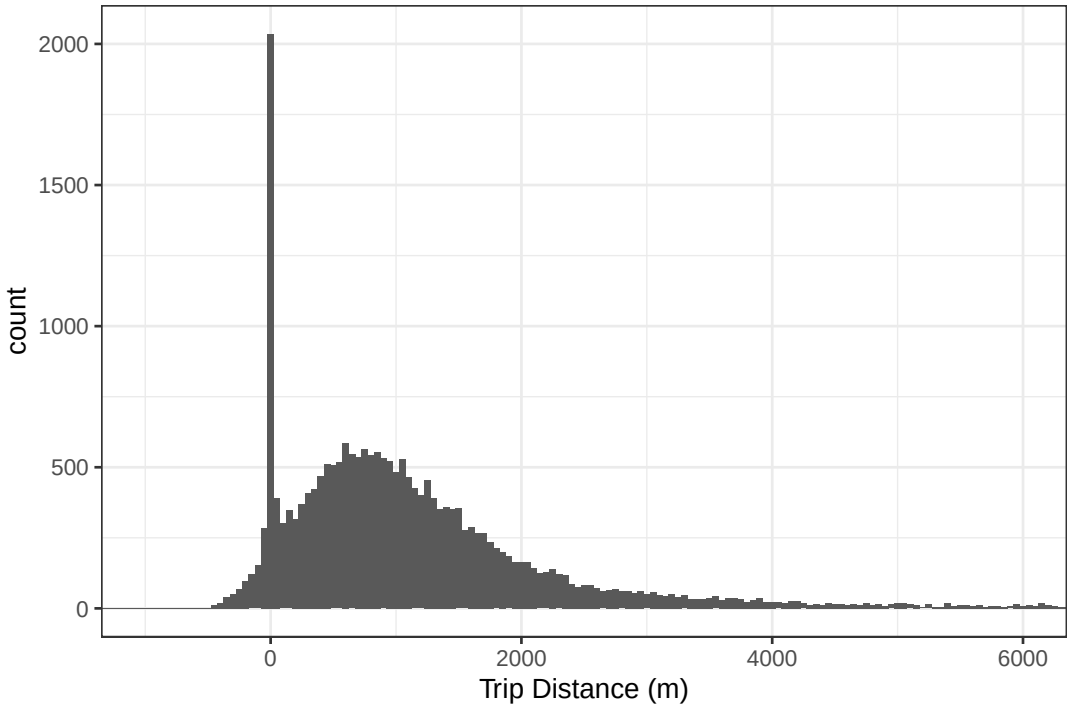
Along with the trip dataset, RIDE provided a snapshot dataset containing half-hourly reports of the current status of all scooters in the system. This contains 369,329 such reports spanning the 15th of June 2019 to the 10th of July 2019. Table 2.2 describes the variables in this dataset that are important for further analysis. In this section we discuss how we processed the data, along with investigating some of the trends in the data.

Variable Name	Type	Notes
DataDate	character	Time of the snapshot
id	character	ID of vehicle
geo0	numeric	Longitude of current scooter location
geo1	numeric	Latitude of current scooter location
battery	numeric	Current battery level of the scooter, from 0-100.

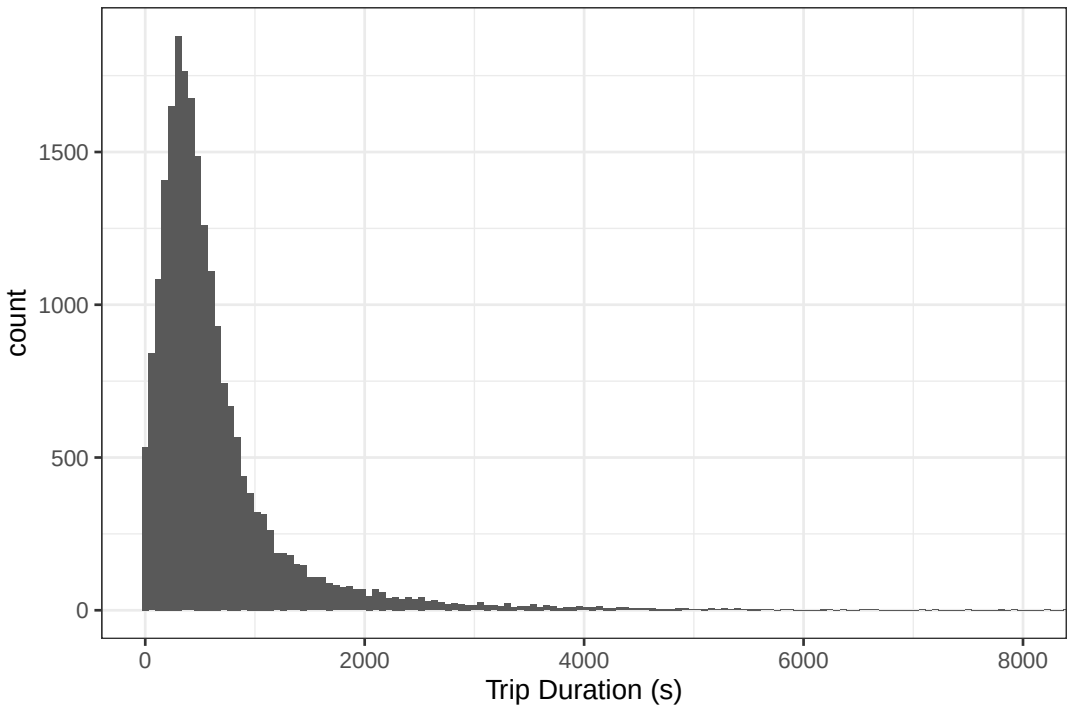
Table 2.2: List of key variables included in the snapshot data from the RIDE system. Each variable is collected for all scooters at regular intervals.

We process this data similarly to the Trip Data. Firstly, we filter the data to contain only points inside the service area. In this step, we remove 108,013 snapshots. While this is approximately 30% of the data, we remove these entries entirely as these snapshots occur outside the service area, and so are not relevant to the demand estimation inside the service area. We also remove points that occur in the area around the RIDE office in the south-west of the service area.

We next consider the number of unique vehicles present in the data over time. We expect the number of scooters to remain roughly constant over time. However, as shown in Figure 2.7, the number of scooters fluctuates significantly over time. In general, the number of scooters appears to decrease over time with a sharp increase in the number of



(a) Histogram of trip distances reported by vehicles in the RIDE system, with a bin width of 50m. The plot here is truncated; the trip distance ranges from -22km to 178km.



(b) Histogram of trip durations reported by vehicles in the RIDE system, with a bin width of 60 seconds. The plot here is truncated; the trip distance ranges from 0 to 1.249×10^7 seconds (approximately 144 days).

Figure 2.6: Distributions of distance and duration metrics as reported by scooters during trips taken in the RIDE system.

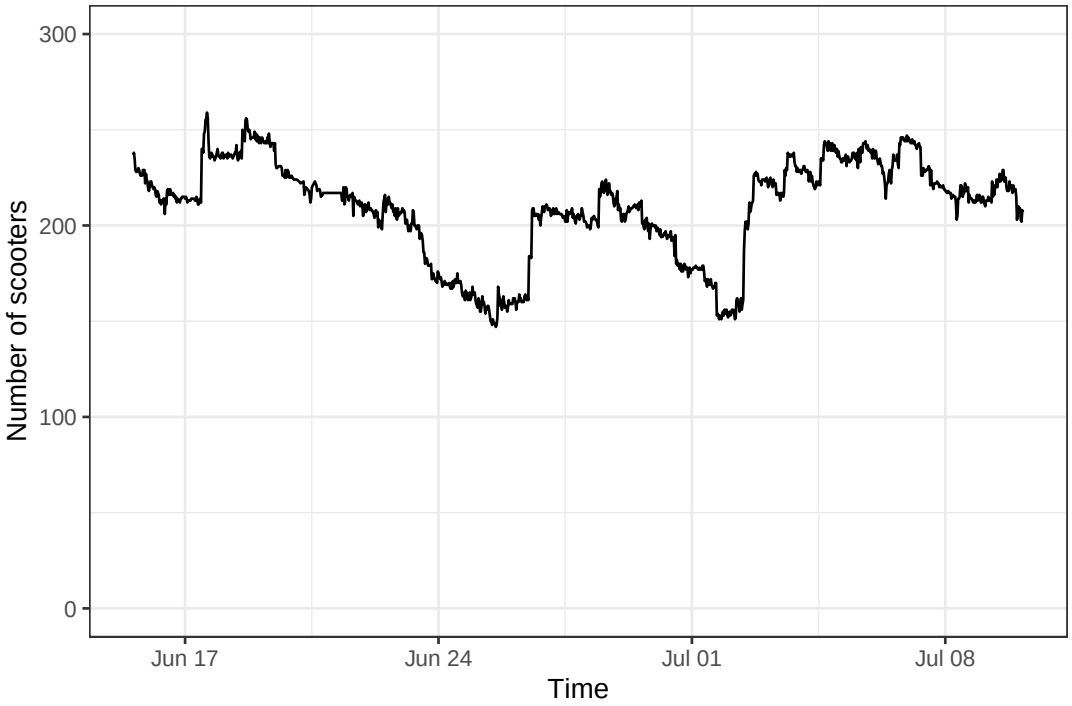


Figure 2.7: Number of unique scooters present within the service area in the RIDE Snapshot Data during 2019.

scooters on three occasions. This behaviour may be due to scooters becoming damaged or running out of battery over time, then being repaired or recharged by RIDE before being redeployed into service. There are 150-250 vehicles in operation over the time window of the data, with a peak of 259 scooters on the 17th of June 2019.

The RIDE Snapshot Data also contains the current battery level of each vehicle, recorded as an integer between 0 and 100. Figure 2.8 shows how the average battery level across all scooters changes over time. The mean battery level appears to stay around 60% for all times, with fluctuations between 45% and 65%. There are several large jumps in the battery levels, with the largest occuring on the 8th of July 2018. This does not coincide with a sharp change in other values such as the number of scooters, so it is likely due to a large number of battery recharges occuring in a short period of time. We further analyse the battery level data by modelling the battery usage in Section 2.5.

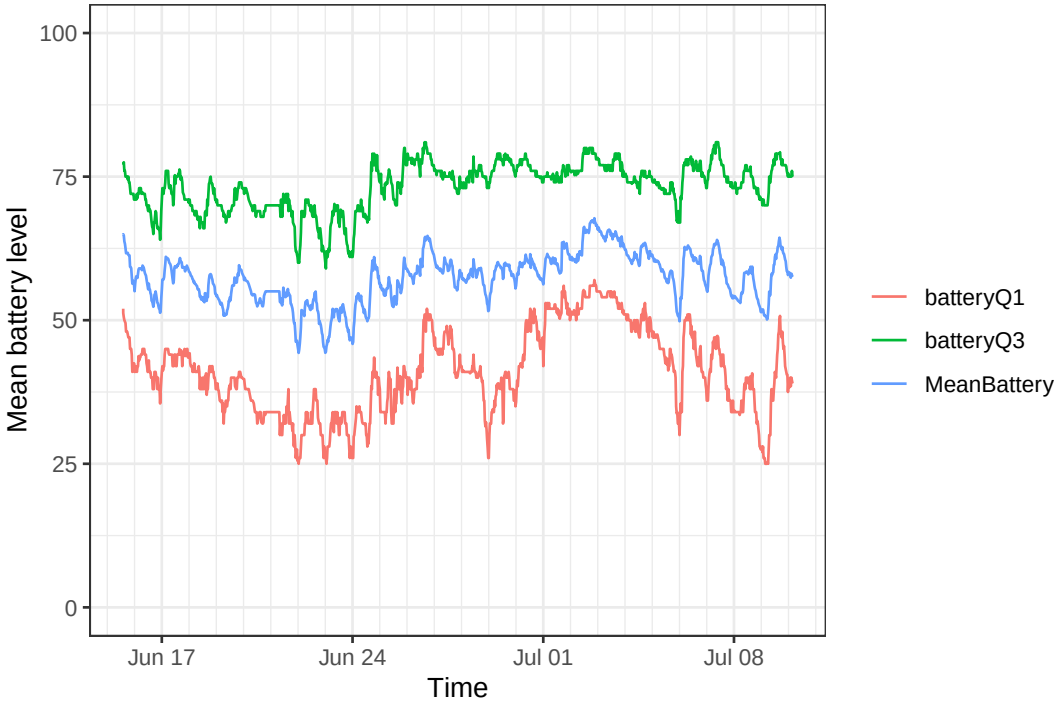


Figure 2.8: Battery level of scooters present in the RIDE Snapshot Data over time. The mean battery level is shown in the blue line, whereas the green and red lines show the 0.75 and 0.25 quantiles at each time respectively.

2.4 Beam Data Analysis

2.4.1 Beam Trip Data

Similarly to the RIDE data, the Beam Trip Data contains information on all trips that occur in the e-scooter system. While the data was accessible from a web API, the subset of the data we used spans from the start of operation on the 28th of September 2020 to the 1st of November 2021. During this time, 441,443 trips were taken.

Figure 2.9 shows the number of hires taken each day in this period. There appears to be a strong cyclic component to the number of hires, with higher hire rates occuring on weekends. There are also two periods of time where the hires are much lower than expected. The first of these takes place between the 19th and the 22nd of November 2020, where no hires are recorded for the entire four-day interval. This coincides with a COVID-19-related lockdown in Adelaide at the time (Poulson et al. (2020)). Similarly, the drop in hires between the 21st and 27th of July 2021 occured due to another lockdown in Adelaide for the 7-day period (Keane & Opie (2021)). We do not modify the data during these times, however these lockdowns and other related restrictions may have an affect on the behaviour of commuters using e-scooters. We also see a generally increasing trend in the number of hires over time. February and March 2021 recorded the highest numbers of hires, which coincides with the Adelaide Fringe Festival which attracted a large number of commuters to Adelaide (Sutton (2021)).

In this data, all variables used in our analysis are present for all trips. However, a small number of trips appear unrealistic. We consider the trip distance and trip duration metrics. As we expect the e-scooter system to be used for last-mile transportation or other relatively short trips, the trip distances and durations should also be short. The trips primarily follow this with trip distances of less than five kilometres and durations of less than two hours, however 667 trips take longer than two hours, 181 trips travel more than five kilometres, and 8 of these trips take longer than two hours and travel more than five kilometres. We assume that in these instances, the data recorded for the trip is unrepresentative of a typical trip; for example, a scooter may have been hired and then left idle for a long period of time causing the long trip duration. In the simulation and optimisation models developed in Chapters 4 and 5, we take the average trip duration and time to inform scooter movement, so these extreme trips would heavily impact the results of these models. Hence, we remove the 859 trips that are either longer than two hours or five kilometres. These trips comprise 0.2% of the original trip data.

The Beam e-scooter ride-sharing system has changed service areas several times throughout its operation, adding areas such as the beach to the west of the city and some inner suburbs to the east of the city. Due to this, we do not restrict the data to the service area. As discussed later in Section 3.5.4, prior to estimating the demand for e-scooters we bin the data into a grid across the CBD. This has the effect of restricting the data to the chosen grid.

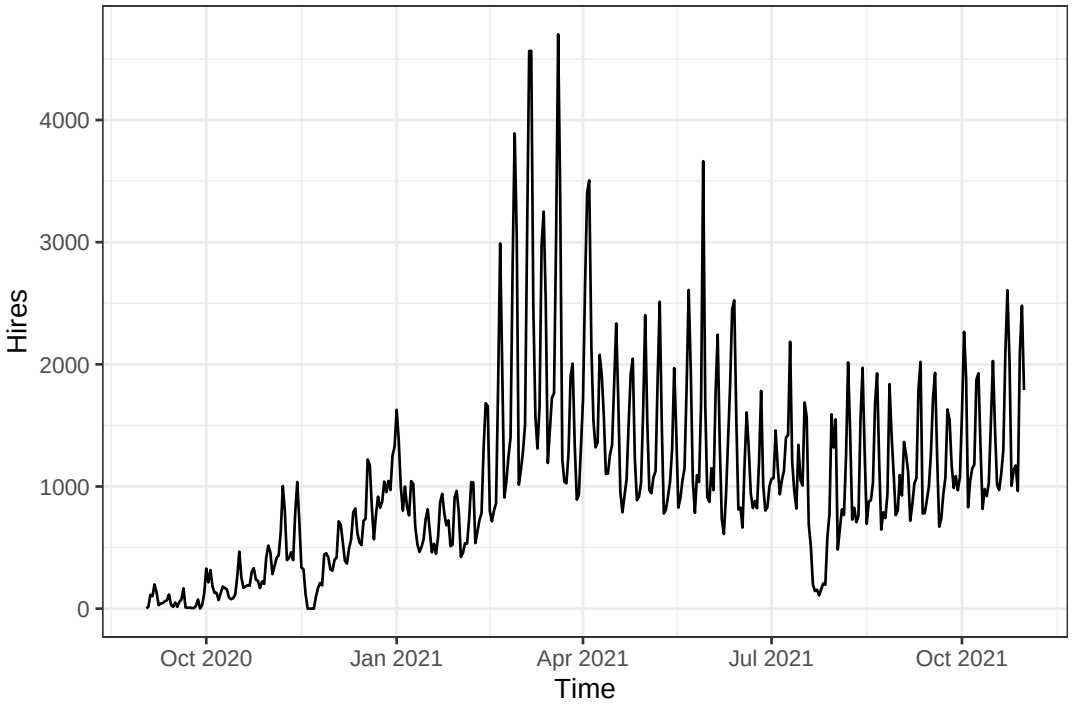


Figure 2.9: Number of hires taken each day in the Beam trip data set, spanning from September 2020 to November 2021.

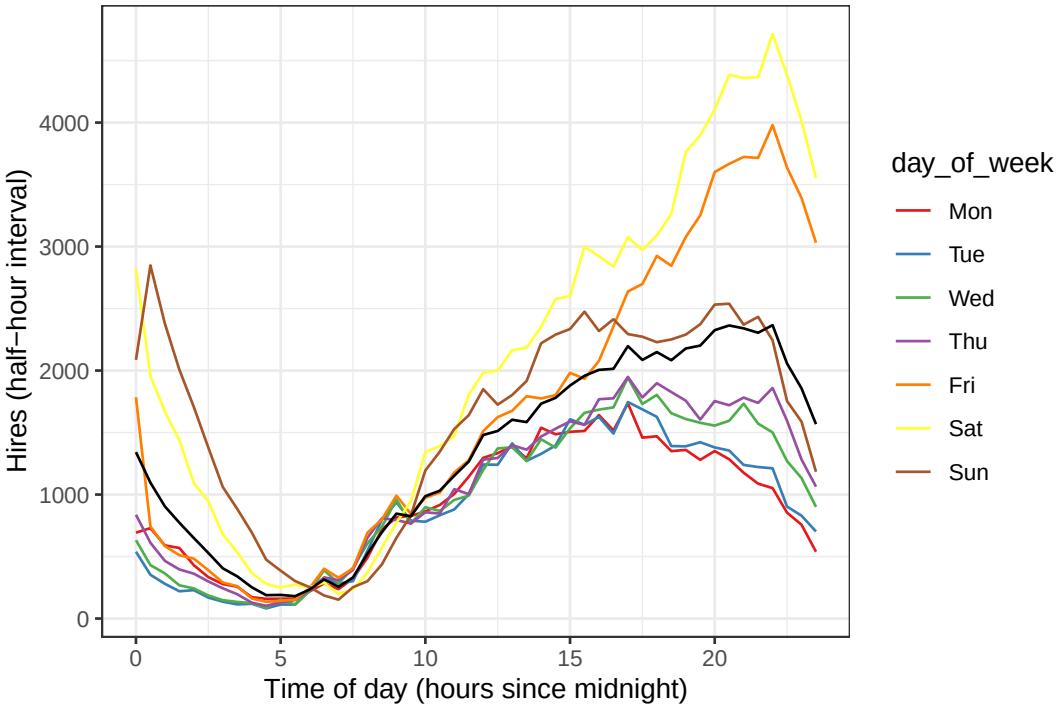


Figure 2.10: Number of hires recorded in the Beam trip data that occur in half-hour intervals for each day of the week. The black line shows the number of hires averaged across all days of the week.

We now consider the times that trips occur in the Beam system. In the Beam data, all times are stored in unix epoch time (time in milliseconds since January 1st 1970), which can be converted into R `datetime` objects. Figure 2.10 shows the total number of hires that occur in each half-hour for each day of the week, over the time window of the data. Several trends are apparent in this plot.

First, we see that the number of hires is cyclic within each day, with the lowest usage seen between 5am and 6am on all days. The peak demand for Sunday through to Thursday occurs between 2pm and 8pm, whereas on Friday and Saturday the peak occurs between 4pm and midnight. This shifted peak window indicates that the e-scooter system experiences different usage patterns on these days.

We also see that the demand on Friday afternoons through to Sunday morning appears to be much higher than the demand at other times. Saturday evenings have the highest number of trips, with 4,666 trips taken between 9:45pm and 10:15pm, almost double the number of trips averaged across all days for the same time, 2,366. However, all days appear to exhibit similar behaviour between 5am and 9am.

Despite the significantly increased number of total trips that occur in the Beam e-scooter system, the variance in the number of hires through time appears to be lower

relative to the frequency of hires compared to the variability in the RIDE system shown in Figure 2.5.

The next components of the Beam Trip Data we investigate are the trip metrics, distance and duration. These are recorded in the Trip Data in metres and seconds respectively for each trip. Like in the RIDE Trip Data, these allow us to investigate general usage patterns across all users of the system. Unlike in the RIDE data, however, we do not see any negative values in either metric. The distributions of these metrics for all trips is shown in Figure 2.11. The distribution of trip duration is positively skewed with a mean of 663 seconds, just over 11 minutes. Of the trips less than two hours in length, only 0.7% take more than one hour. This indicates that restricting the trips to only those that take less than two hours retains the majority of useful trips. The trip distance is also positively skewed, however is bimodal with a peak at both 0m and 600m. This indicates that a large proportion of trips only take place over a short distance. Around 12.7% of the trips travel less than 100m.

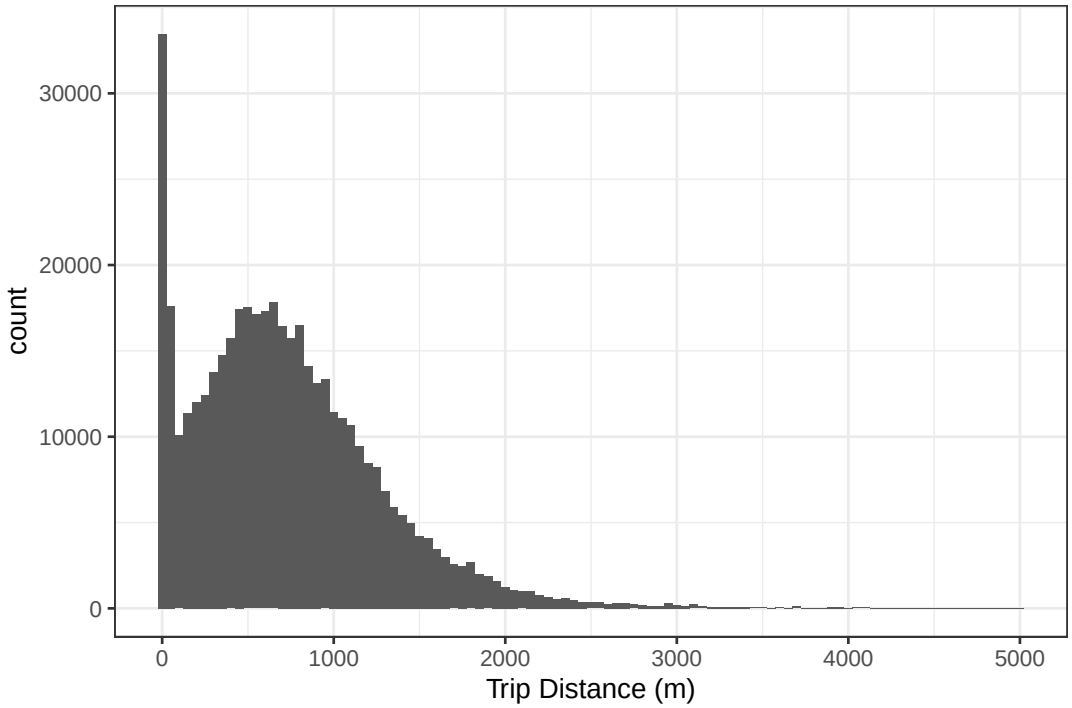
2.4.2 Beam Status Data

The final dataset we investigate is the Beam Status Data. This contains a list of all status changes that occur for any vehicles in the Beam system, including hires, dropoffs, maintenance pickups and maintenance dropoffs. This allows us to track vehicle usage and maintenance without comprehensive snapshot data. There are 1,048,145 status updates for 1415 unique scooters between the 20th of August 2020 and the 18th of November 2021. Table 2.3 shows the variables we consider important to the analysis performed in the remainder of this thesis.

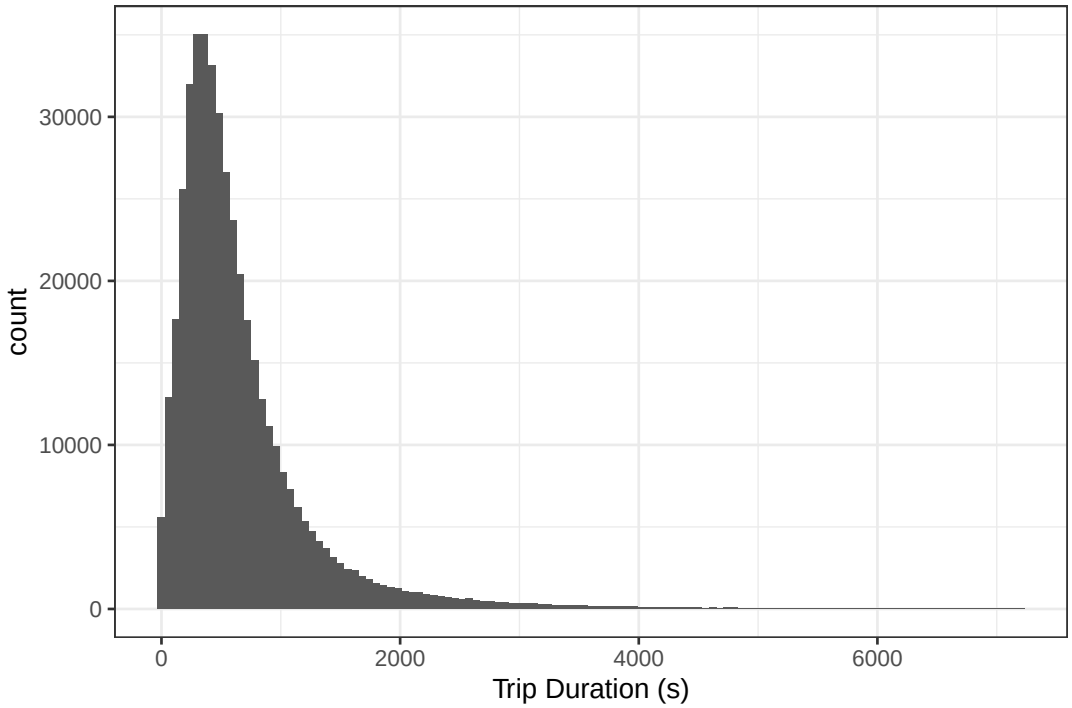
Variable Name	Type	Notes
vehicle_id	character	ID of the vehicle
event_type_reason	character	Type of status update
location_timestamp	numeric	Time of status update in milliseconds
event_time	character	Human-readable time, in the local time zone
battery	numeric	Current battery level of the vehicle, from 0-1
lon	numeric	Current longitude of the scooter
lat	numeric	Current latitude of the scooter

Table 2.3: List of key variables included in the Status Data from the Beam system.

Like the Beam Trip Data, this dataset contains few errors. The primary processing we perform is removing duplicated rows. There are several instances of entries in the dataset where all variables are identical except for the `event_time` variable, which contains the human-readable time in the local time zone, which differs by exactly 9.5 or 10.5 hours. As



(a) Distribution of trip distances reported by vehicles in the Beam system.



(b) Distribution of trip durations reported by vehicles in the Beam system.

Figure 2.11: Distributions of distance and duration metrics as reported by scooters during trips taken in the Beam system.

these times are the offset from UTC of the local time zone, ACST or ACDT respectively, these are likely errors. We remove the later of these duplicated rows.

The battery level of a scooter is reported at every status change, allowing us to investigate the battery usage of Beam e-scooters. In Figure 2.12 we show the average battery level of all scooters in the Beam system over time. Due to the non-regular times of status updates, we aggregate the data into three-hour bins. This causes the battery level to fluctuate as the number of scooters used for each battery calculation changes. We see that the battery level generally stays at around 0.65, or 65% of the capacity of the battery. This matches the result from the RIDE data. The average battery level spikes negatively in late November 2020, coinciding with the COVID-19 lockdown. During this time, the average battery level reaches 0.13, however the number of status updates is greatly reduced, with less than three updates used for all estimates of battery level in this time period.

We also consider the types of status updates and their frequency over time. Figure 2.13 shows the frequency of each type of status update. The majority of events are `user_pick_up` and `user_drop_off` events, which correspond to the trips discussed in Section 2.4.1. The number of trips recorded in the Status dataset match the number of trips in the Trip dataset. We also see that there is only a small difference between the number of pickups and dropoffs within each day, which indicates that the trip endpoints are recorded correctly. The other two types of events, `maintenance_pick_up` and `maintenance_drop_off`, are much less frequent with a peak of 501 maintenance pickups and 493 maintenance dropoffs on the 20th of March 2021. The overall frequency follows the number of hires, with the number of maintenance hires and dropoffs at approximately 10% of the number of user trips each day.

The difference in event frequency is also shown in the total number of each event type; there are 468,807 user dropoffs and 468,626 hires compared to 50,739 maintenance dropoffs and 58,180 maintenance pickups. The difference in the number of user hires and dropoffs is likely due to the fixed time window of the data; any trips in progress at the start or end of the data set would only have one trip endpoint recorded. However, 12.7% of the maintenance pickups do not have a corresponding dropoff, which is much higher than the 0.04% difference in the number of hires and dropoffs.

Finally, we investigate the frequency of pairs of event types, and the length of time between each pair of events. Table 2.4 lists all of the pairs present in the data. There are 14 pairs present, the most common of which is a user hire then dropoff, which corresponds to a regular trip. The average time between the two events in the pair is 671s, which matches the average trip length of 663s in Section 2.4.1. This is to be expected, as both measurements are of the average trip length of all trips in the Beam system. The other most frequent pairs include a maintenance pickup followed by a dropoff with 50,682 occurences, and all four combinations of either a user or maintenance dropoff then a user or maintenance pickup. However, there are several pairs of events present that are not

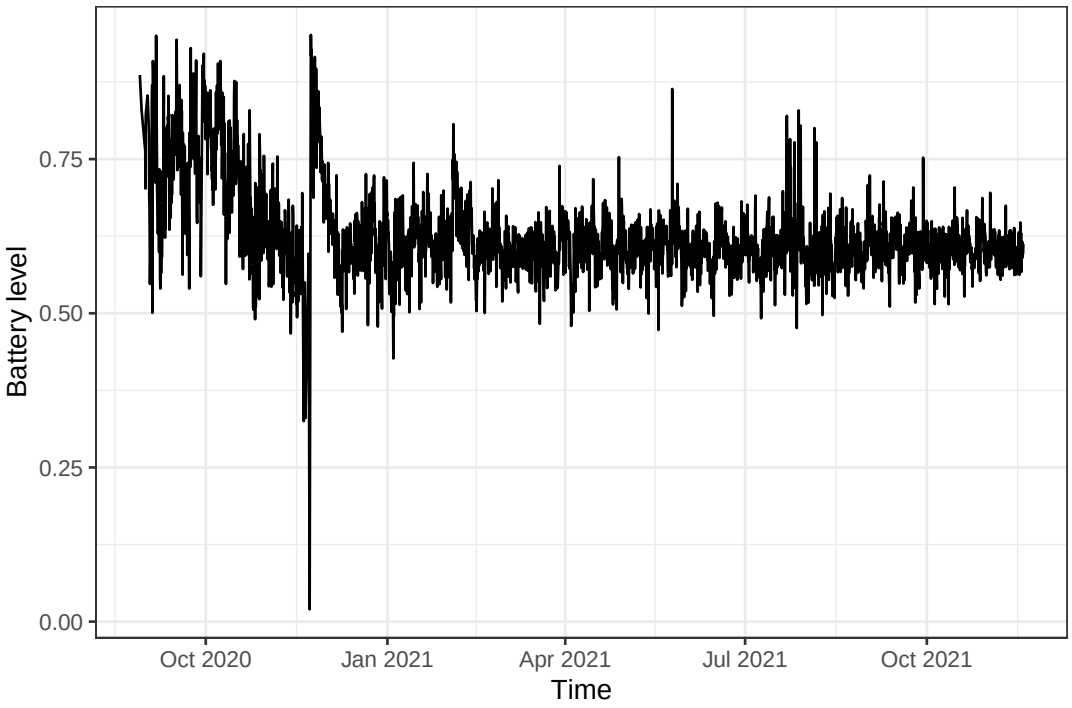


Figure 2.12: Mean battery level in the Beam Status Data, with 3-hour aggregating. The large amount of variability is due to the variance of the number of scooters present in each 3-hour window, as status updates are not performed at regular times for all vehicles. The large drop in battery level in late 2020 coincides with the COVID-19 lockdown at the time, where the number of status updates was significantly impacted. The lowest mean battery level outside of this time occurs on the 4th of January 2021, with a battery level of 0.43.

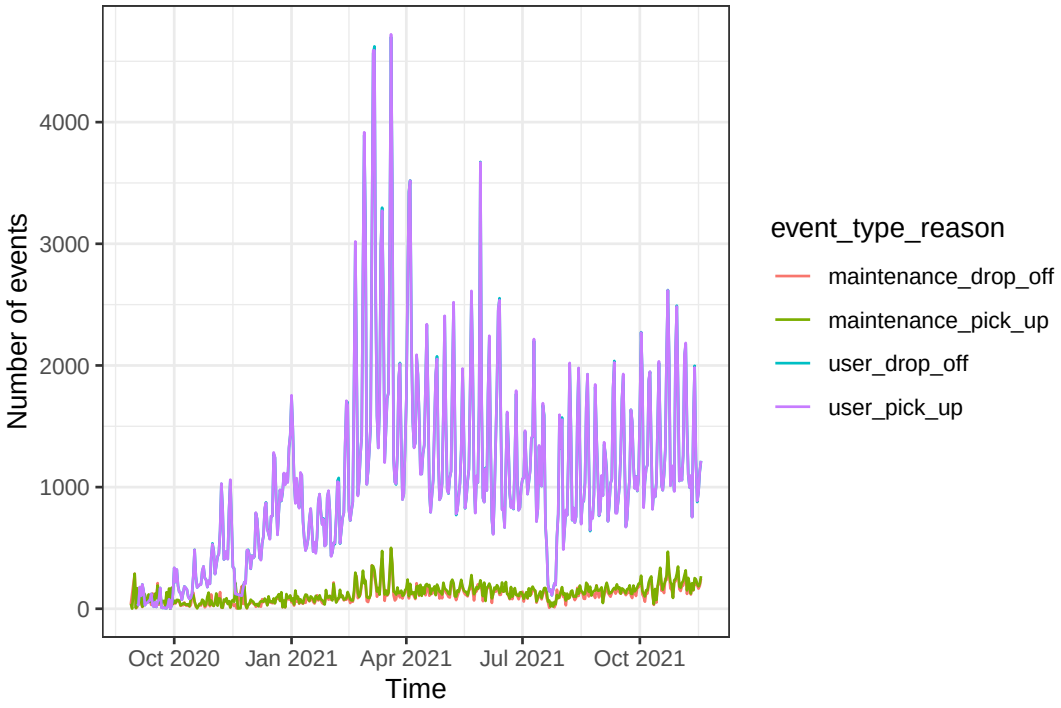


Figure 2.13: Frequency of each type of status update, recorded by a scooter whenever it gets used by either a customer or for maintenance, in the Beam Status Data.

valid, for example a pair consisting of two pickup events or two dropoff events. These are highlighted in the table. We assume that these pairs are caused by errors in the data, for example repeated or missing rows. To account for this, we infer events that must have occured in the system for the events present in the data to become valid. Section 2.6 discusses this process.

As this dataset provides data on all changes in the Beam e-scooter system, it provides sufficient information for modelling demand across the service area. Due to this, the Beam Status Dataset is the primary dataset used in later analysis, with the RIDE datasets used as a comparison.

Event Pair	Occurences	Mean time between events (s)
user_pick_up → user_drop_off	468,548	671.02
user_drop_off → user_pick_up	422,438	41,559.31
maintenance_pick_up → maintenance_drop_off	50,682	19,010.81
maintenance_drop_off → user_pick_up	45,623	80,099.13
user_drop_off → maintenance_pick_up	45,417	84,324.46
maintenance_pick_up → maintenance_pick_up	6990	664,670.9
maintenance_drop_off → maintenance_pick_up	4874	135,725.9
user_drop_off → user_drop_off	258	1,137.86
user_pick_up → user_pick_up	66	51,193.60
maintenance_pick_up → user_pick_up	23	67,121.87
user_pick_up → maintenance_pick_up	7	57,201.55
user_drop_off → maintenance_drop_off	4	767,119.7
maintenance_drop_off → maintenance_drop_off	3	9.21
maintenance_drop_off → user_drop_off	1	49,380.81

Table 2.4: Frequency of pairs of events in the Beam Status Dataset, along with the mean time between the events. Invalid pairs of events are highlighted.

2.5 Battery Usage

As e-scooters are powered solely by batteries, investigating the battery usage and recharging patterns is an important component to modelling how the systems operate. To do this, we can use the battery data present in the RIDE Snapshot Data, analysed in Section 2.3.2, and the Beam Status Data, analysed in Section 2.4.2. These data sets provide the current scooter battery level as a percentage. In the case of the RIDE data, this is between 0 and 100, while in the Beam data this is between 0 and 1. We divide the battery levels in the RIDE data by 100 to standardise the measurements, so all battery levels are between 0 and 1.

As the RIDE snapshot data only contains scooter battery levels every 30 minutes, we cannot accurately estimate the battery usage of a single trip. To account for this, we round the time of both ends of the trips to the snapshot that occurs prior, and in doing so cause trip lengths to be multiples of 30 minutes. For example, if a trip started at 6:15pm and ended at 7:20pm, we round the trip endpoint times to 6:00pm and 7:00pm. We also remove any trips with length zero after the rounding, so only trips that contain a snapshot are used. This approximation will cause an inaccurate estimate of battery usage, however it is sufficient for determining the magnitude of the battery drain. The Beam status data contains the battery level at the start and end of every trip, and so the battery usage estimates are more accurate.

Firstly, we investigate the battery data for RIDE scooters. To obtain a picture of how the battery changes over time in general, Figure 2.14 shows how the battery levels of the four most-used scooters (by number of trips) change over time. In general, the battery levels decrease as expected. However, the decreases do not always coincide with trips, as seen with the scooter "OHDD" in late June, where there is a rapid drop in the battery level prior to a trip taking place. The batteries are recharged regularly, with the battery level only reaching zero in two instances. There are several issues regarding the accuracy of the measurements, however. Firstly, very few battery readings occur between 0.85 and 1, despite the level reaching 1 multiple times across all scooters. This suggests that if the battery is charged above a certain threshold, the reported level is 1 regardless of the actual charge. A similar effect is shown for scooters that drop below a certain battery level, where the reported charge is 0. This is clearly shown by the existence of trips using scooter "OHDD" while it reported a battery level of zero. We also see a small drop in the reported battery level whenever a trip is taking place, then an increase in battery level after the trip has concluded. This indicates that there is an error in the battery level reporting while a vehicle is being used. Additionally, as the battery levels are only available in the half-hourly snapshots, we do not know the exact battery usage during trips. Due to these issues, any battery models using the RIDE data may be inaccurate.

To estimate the battery usage of RIDE scooters, we fit a linear model to the change in battery level against the *time in the trip*. We define the time in the trip to be the time since the first snapshot that took place after to the trip starting, and is computed for every snapshot during the trip until the first snapshot after the trip ends. As an example, consider a trip that starts at 9:03am and finishes at 10:12am. The first snapshot after the trip starts at 9:30am, so the time in the trip at the snapshot at 10:00am is 30 minutes, and at 10:30am is 60 minutes. As these occur at snapsot times, we also have the battery level at each time so we can model the battery usage during trips. This method may introduce error due to the estimation of trip duration; the trip is assumed to take the entire time between the two snapshots. The estimated battery change per second during trips is -2.541×10^{-6}, approximately 0.9% of the total capacity of an e-scooter battery per hour. This rate of battery usage is small relative to the average charge of e-scooters,

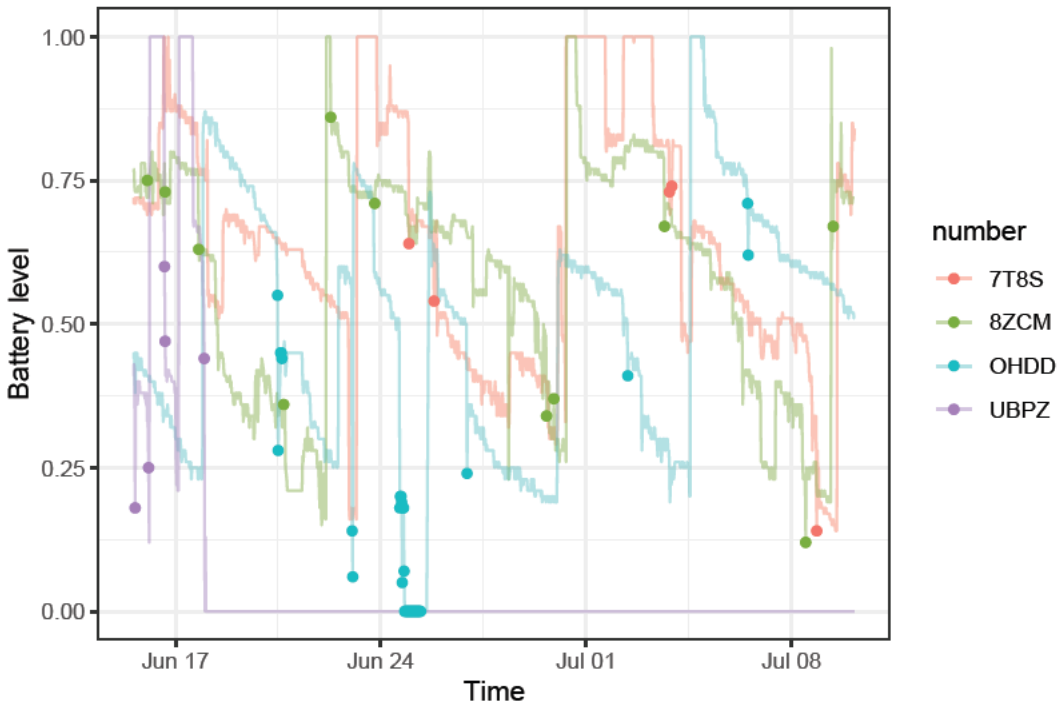


Figure 2.14: Battery levels of four vehicles in the RIDE system. The paths shown are the battery levels at each snapshot. The points highlight the battery levels during trips.

shown in Section 2.3.2 to be between 45% and 65% at all times. Hence, battery levels are unlikely to have a significant effect on the RIDE data.

We also investigate the battery changes in the Beam Status Dataset, which contains the battery level of a scooter whenever it undergoes a status change. This allows us to accurately estimate the battery usage during trips. To calculate the battery usage rate, we first define a scooter as *in use* at time t if the status update before t has the event type `user_pick_up` and the status update after t has type `user_drop_off`. We also define the scooter to be `idle` at time t if the status update before t is a dropoff of either type, and the status update after t is a pickup of either type. These two variables determine the current status of the scooter at time t - if the scooter is available for hire, it is `idle` and not `in use`. If a scooter is being used for a trip, it is `in use` and not `idle`. If a scooter is under maintenance, it is both not `idle` and not `in use`.

For each scooter, we compute the two indicator variables above, the change in battery level since the previous status update, and the time difference between the updates. The average change in battery is then the sum of all of the battery level changes divided by the sum of the time differences. This is repeated for each combination of the indicator variables to estimate the battery usage rate for scooters that are in use by a user, are idle, or are under maintenance. These estimates are shown in Table 2.5.

Scooter status	Battery usage rate (s^{-1})	Battery usage rate (hr^{-1})
In maintenance	-3.515648×10^{-7}	-0.00127
In a trip	-3.859802×10^{-5}	-0.139
Idle	-1.080345×10^{-7}	-0.00039

Table 2.5: Estimated battery usage rates for Beam e-scooters in different situations.

The battery usage rate for a scooter currently in use for a trip is 13.9% of the scooter battery's capacity per hour. This is much higher than the estimated rate for RIDE scooters, which was 0.9% per hour. This may be due to the approximations used in estimating the battery usage for RIDE scooters, or differences in the physical hardware of the scooters. However, the battery usage is still low enough that for an average trip, lasting approximately 11 minutes, the battery only drains by approximately 2.5%.

As we have shown, the rate of battery decay during trips is low relative to the maximum charge of a scooter. Combined with the high minimum charge of scooters that have been deployed, it is unlikely that the battery levels would have a large impact on the deployment of e-scooters. Hence, we assume that recharging is performed such that all scooters are sufficiently charged for any trips that may occur, so further modelling of battery usage is not required. This may not be a reasonable assumption, especially when considering possible variations of this research including alternative vehicle types, and so incorporating battery usage into the models described in Chapters 4 and 5 may improve accuracy.

2.6 Redistribution Estimation

In the data cleaning and processing performed in previous sections, we remove some of the data. As we discuss later in Section 3.5.2, in order to estimate the demand for e-scooters we need to accurately track the numbers of scooters in different locations. However, in the filtering process we may remove some of the data that is critical to accurately tracking the positions of scooters over time. For example, consider a scooter that gets hired in the service area, dropped off outside the service area, then returned to the service area. In this case, the rows in the trip data sets corresponding to the movements outside the service area are removed in the filtering process, so the scooter appears to move between two locations without a trip taking place. In addition to errors introduced by the processing, the datasets themselves may contain errors such as duplicated or missing rows. This was observed in the Beam Status Data, where there exist a large number of pairs of events that are not feasible, such as a scooter being hired twice in a row. Hence, we need to infer the vehicle movements and redistributions that occur without being recorded in the data prior to estimating the demand for e-scooters. In this section, we describe how this process was implemented for the Beam Status Data set. We also break the service area into discrete locations using a 100×100 grid, with each cell being approximately 60m in width. The binning process is described in detail in Section 3.5.4.

There are two types of errors that may occur that could cause issues for the scooter tracking algorithm. The first is that the location of the scooter, which in the case of binned data is the number of the cell the event occurs in, may change unexpectedly. An example of this is if a scooter is dropped off in one cell, then picked up in an adjacent cell with no explanation for moving between the two cells. A second type of error that may occur is in the status data, if a scooter records a pair of events that are not possible, for example two hires in a row. This may be caused by errors in the data sets such as missing or duplicated data, or caused by the filtering process removing some points in the datasets.

We first consider calculating the unexpected movement of vehicles. We denote these movements *redistributions*, as they may be due to the ride-sharing operator redistributing vehicles to areas with higher usage or users moving scooters without hiring the vehicle first. A given pair of sequential status updates (i, j) for a single scooter indicates a redistribution is present if:

- Status update i takes place directly before update j; and,
- At update i, the event type is a dropoff; and,
- At update j, the event type is a pickup; and,
- At update i, the scooter is left in cell a, and at update j the scooter is left in cell $b \neq a$.

To correct for the redistribution, we add two additional events to the dataset whenever the above conditions apply, of types `redistribution_pick_up` and `redistribution_drop_off`. The location of the `redistribution_pick_up` is the same as the location of update i, and the location of the `redistribution_drop_off` is the same as the location of j. As we do not know the actual time of the movements of the scooters, we assume that the scooter spends equal time in both cells, and that this time is maximised. Hence, the time of the redistribution events is set to the midpoint between update i and update j, offset by $\pm$ 1 second to avoid ordering issues. By setting the redistribution time to halfway between the two updates in the data, we avoid significantly modifying the demand estimates. This process is repeated for all scooters in the data to provide a feasible sequence of locations through the Status Data.

Correcting for the second type of error follows a similar process. Firstly, we define an invalid pair of events to be status updates where the event types are infeasible. They can be categorised into the following:

- Two pickup (dropoff) events in a row; there must be a dropoff (pickup) event between the two. This type is the most common in the data, with 7352 occurences.
- A user (maintenance) pickup followed by a maintenance (user) dropoff; there must be a user (maintenance) dropoff then a maintenance (user) pickup between the two events. This type of error does not occur in the data, however we implemented a check for this.

We correct for this issue by inserting additional events to the Status Data. To distinguish them from the redistributions above and the data itself, we denote the events `extra_pick_up` and `extra_drop_off`. Where two pickups occur in a row, a dropoff event is added, and if two dropoffs occur a pickup event is added. In the former case, the location of the dropoff is at the second pickup, and for the latter case the pickup occurs at the first dropoff. This ensures that the scooter movements are explained by the combined Status Data after processing. The time of the extra events is assumed to be the midpoint of the time interval we are considering; halfway between the two pickups, or halfway between the two dropoffs.

These methods were implemented on the Beam Status Data. Figure 2.15 shows the frequency of all event types each day, including both the data and the imputed events. Both of the redistribution event types have frequencies that appear to match the user trips well, with the same peak in March 2021. The number of these events is lower than the number of user events, with a peak of 2408 redistributions on the 20th of March 2021, compared to the peak of 4627 user trips the same day. This indicates that redistributions may occur simply due to the increased system usage. GPS accuracy may cause some of these redistributions, as the increased number of status updates increases the probability of a scooter seemingly jumping over a cell boundary. We also see that the redistribution

pickup and dropoff events are balanced; each day, the number of each event is equal. However, this behaviour is not exhibited in the extra events. The number of these events is much lower, with a peak of 118 dropoff events on the 24th of November 2020 and 24 pickup events on the 7th of February 2021. The imbalance between the frequency of extra events is due to the types of invalid pairs of events not being uniform; by far the most frequent occurence is a repeated maintenance pickup, which causes an extra dropoff event to be added.

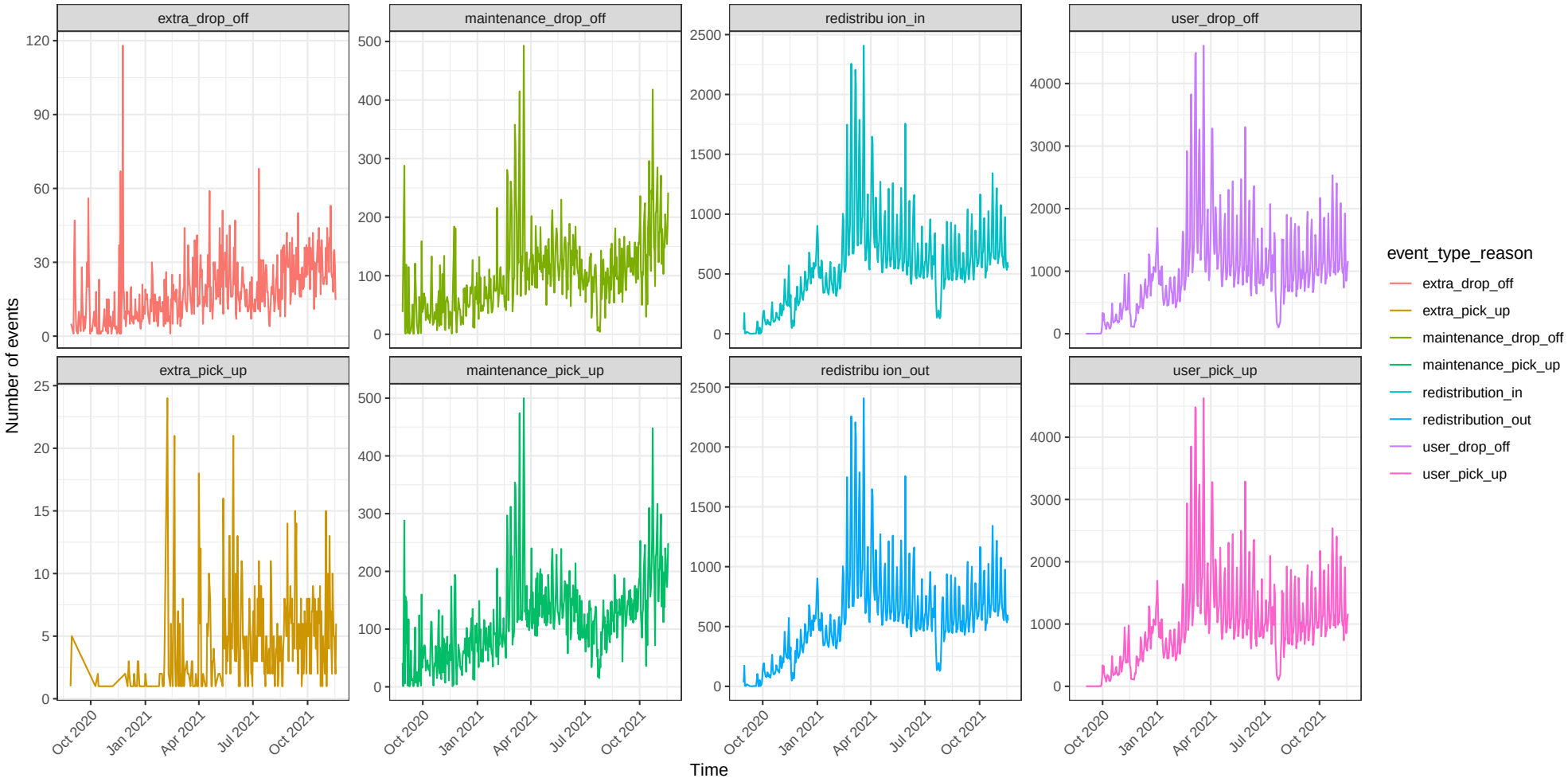


Figure 2.15: Frequency of all event types in the Beam Status Data, including imputed "redistribution" and "extra" event types. The redistributions, corresponding to unexplained movement of scooters, appear to correlate to the user and maintenance trips in the data, however the extra events do not, with the large peak in March 2021 not present.

2.7 Conclusion

In this chapter, we discussed the data provided by RIDE and Beam. Both e-scooter ride-sharing companies operated in Adelaide, with RIDE operating in 2019 and Beam operating from 2020 to present. Both datasets contained Trip Data, which provided information on all trips that occurred within the systems. From this data, we gathered several key insights. The hire frequency is much higher in both systems in the afternoon and evening, with Fridays and Saturdays experiencing more hires than other days in this time. The average trip length is approximately 750m for the RIDE system, slightly longer than the mode of 600m for the Beam data. However, the average trip duration for RIDE data is shorter than that of the Beam trips, with 7.5 minutes compared to 11 minutes.

We also obtain insights from the datasets unique to each system. Part of the RIDE data consists of snapshots, which show the current status of the system at regular, half-hourly intervals. This allows us to know the exact location of all scooters through time, as well as their status. From this, we determined that the battery level of the e-scooters is likely to not influence the ability for scooters to be hired, if the current recharging scheme is maintained. Similar results were obtained from the Beam Status Data, which contained entries for every time a scooter's status changed, for example when a trip starts or ends. The battery usage for Beam scooters appears to be higher than the battery usage for RIDE scooters, however is likely to still not impact results. The battery usage of vehicles is, however, a potentially useful direction for future research as recharging is an important aspect of operating an electric vehicle ride-sharing system, and optimising the recharging process may lead to significant efficiency increases.

The data analysis performed here provides the foundation for all future chapters in this work. The time-based hire rates lead directly to the time-splitting performed in Chapter 3, where we confirm that the demand for e-scooters significantly fluctuates through time. We also utilise metrics such as trip duration and distance in our simulation models used in Chapters 4 and 5 to improve the accuracy of the simulations. The redistribution estimation performed also allows for the locations of all scooters to be tracked accurately as they move through the service area, which is used in the demand estimation to determine the number of scooters in each location through time.

Chapter 3

Demand Estimation

3.1 Introduction

To accurately model an e-scooter ride sharing system, we require estimates of the demand for vehicles across the service area. To construct these estimates, we assume that people are willing and able to walk a short distance to hire a scooter. This allows us to discretise the service area into finitely many cells, which simplifies the demand estimates by only requiring a single estimate for a finite number of locations rather than a continuous distribution. We then model the demand individually in each cell.

In this chapter, we develop multiple estimates of demand for e-scooters based on the data. Section 3.2 describes a simple estimate, based on the assumption that hires are independent and identically distributed (i.i.d.) through time. In this estimate, we assume that the demand for e-scooters only depends on location, and does not change through time, and that vehicles are always available in all locations. We improve on this estimate in Section 3.3 by accounting for the property that hires cannot occur, and are not recorded in the data, if a scooter is not present in an area. Section 3.4 further improves upon the demand estimates by introducing a Bayesian framework to limit the effect of cells with limited data. In all of these demand estimates, we assume that the demand in each cell is independent of the demand in other cells.

In Section 3.5.4, we discuss the various types of binning methods we use to analyse the demand patterns across the service area, including a grid of equally-sized cells and one based on k-means clustering. The binning process is vital to the demand estimates, as it controls both the resolution of the estimates and the amount of data used in the demand estimates in each area. The binning performed also determines the state space for the simulation model implemented in Chapter 4, and the discrete locations where scooters can be placed in the deployment optimisation in Chapter 5.

Section 3.6 discusses the demand estimates in detail. In this section we also show how demand varies across time, in both long-term usage trends and weekly and daily demand

cycles. Finally, in Section 3.7 we discuss drawbacks to the demand estimates and possible avenues for future research.

The demand estimates derived in this chapter are then used to inform the simulation models described in Chapter 4, and hence form the basis of the fleet deployment optimisation in Chapter 5.

3.2 Naïve Demand Estimation

As a baseline for comparison with more sophisticated demand estimation techniques, we construct a simple demand estimate. First, we need to consider the stochastic process governing hire times. We assume that the hire process in each cell follows a Poisson point process. In particular, this assumes that the hires are independent and identically distributed (i.i.d.), and that the intervals between each hire time t_i are exponentially distributed with a common rate λ; i.e.,

$$t_i - t_{i-1} \sim \text{Exp}(\lambda) \quad \text{i.i.d.},$$

for $i = 1, ..., n, t_0 = 0$, where n is the total number of hires in the cell. An estimate of the demand, λ, can then be obtained through maximum likelihood estimation.

The likelihood of having n hires in T time – the event $\{N(T) = n\}$ – given a Poisson point process is

$$\mathbb{P}(N(T) = n|\lambda) = \frac{e^{-\lambda T}(\lambda T)^n}{n!}.$$

The maximum likelihood estimate of λ, $\hat{\lambda}$, can then be calculated by setting the derivative of the log-likelihood, $l(\lambda) = \log(\mathbb{P}(N(T) = n|\lambda))$, equal to 0:

$$\begin{aligned} l(\lambda) &= -\lambda T + n\log(\lambda T) - \log(n!) \\ \therefore 0 &= -T + \frac{n}{\hat{\lambda}} \\ \therefore \hat{\lambda} &= \frac{n}{T}. \end{aligned}$$

Hence, the maximum likelihood estimate for demand is simply the total number of hires, n, divided by the time window, T. We denote this estimate $\hat{\lambda}_{\text{n}}$, so that

$$\hat{\lambda}_{\text{n}} = \frac{n}{T}. \tag{3.1}$$

The naïve estimation makes several assumptions that are potentially unrealistic and may impact the accuracy of the demand estimates. One of these is the assumption that scooters are always available for hire. As the demand during the periods of time with no scooters is not reflected in the data, this would lead to underestimating the true demand for e-scooters.

3.3 Vehicle Availability Correction

To account for lower demand in the naïve estimate, we model the hire process as an independent Poisson process within each period of time when scooters are present in a cell. Suppose that there are M intervals of time that scooters are present, the number of hires in each of these intervals is n_j, and the interval lengths are T_j, for $j = 1, \ldots, M$. As hires must occur in these intervals, the number of hires in these intervals add up to the total number of hires, $\sum_{j=1}^{M} n_j = n$. We also assume that the demand rate is constant across all of these intervals within one cell. The joint density of the hires is then

$$\begin{aligned}
\mathbb{P}(N_j(T_j) = n_j \quad \forall j = 1, ..., M|\lambda) &= \prod_{j=1}^{M} \mathbb{P}(N_j(T_j) = n_j|\lambda) \\
&= \prod_{j=1}^{M} \frac{e^{-\lambda T_j}(\lambda T_j)^{n_j}}{n_j!} \\
&= \frac{\exp\left(-\lambda \sum_{j=1}^{M} T_j\right)(\lambda T)^{-\sum_{j=1}^{M} n_j}}{\prod_{j=1}^{M} n_j!}.
\end{aligned}$$

The maximum likelihood estimate can once again be derived:

$$\begin{aligned}
l(\lambda) &= -\lambda \sum_{j=1}^{M} T_j - \log(\lambda T) \sum_{j=1}^{M} n_j - \log\left(\prod_{j=1}^{M} n_j!\right) \\
\therefore 0 &= -\sum_{j=1}^{M} T_j + \frac{\sum_{j=1}^{M} n_j}{\lambda} \\
\therefore \hat{\lambda} &= \frac{\sum_{j=1}^{M} n_j}{\sum_{j=1}^{M} T_j} \\
&= \frac{n}{\sum_{j=1}^{M} T_j}.
\end{aligned}$$

Hence, an estimate of the demand within a cell that takes into account the potential unavailability of scooters is the total number of hires divided by the total amount of time at least one scooter is present. We denote this demand estimate $\hat{\lambda}_{\text{v}}$, so that

$$\hat{\lambda}_{\text{v}} = \frac{n}{\sum_{j=1}^{M} T_j}. \tag{3.2}$$

As the sum of lengths of intervals of time where scooters are present, $\sum_{j=1}^{M} T_j$, must be less than the overall time window T, we find that $\hat{\lambda}_{\text{v}} \geq \hat{\lambda}_{\text{n}}$ in all cases. The two demand estimates are equal iff scooters are available in the cell for all time.

3.4 Bayesian Demand Estimation

One of the issues that the MLE framework faces is that if scooters are not present in a location for a large proportion of the overall time window, the estimate may be inflated. For example, if a scooter is dropped off in an otherwise unused location and then picked up quickly, the demand would only depend on how long that scooter remained in the location. We will show in Section 3.5.4 that this is a noticeable issue in this dataset. Bayesian methods can be used to limit the effect of this error.

Consider a prior for the demand rate λ, $p(\lambda)$. We take the prior to be an exponential kernel with rate n/T, i.e. the naïve demand estimate. This allows the prior demand to vary across the service area. Using an exponential prior also has the effect of heavily weighing down demand estimates with few hires, while having a limited effect on demand estimates with large numbers of hires.

To estimate demand, we first derive the posterior distribution, $\mathbb{P}(\lambda|N_j(T_j) = n_j \; \forall j = 1, ..., M)$. The prior is

$$p(\lambda) = \frac{T}{n} \exp\left(-\frac{T}{n}\lambda\right),$$

and the likelihood is

$$\mathbb{P}(N_j(T_j) = n_j \; \forall j = 1, ..., M|\lambda) = \frac{\exp\left(-\lambda \sum_{j=1}^{M} T_j\right) (\lambda T)^{-\sum_{j=1}^{M} n_j}}{\prod_{j=1}^{M} n_j!}.$$

Therefore, by Bayes theorem,

$$\begin{aligned}
\therefore \mathbb{P}(\lambda|N_j(T_j) = n_j \; \forall j = 1, ..., M) &\propto p(\lambda)\, \mathbb{P}(N_j(T_j) = n_j \; \forall j = 1, ..., M|\lambda) \\
&\propto \exp\left(-\frac{T}{n}\lambda\right) \frac{\exp\left(-\lambda \sum_{j=1}^{M} T_j\right) (\lambda T)^{-\sum_{j=1}^{M} n_j}}{\prod_{j=1}^{M} n_j!} \\
&\propto \exp\left(-\lambda \left(\frac{T}{n} + \sum_{j=1}^{M} T_j\right)\right) \prod_{i=1}^{M} \lambda^{n_j}.
\end{aligned}$$

Taking the log and setting its derivative to 0, we find the maximum *a posteriori* estimate

(MAPE):

$$l(\lambda) = -\lambda\left(\frac{T}{n} + \sum_{j=1}^{M} T_j\right) + \sum_{i=1}^{M} n_j \ln(\lambda)$$
$$\therefore \frac{\partial l}{\partial \lambda} = \frac{T}{n} + \sum_{j=1}^{M} T_j + \frac{n}{\hat{\lambda}} = 0$$
$$\implies \hat{\lambda} = \frac{n}{\frac{T}{n} + \sum_{j=1}^{M} T_j}.$$

We denote this demand estimate $\hat{\lambda}_{\mathrm{b}}$, so that

$$\hat{\lambda}_{\mathrm{b}} = \frac{n}{\frac{T}{n} + \sum_{j=1}^{M} T_j}. \tag{3.3}$$

As $T/n > 0$, we have $\hat{\lambda}_{\mathrm{b}} < \hat{\lambda}_{\mathrm{v}}$.

3.5 Implementation

3.5.1 Naïve Demand Estimation

As discussed in Section 3.2, an initial estimate of the demand in each cell is the naïve estimate for each cell. To apply this estimation technique to both the Beam and Ride data sets, we require only the trip data. From this data, we obtain the total number of trips in each area, as well as the time window that the data covers. However, to estimate demand by location, we require a binning method, to discretise the spatial component of the data. In this section, we consider a simple 50×50 grid across the service area. Section 3.5.4 discusses variations of this grid, as well as alternative binning algorithms.

Figure 3.1 shows the estimated demand for both sets of data, and Table 3.1 lists the top 10 cells by demand in each data set. As shown, the popular locations are similar between the two datasets. For example, cells 1221 and 1123 appear as the most popular locations in both lists; these correspond to the corner of King William Road and Rundle Mall, and outside the Adelaide Railway Station respectively. As the train station is likely to be used extensively for transportation, this indicates that scooters are being used for last-mile transportation. The high demand at the corner of King William Road and Rundle Mall is likely due to the restriction on e-scooters in Rundle Mall, and the central location. Other cells in common include 1521 and 1522, which are located near the corner of Rundle Street and Pultney street, which is another busy intersection in Adelaide.

There are several cells that are unique to one system. In the Beam dataset, cells 1722 and 1822, located near the corner of East Terrace and Rundle Street, have a relatively high rate of hires, whereas in the Ride dataset it is not a popular location. Conversely,

in the Ride data set cells 1558 and 1559, located near Bonython Park in the northwest of the service area, are hotspots in demand.

The demand estimates also show that demand for Beam scooters is significantly higher than the demand for Ride scooters in the hotspot areas, with an estimated demand of approximately 1.033 hires per hour compared to 0.182 per hour outside the train station.

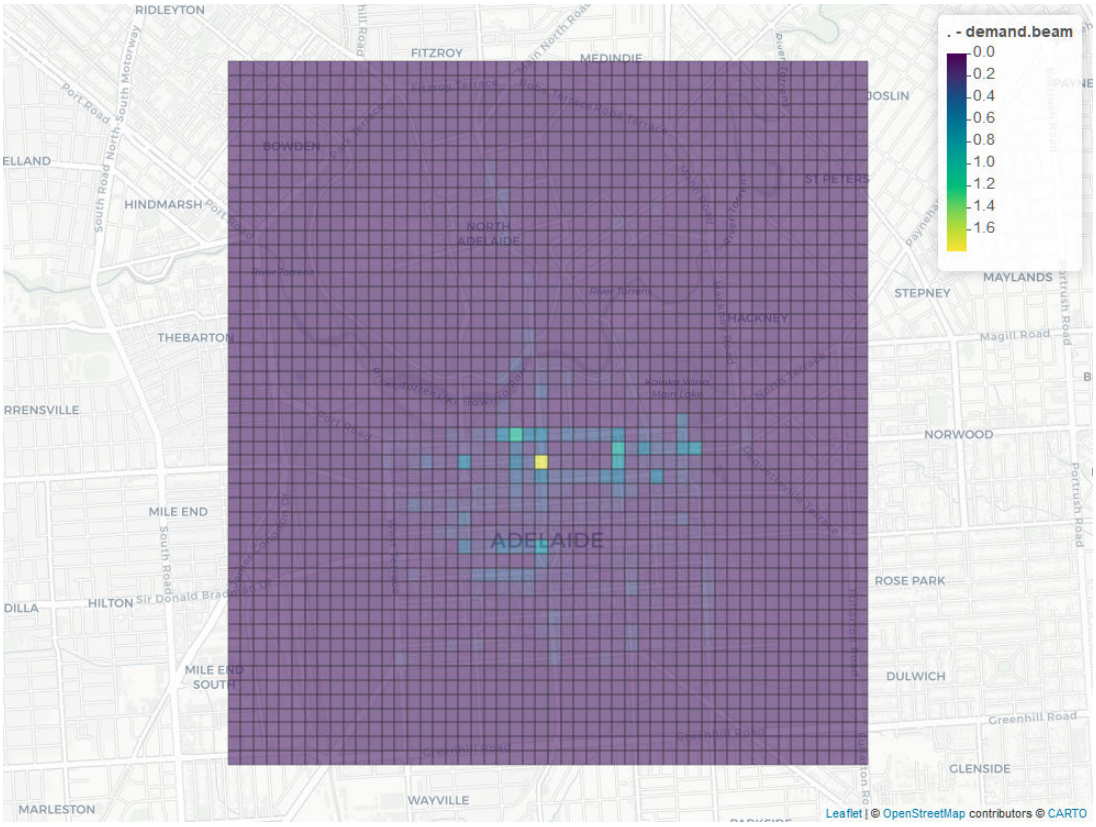
Beam		Ride	
Cell	Demand (hr^{-1})	Cell	Demand (hr^{-1})
1221	1.793	1123	0.182
1123	1.083	1221	0.169
1522	0.912	1558	0.121
1521	0.718	1521	0.114
1215	0.654	1559	0.103
1822	0.575	1122	0.080
1073	0.479	1215	0.069
1772	0.458	1013	0.068
1223	0.445	1220	0.062
921	0.433	1522	0.061

Table 3.1: Cells with the highest demand, using the naïve method, in both the Beam and Ride data. Shaded cells are common between the two systems.

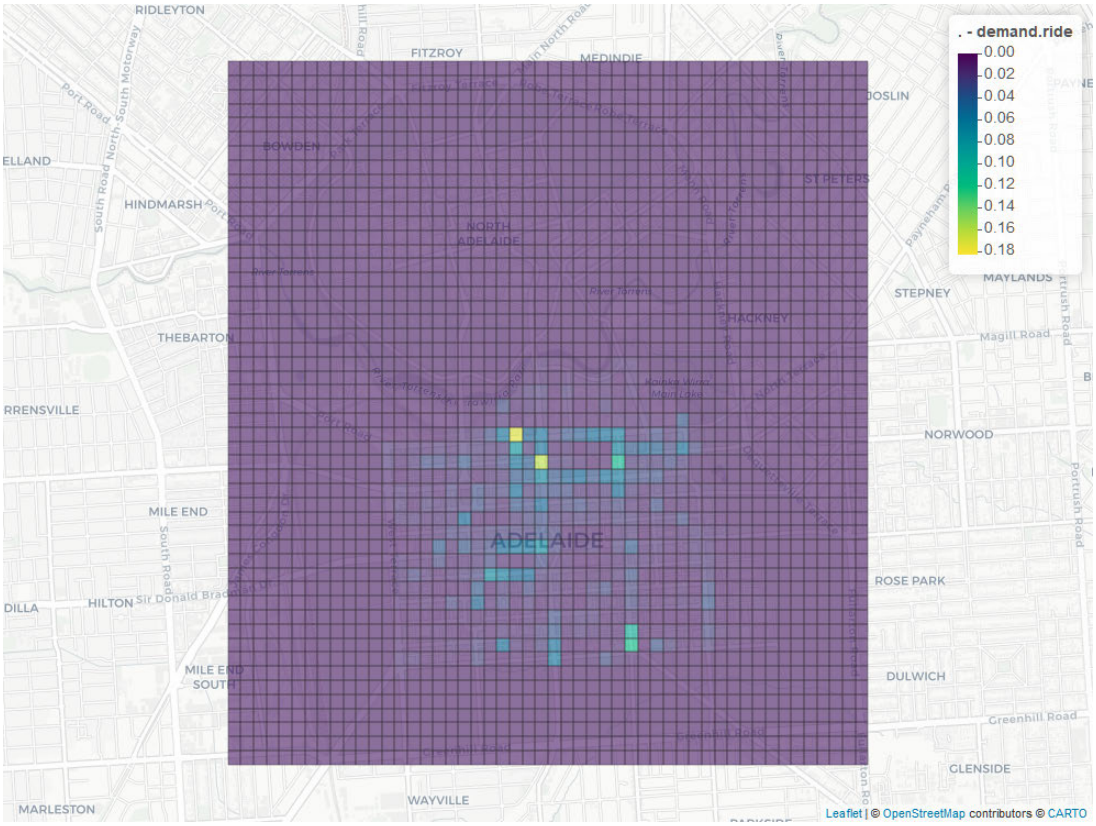
However, as discussed in Section 3.3 there is an issue in these estimates, in that scooters are not always present when a hire is requested. This is seen by plotting the proportion of cells that have scooters over time, as shown in Figure 3.2. Clearly, a large number of cells do not have scooters for at least part of the time, with the proportion of cells with RIDE scooters approximately 4 times less than cells with Beam scooters. This leads to lower demand estimates than the areas should have, as no hires can occur while scooters are unavailable. Due to this, we implemented the demand estimate vehicle availability correction.

3.5.2 Demand Estimation with Vehicle Availability Correction

The demand estimate with the vehicle availability correction, derived in Section 3.3, is significantly more complicated than the naïve estimate as we need to record the number of scooters in each cell at each time, so that the total amount of time when scooters are present can be calculated. However, in the case of the Beam data we do not have the locations of scooters in the system for the entire time period we are considering, only the events that occur in the system. Hence, we have to solve two problems: how to find the number of scooters in each cell at specific points in time, and how to track the changes in scooter numbers over time between these known points.



(a) Naïve demand estimates for Beam e-scooters.



(b) Naïve demand estimates for RIDE e-scooters.

Figure 3.1: Naïve demand estimates, measured in hires/second. Due to the differing overall system usage, the colour scales are not the same between plots.

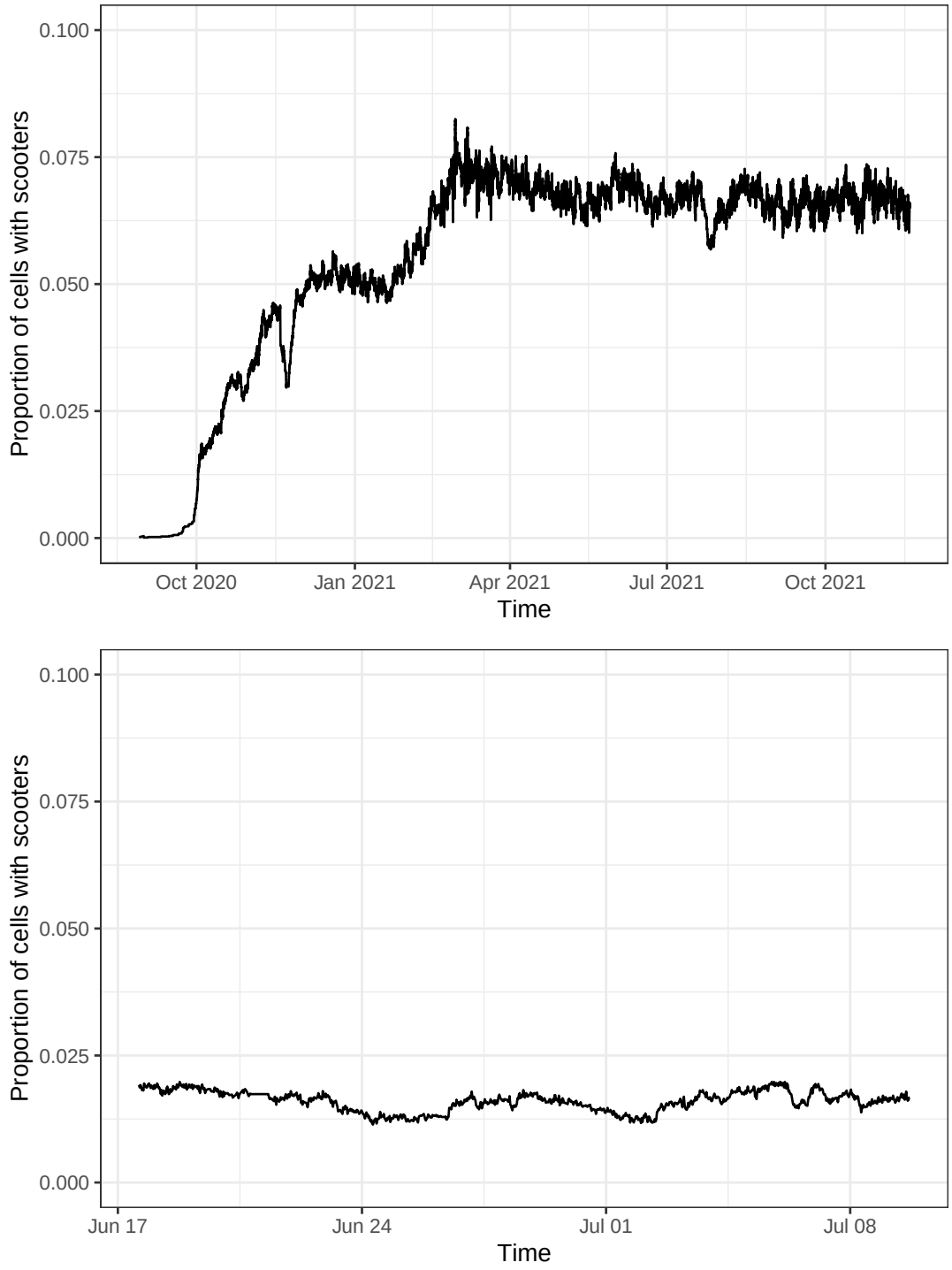


Figure 3.2: Proportion of cells where scooters are present over time, for Beam (top) and Ride (bottom) data sets, using a 50×50 grid for a total of 2500 cells. The proportion is sampled every hour in both data sets. The increased variability in the Beam proportion is due to the significantly expanded time window, as it spans over a year compared to less than one month.

In this and subsequent sections, we only consider the Beam data set due to the short time window of Ride data.

Consider the Beam dataset. To find the locations of scooters at a specific time, t, we take the location of a scooter to be either:

- The last location the scooter was dropped off, if the scooter has not been picked up between that time and t; or,
- The location the scooter is next picked up, if the scooter was not dropped off between t and that time; or,
- Unknown, if neither of the above are possible.

These unknown locations correspond to the scooters that are in trips at time t and so cannot be attributed to any location at t, or have not yet entered the system.

We then track the changes in the system using the status data set independently in each cell. If a dropoff of any type (from a user's trip, system maintenance or an inferred redistribution) occurs, the cell's population increases by 1, and if a pickup occurs the population decreases by 1. By looping through every entry in the status data set, we then find the cell populations over time, and hence the amount of time each cell has at least one scooter.

This method does have some limitations. The primary issue is that any errors in the data, for example missing entries or incorrect locations, impact the scooter counts at the time of the event and the scooter counts for all times after the event. While this effect is mitigated by the redistribution estimation described in Section 2.6, there may still be errors present. Due to this, we split time into months. At the start of each month, we calculate the locations of all scooters, then infer cell populations during the month. Appendix A shows the algorithm used in the demand estimation.

3.5.3 Bayesian Demand Estimation

As derived in Section 3.4, the Bayesian demand estimate for a cell is based on the number of hires in the cell, the length of time that at least one scooter is present, and the time window for the data. Hence, to calculate the Bayesian estimates we use the same algorithm as the standard demand estimates to find the length of time with scooters present, however with the alternative demand formula.

3.5.4 Spatial Binning

Several approaches can be taken to discretise the data. A simple approach is to create a grid of fixed-size, equi-spaced cells that span the service area, as we implemented above. This way, the size of cells can be controlled by a single parameter that would vary the

precision in demand estimates. However, this may cause issues such as splitting an area of high demand, as it is not necessarily informed by the data.

Binning methods such as k-means and mean-shift algorithms may solve this problem, as they allow the data to inform the binning process. K-means is particularly useful in this process, as it contains a controllable parameter for the number of groups. They may also reduce the number of cells required to obtain a reasonable estimate of demand across space, as areas without much activity that are similar can be combined into a single group.

In this section, we use the demand estimation algorithms to investigate the precision from using different types of binning, as well as the accuracy of the resulting demand estimates.

A similar approach was used by Weikl & Bogenberger (2015) to investigate vehicle movement in a free-floating car sharing system. In their work, a two-level binning method was used so that the movement can be analysed at both a macro- and micro-scale. Biondi et al. (2016) also implement a grid-based cell system using $100m \times 100m$ cells, each of which representing a potential station in a station-based car-sharing system. In their model, each of these cells has values such the local demand, similar to the demand estimation we describe in this chapter. Biondi it et al. then use per-cell demands to estimate the local demand across the service area, and to deploy stations. In the case of a free-floating system, Yu et al. (2021) discuss how the finite accuracy of GPS systems used in electric bikes creates an implicit grid in the data, where vehicles must be located at vertices.

Grid-Based Binning

The initial type of binning constructed was a simple grid across the service area. Four points were chosen in a rectangle that contained the Adelaide CBD and parklands, which form the corners of all grids. The data is also filtered to only the points within this rectangle. The grid is then constructed by evenly splitting the latitude and longitude into n bins, to create an $n \times n$ grid across the CBD. An example of this with a 4×4 grid is shown in Figure 3.3. The cells can then be indexed by latitude and longitude. If the longitude index is p and the latitude index is q, which each run from 1 to n, then the cell number is denoted

$$c = (q - 1) + n(p - 1),$$

so that c ranges from 0 to $n^2 - 1$ and each cell has a unique index. An example of a grid is shown in Figure 3.3.

As an initial attempt of binning, we used a 100×100 grid, for a total of 10,000 cells. Each of these cells was approximately 50m×70m. This size is large enough to contain a reasonable number of events, while also representing a reasonable search distance for a potential e-scooter customer. We also tested demand estimates with grid sizes ranging from 20×20 to 200×200. Figure 3.4 shows the estimated demand, using the MLE

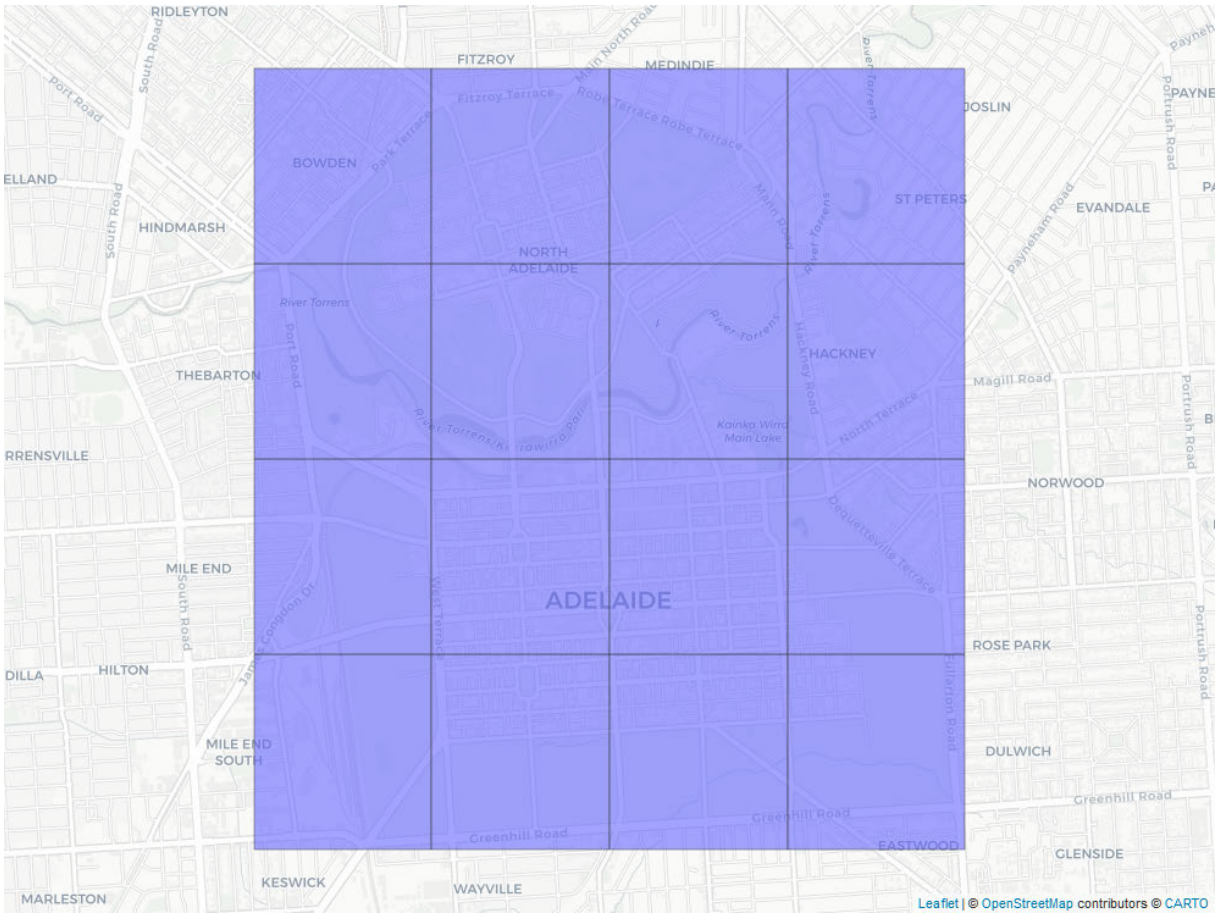
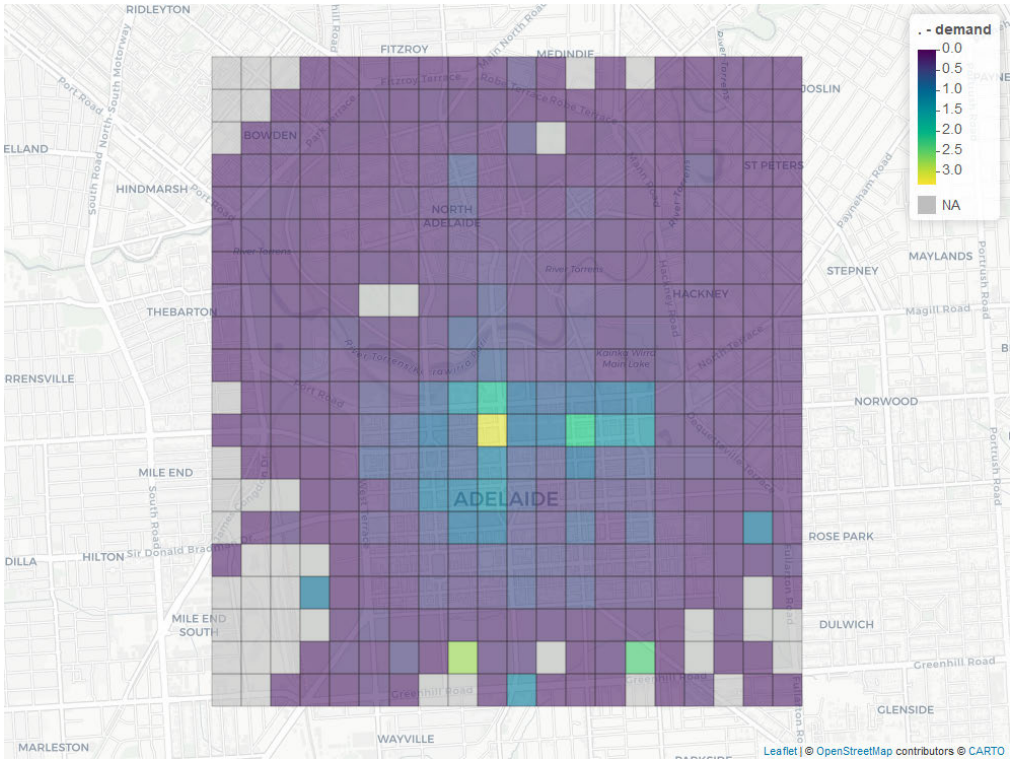


Figure 3.3: Example of constructing a 4×4 grid across the Adelaide CBD. Each cell is approximately 625m by 875m.

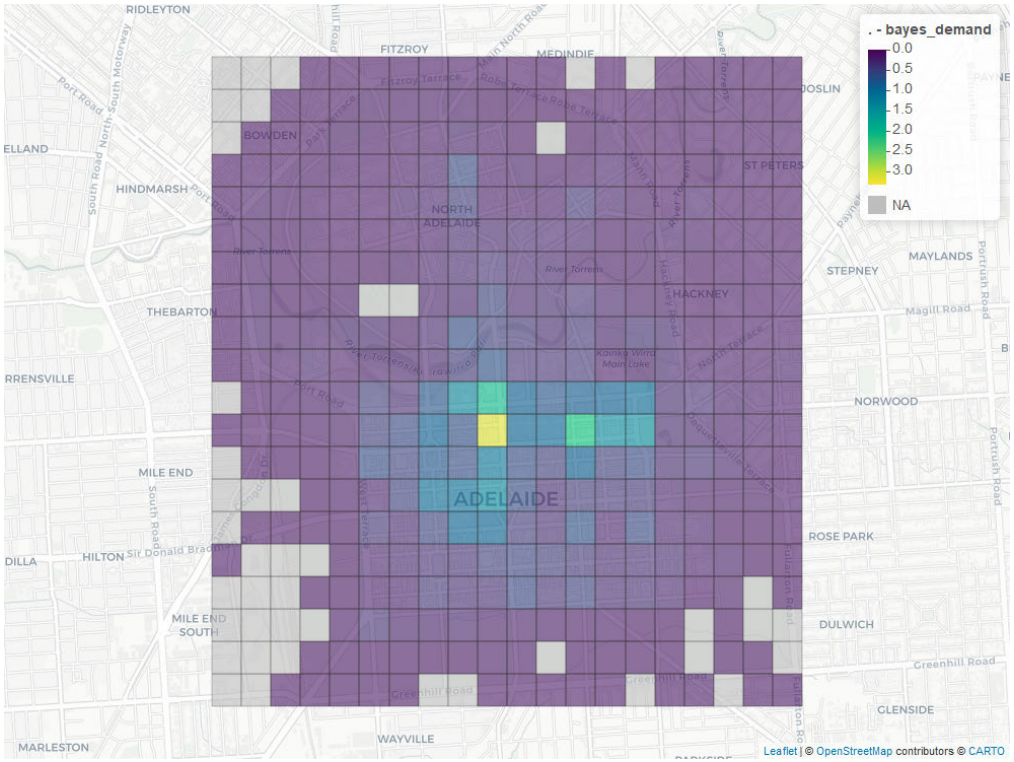
and Bayesian methods, for these different grids. In this section, we aggregate demand estimates over all months.

For a small number of cells, for example the 20×20 case, the standard demand estimates are very similar in general to the Bayesian demand estimates. However, in Figure 3.4a there are several cells near the South edge of the grid, on the bottom of the plots, with high estimated demand, that are not shown in the Bayesian estimates in Figure 3.4b. This difference is likely due to scooters not being present in the outer cells for a long period of time. This causes the Bayesian estimate to tend towards the prior estimate.

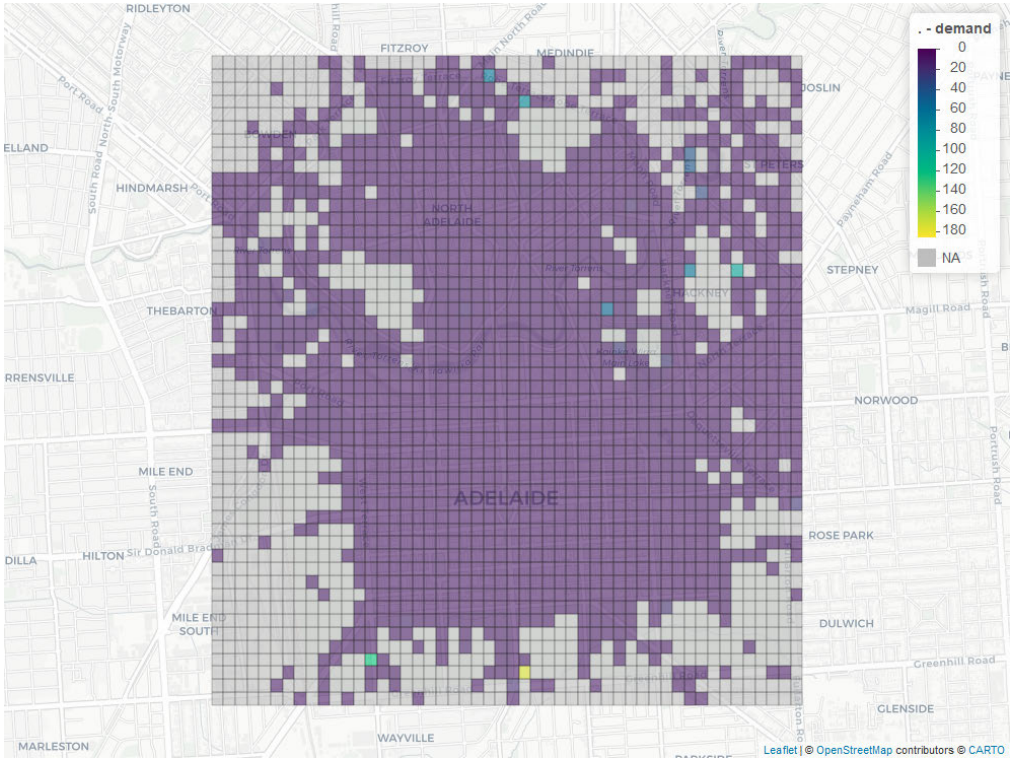
This effect is amplified in the cases of higher cell counts. For all grids greater than 20×20, the standard demand estimates are completely overshadowed by a small number of outlying cells near the edges of the city, where a few hires occur over a small duration of time with scooters present. The Bayesian estimates appear to be more consistent, with similar demand patterns appearing across all grid types.



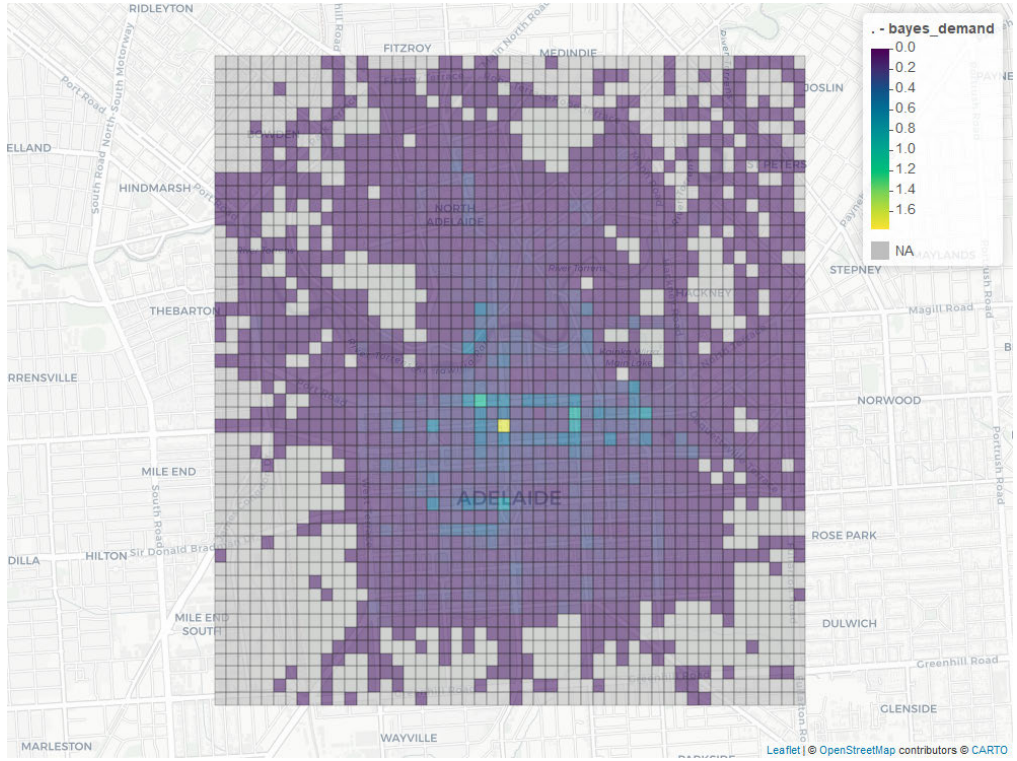
(a) Demand estimation with vehicle availability corrected estimate, 20×20 grid.



(b) Bayesian demand estimates, 20×20 grid.

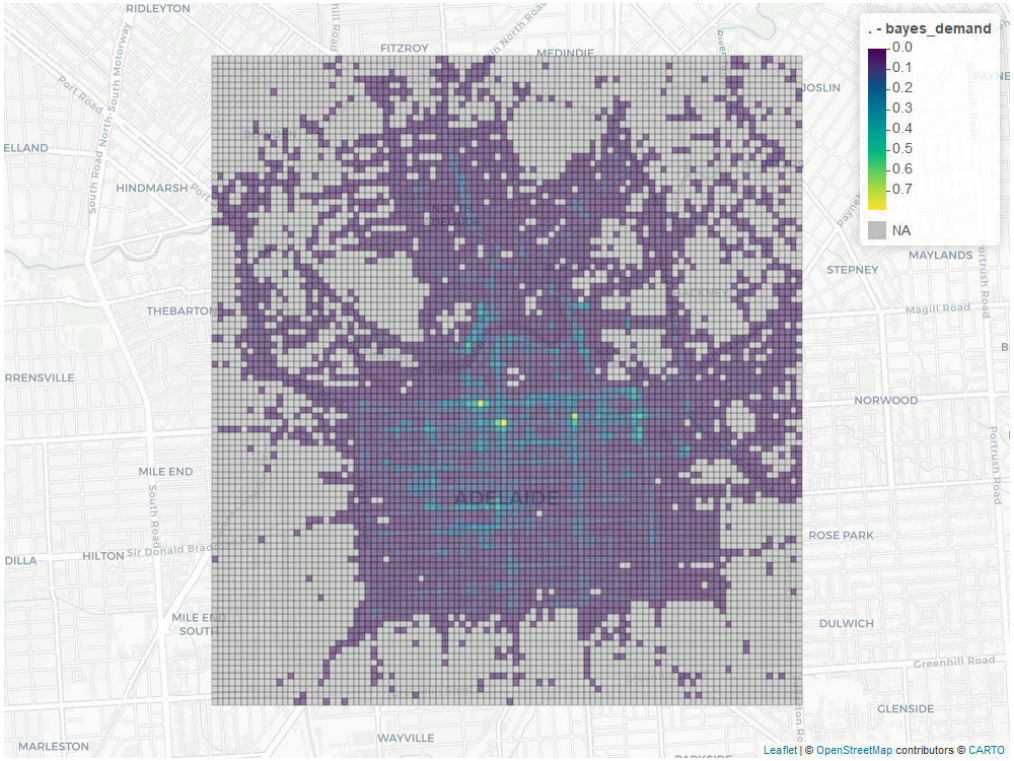


(c) Demand estimation with vehicle availability corrected estimate, 50×50 grid.

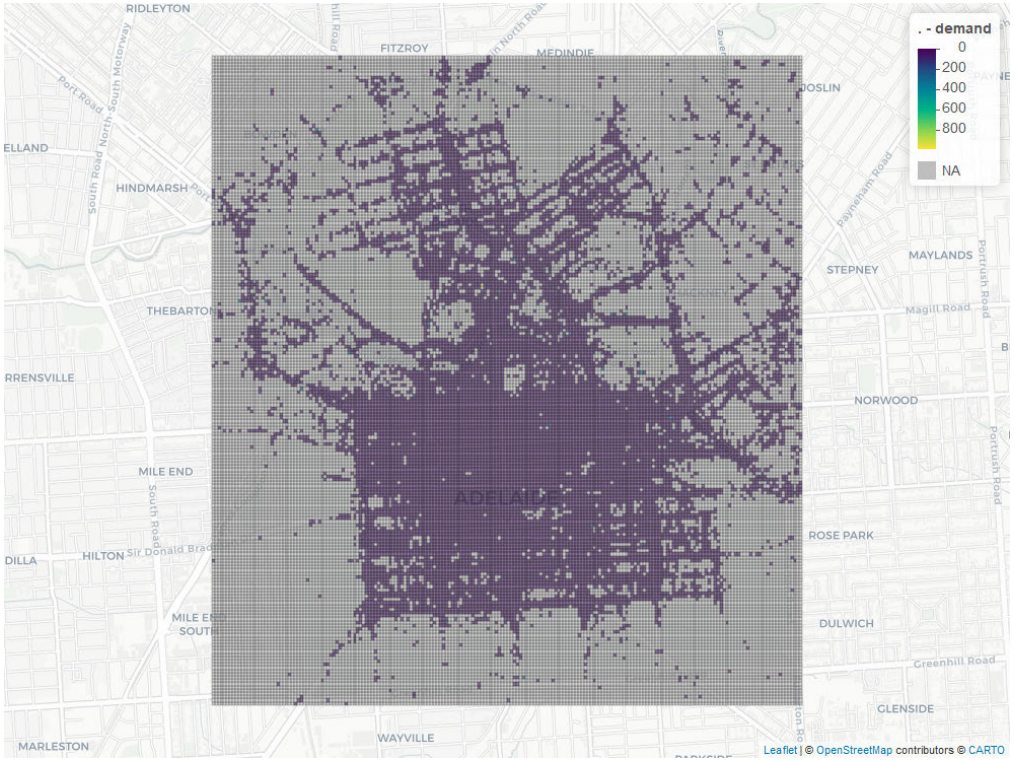


(d) Bayesian demand estimates, 50×50 grid.

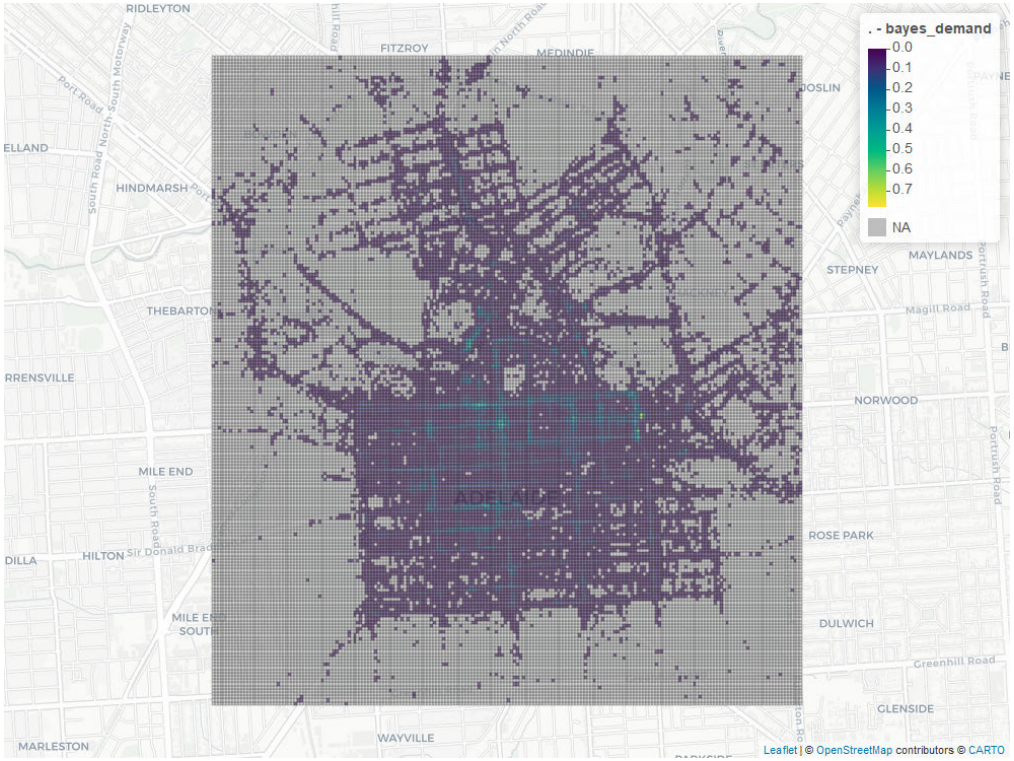
(e) Demand estimation with vehicle availability corrected estimate, 100 × 100 grid.



(f) Bayesian demand estimates, 100 × 100 grid.



(g) Demand estimation with vehicle availability corrected estimate, 200 × 200 grid.



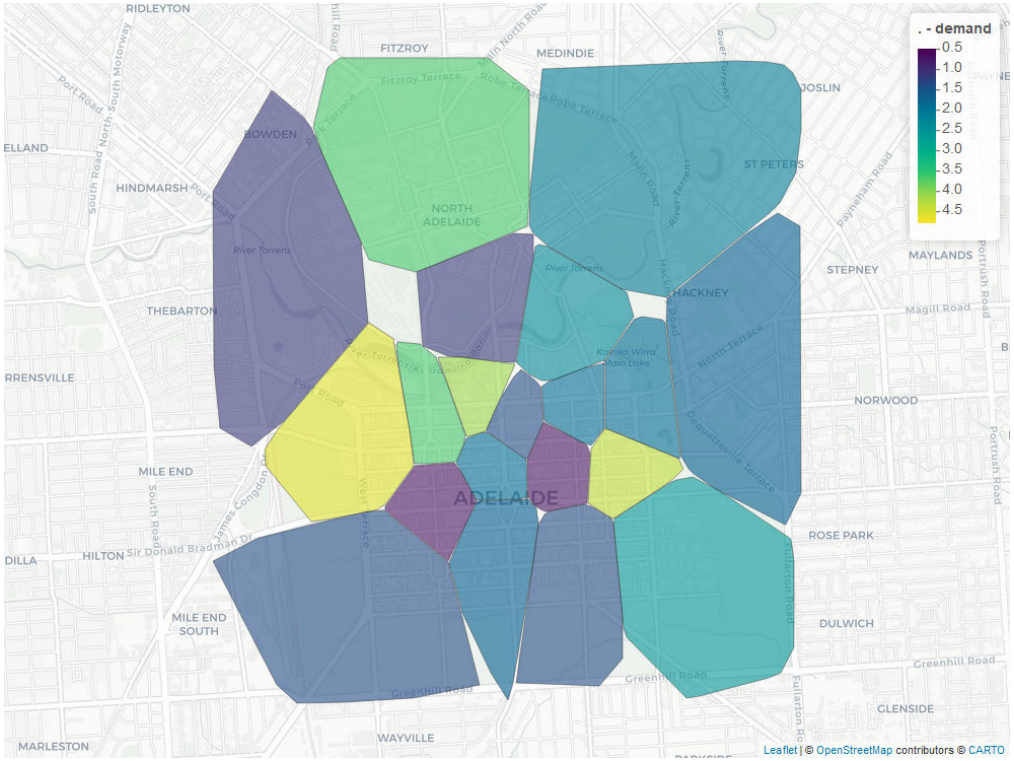
(h) Bayesian demand estimates, 200 × 200 grid.

Figure 3.4: Maps of estimated demand, in hires/hr, for various grids constructed across the Adelaide CBD. In the case of a 20 × 20 grid, both demand estimates appear reasonable, with similar patterns in both maps. However, for smaller grids the MLE demand estimates appear to not be accurate, as areas towards the edges of the service area have heavily inflated demand estimates, which are not shown in the Bayesian demand estimates.

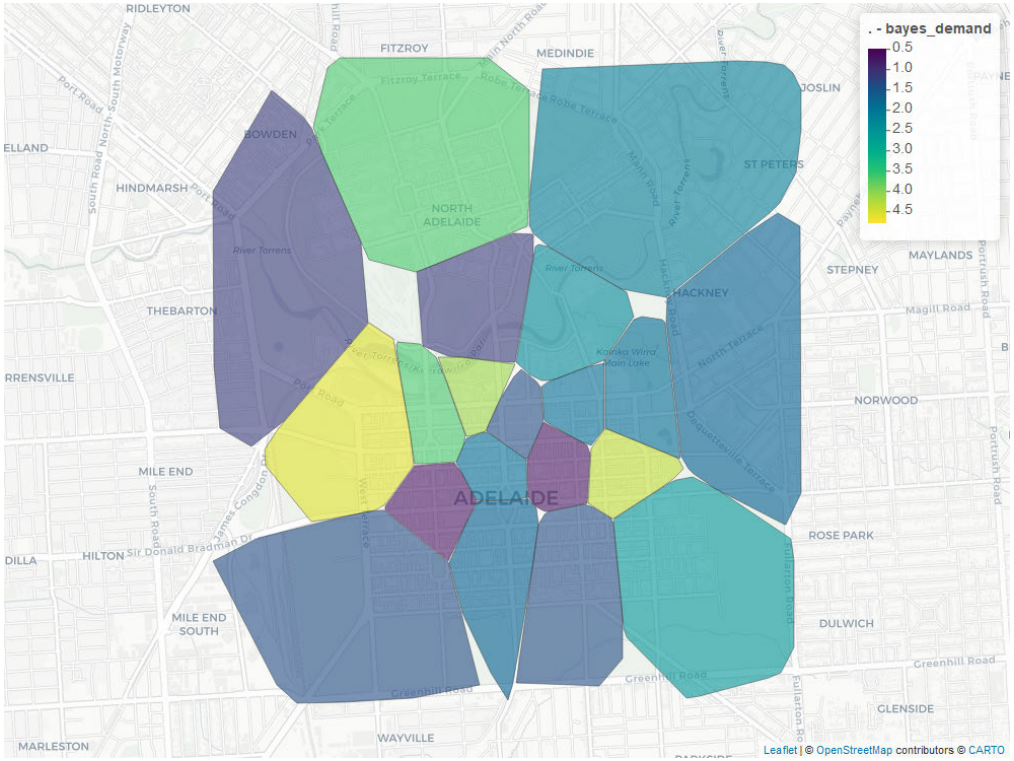
K-means Clustering

In addition to the grid-based binning, we also implemented k-means clustering as a binning method. To perform k-means clustering, we utilised the `kmeans` function from the *stats* R package (R Core Team and contributors worldwide (2021)). This function takes the latitude and longitude of all events in the status data set as inputs, and returns the cluster indices after at most 100 iterations of the k-means algorithm.

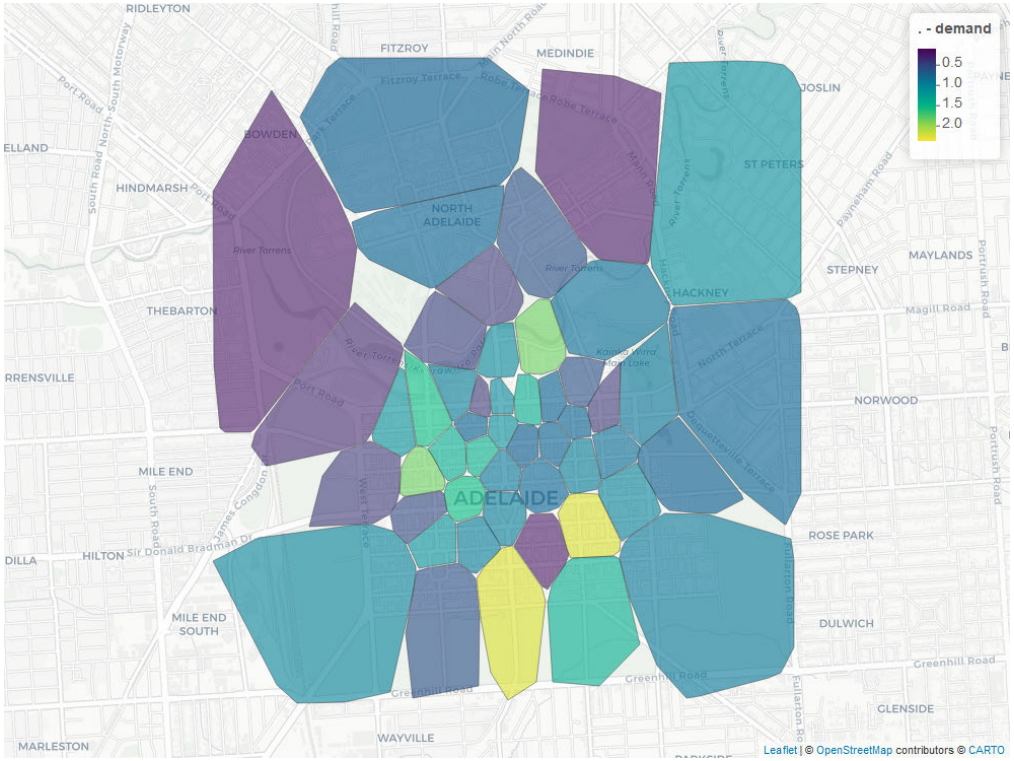
We tested the k-means clustering algorithm for binning the status data for various numbers of clusters from 20 to 200, to determine an appropriate number of cells. Figure 3.5 shows the demand estimates for these cases. Each of the shaded areas shown is the convex hull of all points within the cluster. Any areas outside of these cells are unshaded as there is no data present.



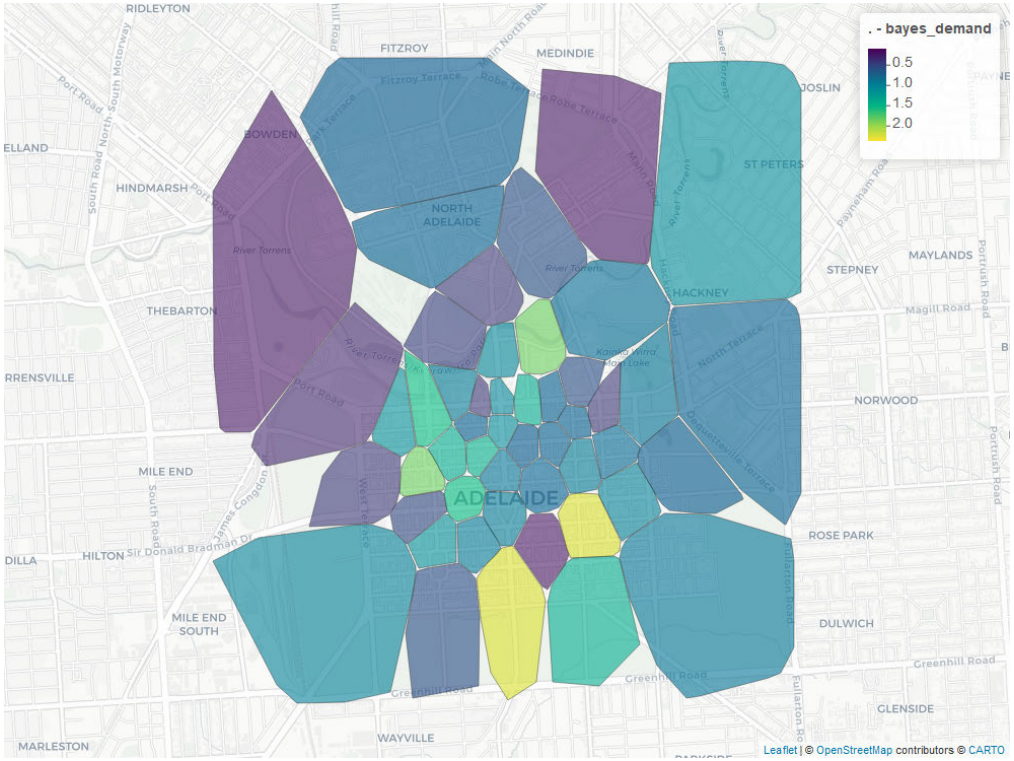
(a) Demand estimation with vehicle availability corrected estimate, k-means clustering with 20 cells.



(b) Bayesian demand estimates, k-means clustering with 20 cells.

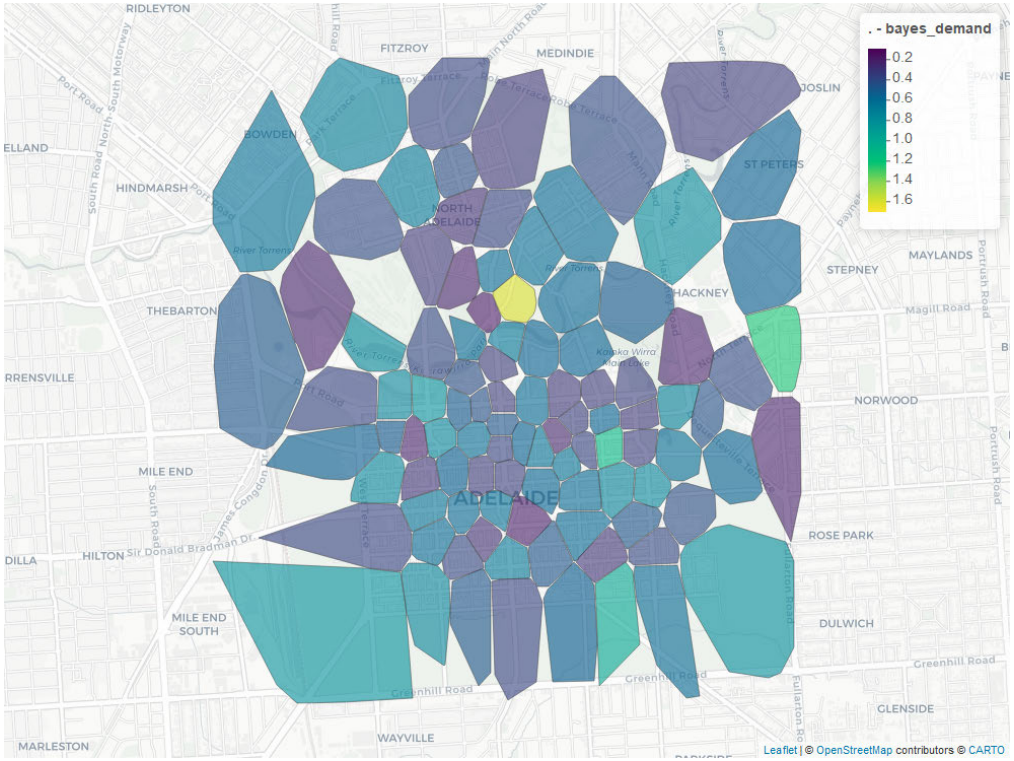


(c) Demand estimation with vehicle availability corrected estimate, k-means clustering with 20 cells.



(d) Bayesian demand estimates, k-means clustering with 20 cells.

(e) Demand estimation with vehicle availability corrected estimate, k-means clustering with 20 cells.



(f) Bayesian demand estimates, k-means clustering with 20 cells.

(g) Demand estimation with vehicle availability corrected estimate, k-means clustering with 20 cells.

(h) Bayesian demand estimates, k-means clustering with 20 cells.

Figure 3.5: Demand estimates, in hires/hr, using k-means clustering on the status data set for various numbers of cells.

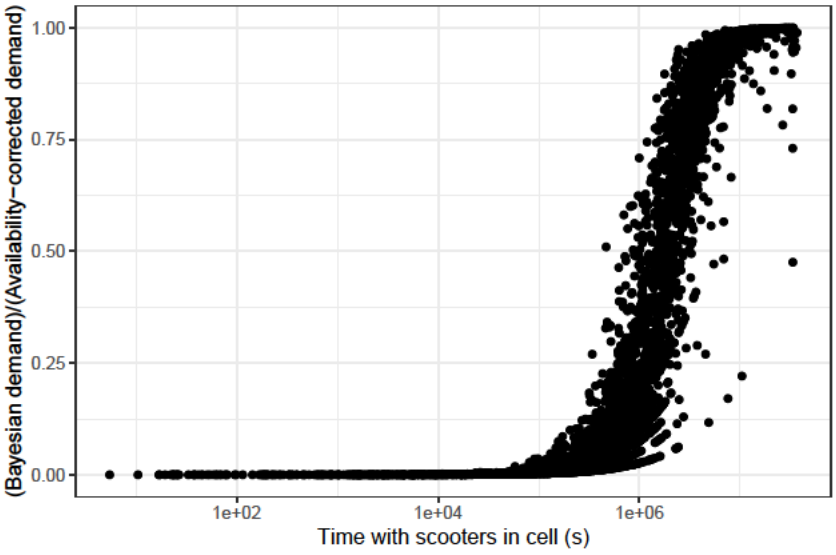


Figure 3.6: Plot showing the relationship between the ratio of the Bayesian estimates and MLE estimates and the time with scooters present in each cell, for a 100×100 grid.

The demand estimates resulting from k-means clustering show some interesting properties when compared to the grid-based binning method. First, due to the non-uniform cell size, the demand patterns across the service area change between numbers of clusters.

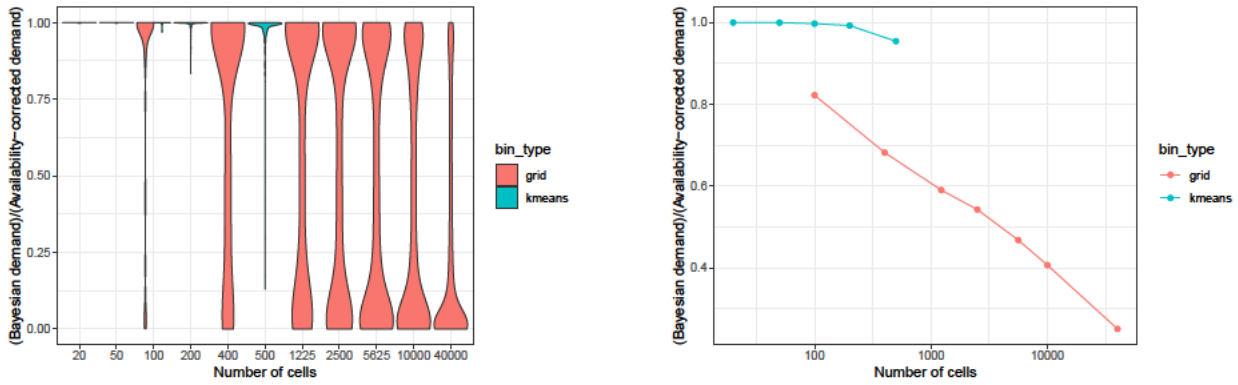
We also see that there is not a significant difference between the Bayesian and vehicle availability-corrected demand estimates, even with 200 cells. This is likely due to the k-means clustering approach resulting in all cells containing a relatively large number of events; this reduces the chance that any cell only has scooters for a short period of time, which causes the difference between the demand estimates.

Comparison Between Binning Methods

As shown above, there is a significant difference between the Bayesian and MLE demand estimates when using the grid binning method, that is not present in the k-means clustering method. To investigate this behaviour more closely, we consider the ratio between the Bayesian estimate and the MLE estimate. As the Bayesian estimate is at most equal to the MLE estimate, this value is between 0 and 1.

First, we consider the behaviour for a single case, a 100×100 grid. As the primary difference between the Bayesian and MLE estimates is the time that scooters are present, we plot the ratio against this time in Figure 3.6. As shown, there is a strong correlation between the ratio and the time with scooters in each cell. As the time increases, the Bayesian estimate approaches the MLE estimate.

We also consider the relationship between the ratio of demand estimates and the



(a) Distribution of the ratio of the Bayesian demand estimate and the MLE demand estimate, for each bin case.

(b) Average ratio of the Bayesian demand estimate and the MLE demand estimate, over all cells in each bin case.

Figure 3.7: Change in the ratio between Bayesian and vehicle-corrected demand estimates, $\hat{\lambda}_b/\hat{\lambda}_v$, against the number of cells in the binning method. A higher value indicates that the Bayesian estimates are close to the vehicle-corrected estimates.

different binning methods and numbers of cells. Figure 3.7 shows both the distribution of this ratio within each binning method, and the average across all cells in each binning method.

Figure 3.7a shows that the two binning methods exhibit significantly different patterns. With the k-means binning methods, almost all cells have a Bayesian demand estimate that matches the MLE demand estimate. In contrast, the grid method shows bimodality in all cell sizes, where the ratio is approximately 0 or 1. The additional points close to 0, corresponding to Bayesian estimates that are much lower than the MLE estimates, are likely due to outlying cells in the grid that do not contain scooters for a long period of time. This effect is increased as the number of cells increases, as the number of cells in areas without scooters increases. K-means binning does not have this issue as all cells contain sufficient data for the Bayesian estimates to match the MLE estimates.

A similar trend is shown in Figure 3.7b, when plotting the mean ratio between Bayesian and MLE estimates across all cells, in each binning method. The k-means binning method results in a significantly higher mean, indicating that the difference between Bayesian and MLE estimates is decreased compared to grid binning.

As shown, increasing the number of cells in grid-based binning has the effect of introducing significant errors in the MLE demand estimates, and is present to a lesser extent in k-means clustering. Due to this, having larger bins may result in more accurate demand estimates. However, this leads to lower resolution of the demand estimates across the service area. We choose a 100×100 grid to balance these two factors. In Section 3.6 we investigate the MLE and Bayesian demand estimates in detail for this grid.

3.6 Results

In the previous sections, we derived and implemented three demand estimation techniques and compared them for a range of different binning methods. Due to the limitations in the maximum likelihood estimation method identified previously, we use the Bayesian framework for all demand estimates. In this section, we investigate the demand patterns using the Bayesian estimates, and a 100×100 grid.

3.6.1 Overall Demand Estimates

As before in Section 3.5.4, we first consider combining data in each cell across all time to produce demand estimates. In doing so, we make the assumption that demand in each cell is constant in time.

Figure 3.8 shows the Bayesian demand estimates across the service area. We also present some demand estimates in Table 3.2, which shows the top 10 locations by estimated Bayesian demand. Each of these locations is also shown on a map in Figure 3.9.

Cell	Naive (MLE, $\hat{\lambda}_n$)	Vehicle-Corrected (MLE, $\hat{\lambda}_v$)	Bayesian (MAPE, $\hat{\lambda}_b$)	Hires
4943	0.696	0.826	0.826	7119
4546	0.679	0.757	0.757	6940
6144	0.673	0.736	0.736	6876
4843	0.468	0.615	0.615	4780
7244	0.394	0.528	0.528	4024
4355	0.092	0.479	0.476	944
6143	0.430	0.467	0.467	4396
4942	0.332	0.419	0.419	3390
7242	0.029	0.437	0.416	297
7241	0.052	0.422	0.416	534

Table 3.2: List of the top 10 cells in a 100×100 grid, sorted by the Bayesian demand estimate using the Beam dataset. All demands are measured in hires/hour.

The most in-demand cell by Bayesian demand estimate is cell 4943, which is located at the corner of King William Road and Rundle Mall. This is similar to the result in Section 3.5.1, which also showed that this location has the highest demand in the system. We also find that the second-most popular location, cell 4546, is located outside the Adelaide Railway Station on North Terrace, again matching the results using naïve demand estimation. The other cells listed follow similar trends to the naïve estimates, with areas such as the corner of Rundle Mall and Pultney Street (cells 6144 and 6143) and the corner of Rundle Street and East Terrace (cell 7241).

However, we also see multiple cells located in areas with significantly fewer hires. Cell 4355 is located outside Adelaide Oval, and has an estimated Bayesian demand of 0.476

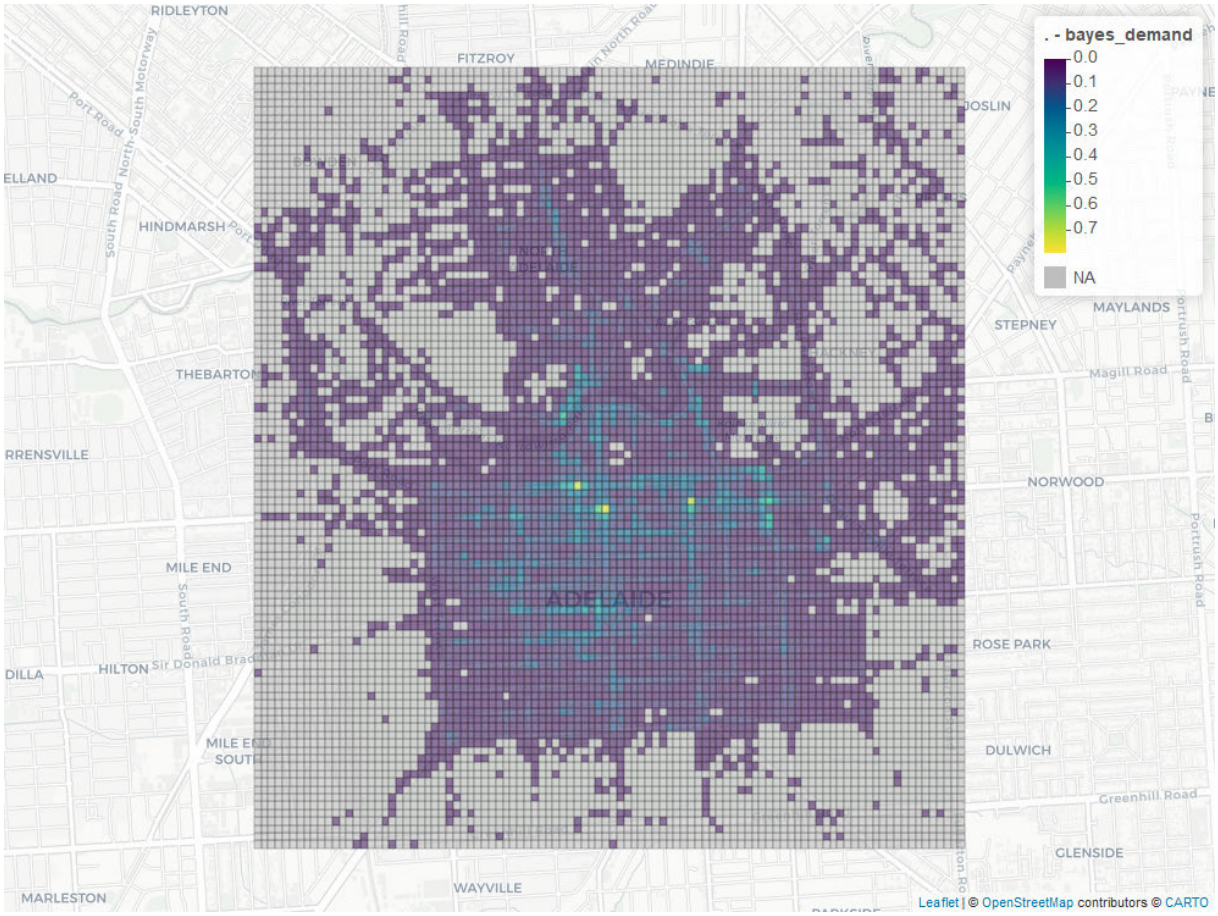


Figure 3.8: Bayesian demand estimates using a 100×100 grid.

Figure 3.9: Map of the 10 locations with the highest estimated demand in a 100×100 grid.

hires/hour compared to a naïve demand estimate of just 0.092 hires/hour. This is due to scooters being present in the area for only a short period of time. Similarly, cells 7241 and 7242, located on East Terrace between Rundle Street and Grenfell Street, have relatively few hires. The two cells have almost identical Bayesian demand estimates, however both the naïve and vehicle-corrected demand estimates differ. This shows that the vehicle-corrected demand estimate assists in preventing underestimating the demand due to hires not being possible, while the Bayesian estimate limits the amount of overestimation that can occur due to a short time with scooters present.

In cells where many hires occur, the vehicle-corrected demand matches the Bayesian demand. However, as the number of hires decreases the difference between the two demand estimates increases.

3.6.2 Time-Dependent Demand Estimates

To investigate how the system usage varies through time, we consider the overall system demand, defined by the sum of the estimated Bayesian demand over all cells in a 100×100 grid. This estimate depends on the binning process, despite being the total system demand, due to the estimates depending on the amount of time that scooters are present in each cell.

Demand Estimation by Month

The demand estimates presented so far in this chapter consider the demand across the entire time window of the data. However, as discussed in Section 2.4, there is a large amount of variation based on the time of day and day of the week, as well as significantly different system usage by month. Hence, we can improve the accuracy of demand estimates by splitting the data by time.

First, we consider splitting time by month. In the implementation of the vehicle-corrected and Bayesian demand estimates, while iterating through the data we define breaks at the start of each month. At the start of each month the number of scooters in each cell is calculated to ensure that the cumulative populations are accurate. By splitting time by month, we are able to estimate demand separately for each month. However, when calculating the Bayesian demand estimate, the time window is critical. When splitting the estimate by month, we restrict the time window to a single month. Hence, to obtain a demand estimate for a single month we replace the T term in Equation 3.3 with the length of the month, calculated as the difference between the month breaks, in seconds.

Figure 3.10 shows the total estimated demand across the service area, calculated as the sum of the estimated demand over all cells, for each month spanned by the data. Due to the demand estimate including the proportion of time that scooters are present in a cell, this total demand depends on the binning process, which in this case is a 100×100 grid. March 2021 contains the highest estimated demand with 211 hires/hour across the

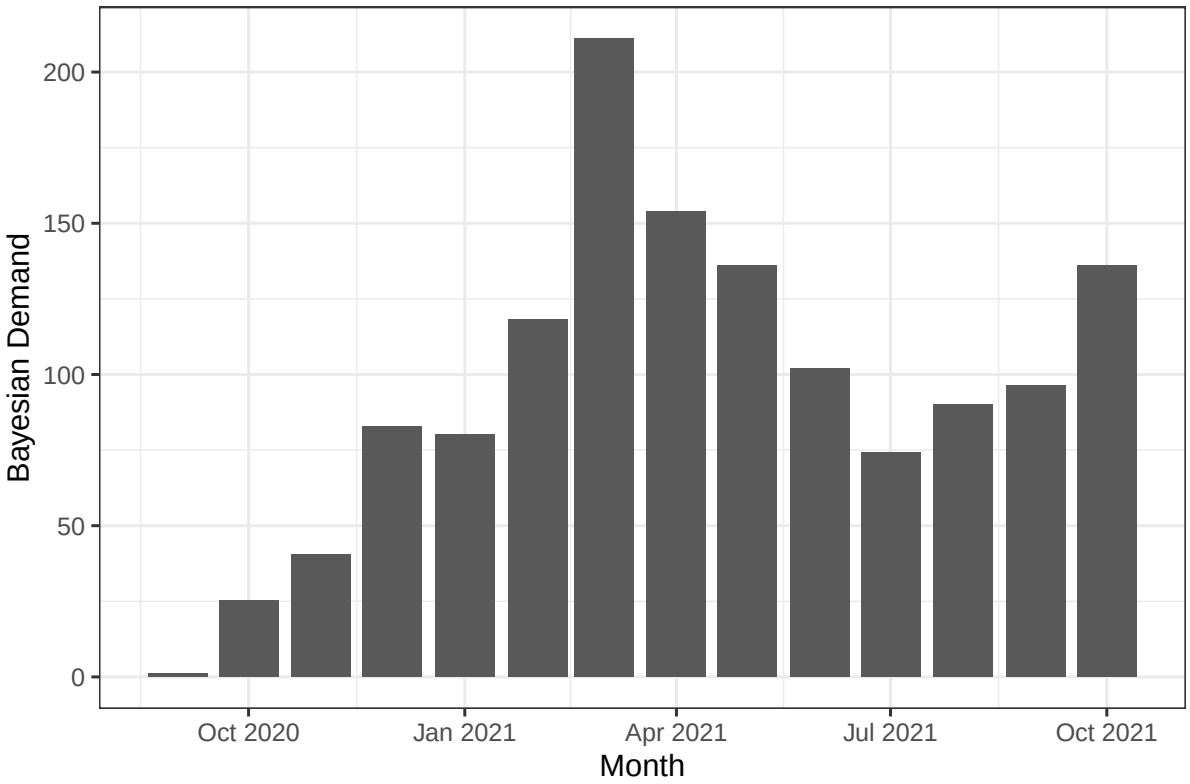


Figure 3.10: Estimated total system demand by month, using Bayesian estimation and a 100×100 grid.

system. We also see an increasing trend in the hire rate, with October 2020 recording a demand of 25 hires/hr compared to October 2021 with 136 hires/hr. This indicates that the overall system demand is increasing over time as more customers begin using the e-scooter ride sharing system.

Demand Estimation by Time

While splitting the demand estimates by month captures the long-term usage trends, there are also patterns in usage in the time of day and the day of the week. We consider splitting the demand estimates by both time of day and the day of the week to account for this variation. In particular, we consider combinations of the following categories of time.
Days of the week:

- Weekdays (Sunday 3pm-Friday 3pm), and
- Weekends (Friday 3pm-Sunday 3pm).

Times of day:

- Early morning (12am-6am),
- Morning (6am-12pm),
- Early afternoon (12pm-3pm),
- Afternoon (3pm-7pm), and
- Evening (7pm-12am).

These times were chosen to split the data by into periods of time when the rate of hire is approximately constant. This is demonstrated in Figure 3.11, which shows the frequency of hires by the day of the week and time of day. Friday Afternoon and Evening is included in the "Weekend" category, as the behaviour shown is more similar to the behaviour of the system on Saturdays. Similarly, Sunday Afternoon is counted as a "Weekday" due to the similarity to Weekday Afternoons.

The first component of time-dependent demand variation we investigate is the effect of Weekends. From the analysis performed earlier, we expect the system demand during Weekends to be significantly greater than the system demand on Weekdays. Similar to the monthly estimates, we modify T in the Bayesian demand estimates to be the total time in each time category for the overall time window of the data. Figure 3.12 shows the two system demand estimates. The Weekend demand is estimated to be 209 hires/hr compared to 114 hires/hr for Weekdays. This indicates that there is a significant difference in e-scooter usage between Weekdays and Weekends.

We also investigate how the demand varies due to the time of day, with the 5 categories of time of day defined above. Here we again consider the overall system demand using Bayesian demand estimates on a 100×100 grid. Demand variation due to the month or day of the week are not included in this model.

Figure 3.13 shows these demand estimates. The demand is highest during the Afternoon and Evening periods, which take place between 3pm-7pm and 7pm-12am respectively. During this time, the overall system demand is estimated to be 200-214 hires per hour. The demand during Early Morning and Mornings is significantly lower with 45 and 71 hires per hour respectively. This agrees with the hire frequency plotted in Figure 3.11, which shows fewer trips taken during the early hours of the morning compared to the high amount of system usage in the afternoon and evening.

We can now combine the three approaches to time splitting to obtain demand estimates for e-scooters for all combinations of times of day, weekends and weekdays, and months. This allows us to investigate the effects of time of day and day of the week together, and how these effects change as long-term usage patterns evolve in the system. Figure 3.14 shows these demand estimates. The 10 individual plots correspond to the 10 combinations of time and day, and the demand estimates are plotted against time in months.

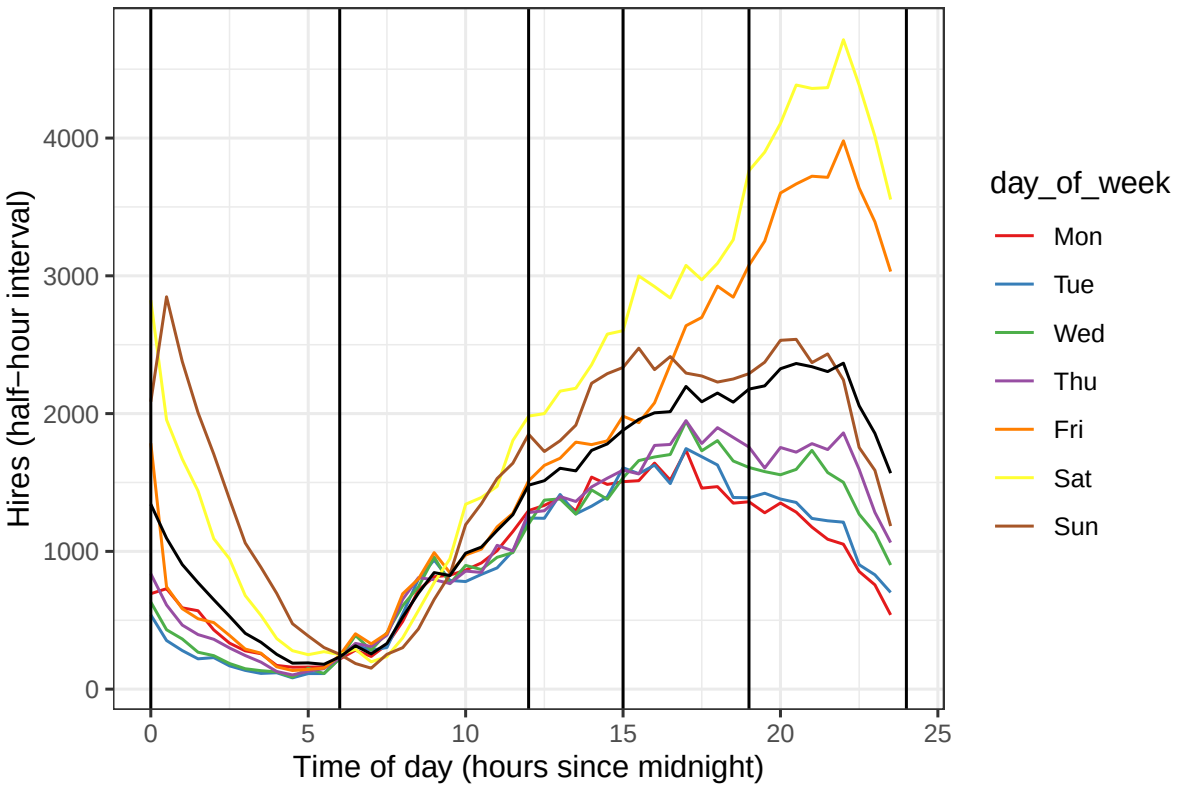


Figure 3.11: Frequency of vehicle hires by time of day and day of week across the service area, showing how time is split for time-based demand estimation.

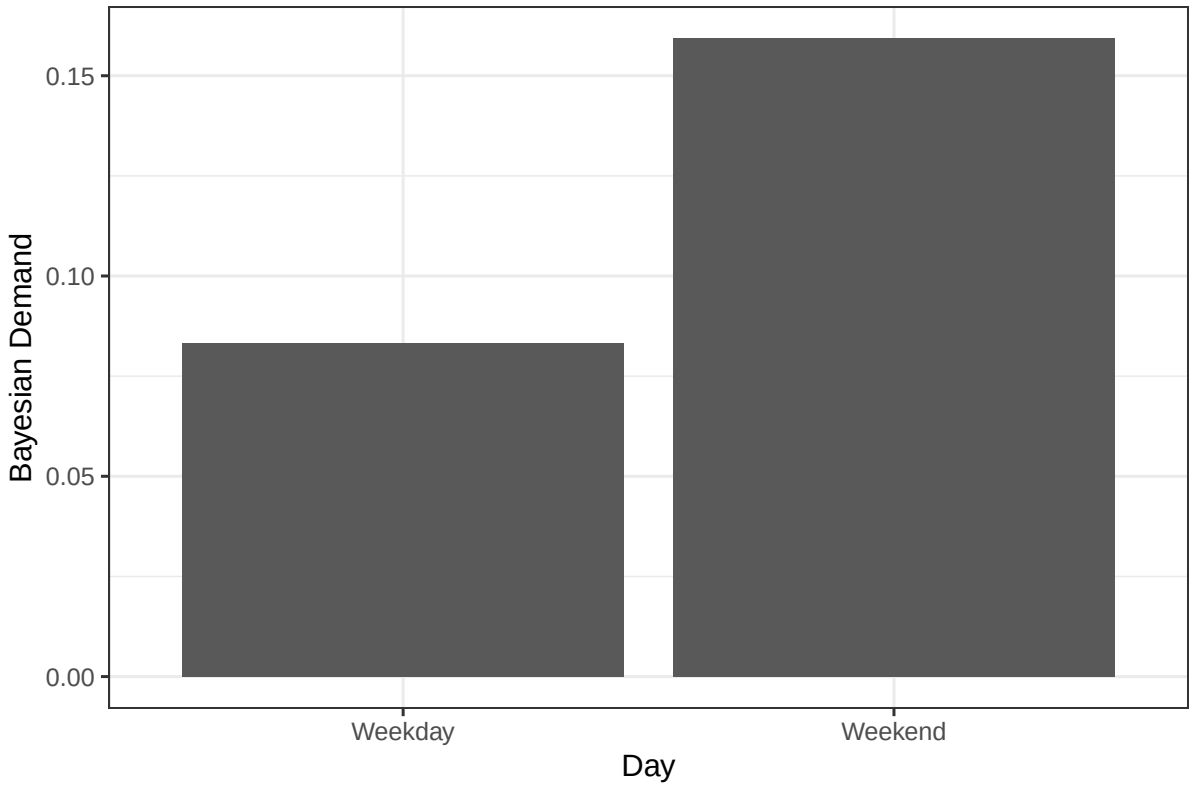


Figure 3.12: Overall system demand in hires/hr, split by weekend vs weekday. The estimates shown are the sum of demands across all cells in a 100×100 grid, using Bayesian demand estimates.

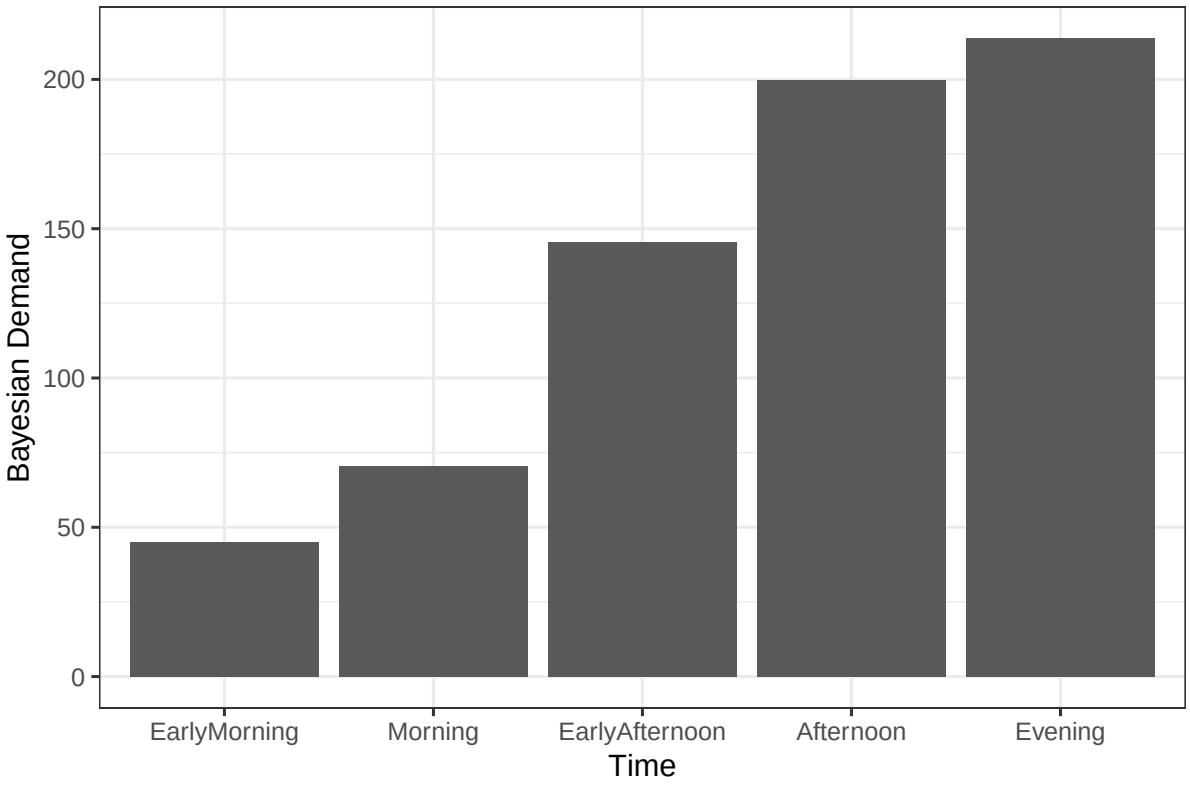


Figure 3.13: Overall system demand in hires/hr, split by time of day. The demand is highest in the afternoon and evening, with much lower usage in the early morning and morning.

In general, the patterns in estimated system demand is similar between weekday and weekend for each time of day. The demand appears to match the monthly overall demand, scaled to different amounts corresponding to each time of day. However, there are several interesting patterns in the demand estimates.

The most apparent behaviour of the demand estimates is the peak in demand in all time categories in February and March 2021. While this was shown earlier in Figure 3.10, by differentiating by time of day we see that a large proportion of the increased demand occurs in the Afternoon and Evening, especially on Weekends.

We also see that in almost all times of day, weekend demand is higher than weekday demand, which matches the earlier results. However, in the morning time category the weekend demand is lower than the weekday demand with an average of 59.5 hires/hr compared to 64.3 hires/hr on weekdays. This indicates that a large proportion of e-scooter customers use the system to travel during Weekday Mornings, which may be due to commuting to work.

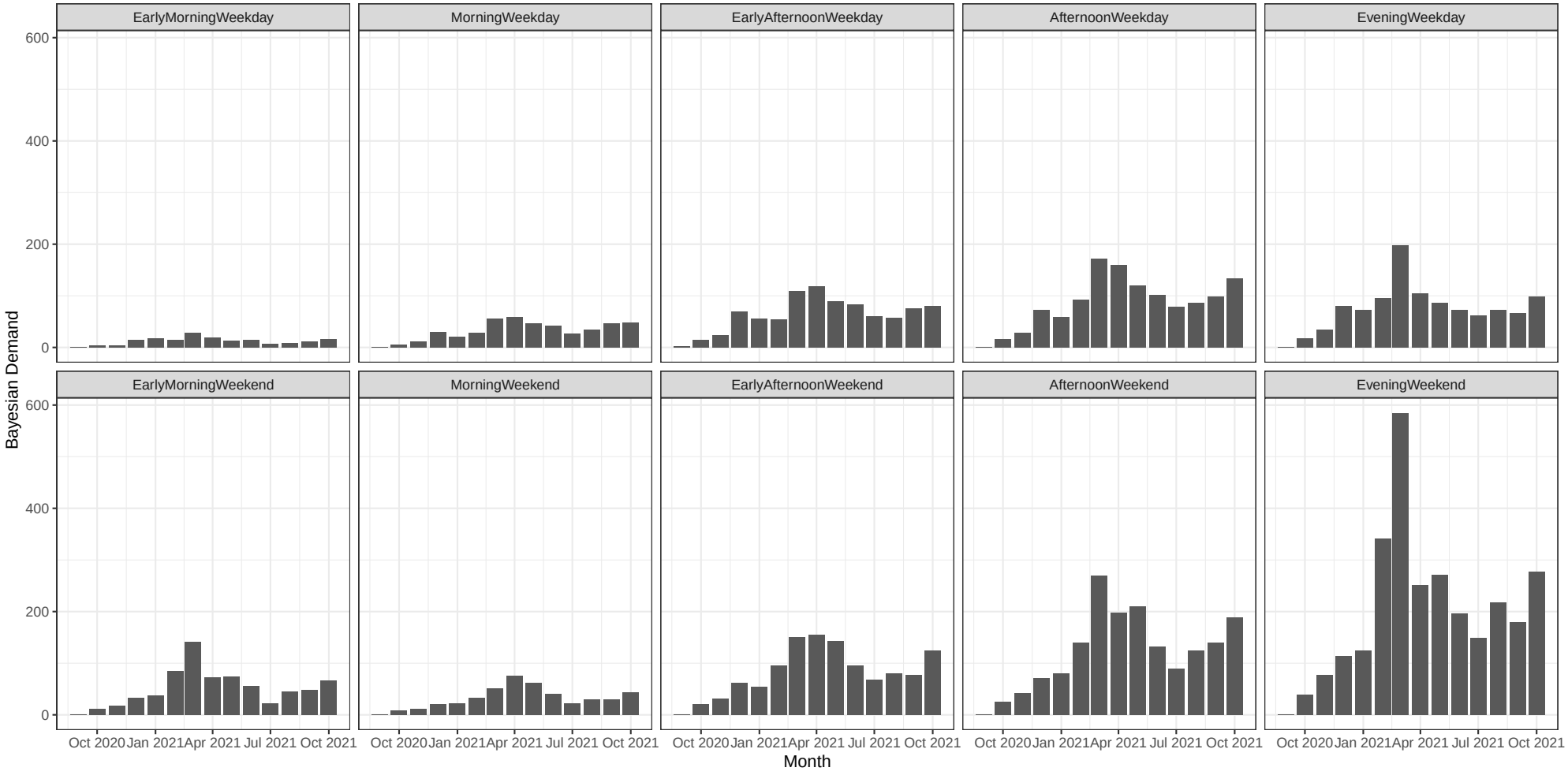


Figure 3.14: Overall system demand, in hires/hr, split by time. The estimates shown are the sum of demands across all cells in a 100×100 grid, using Bayesian demand estimates. The top row shows the demand estimates for weekdays, for each category of time. The bottom row shows the estimates for weekends.

3.7 Discussion

In the previous sections, we derive and implement e-scooter demand estimates across the service area, discretised in both space and time. However, as hires can occur anywhere in the service area and at any time, discretisation removes some detail in the estimates. Hence, one major direction for future research is to implement a continuous demand distribution, in time and/or space. This may allow for significantly more accurate demand mapping, with improved detail in areas with high demand. This approach does have some issues however, such as increased difficulty in implementing a simulation model for vehicles travelling in a continuous space.

Another possibility for future development is to incorporate spatial correlations in the demand estimates, for example using some of the data from a cell to influence the demand estimate in a neighbouring cell. We did not implement this in the demand estimation above as the cell sizes were chosen to be large enough that the actual demand is likely to vary between cells, so spatial correlations are less impactful. Possible methods to do this include applying a smoothing algorithm over the grid, applying a KDE model (Pinna et al. (2022)), or introducing a coupling structure to induce correlations between cell demand estimates.

Another issue with the demand estimation process occured in the implementation of the vehicle population tracking algorithm described in Section 3.3. The algorithm keeps track of the number of scooters in each cell in the system, at all times that occur in the status data set. Clearly, the populations should be non-negative for all locations, for all times. However, in our implementation the population count becomes negative for some cells for a period of time. Figure 3.15 shows the vehicle populations in each cell for each iteration. There are several possible causes for the negative counts. As described in Section 2.6, there are a large number of errors in the data, such as missing and duplicated events. This would cause the population counts to change incorrectly; for example, if a scooter is picked up from the same location twice without being dropped off the population count would decrease by 2, rather than 1. However, we believe all such data errors were accounted for in the redistribution estimation section. Hence, it is likely that a bug exists in the implementation of the vehicle tracking algorithm. The effect of the negative scooter counts is limited, however. In the demand estimates, any non-positive value is treated as having no scooters in the cell. The negative counts also only occur for a short period of time, further limiting the potential impact of the error on the final demand estimates.

In the demand estimation process, we split the demand estimates in each cell by time of day. We use the overall system demand to inform the time categories, as shown in Figure 3.11. We classify Sunday Afternoons and Sunday Evenings as weekdays due to the similarity between these two time categories and the system demand on other weekdays. However, it is likely that the spatial patterns in demand on Sundays are different to other weekdays. Due to this, it may be more accurate to define new categories for Sunday afternoons. However, due to the limited amount of data available the estimates for these

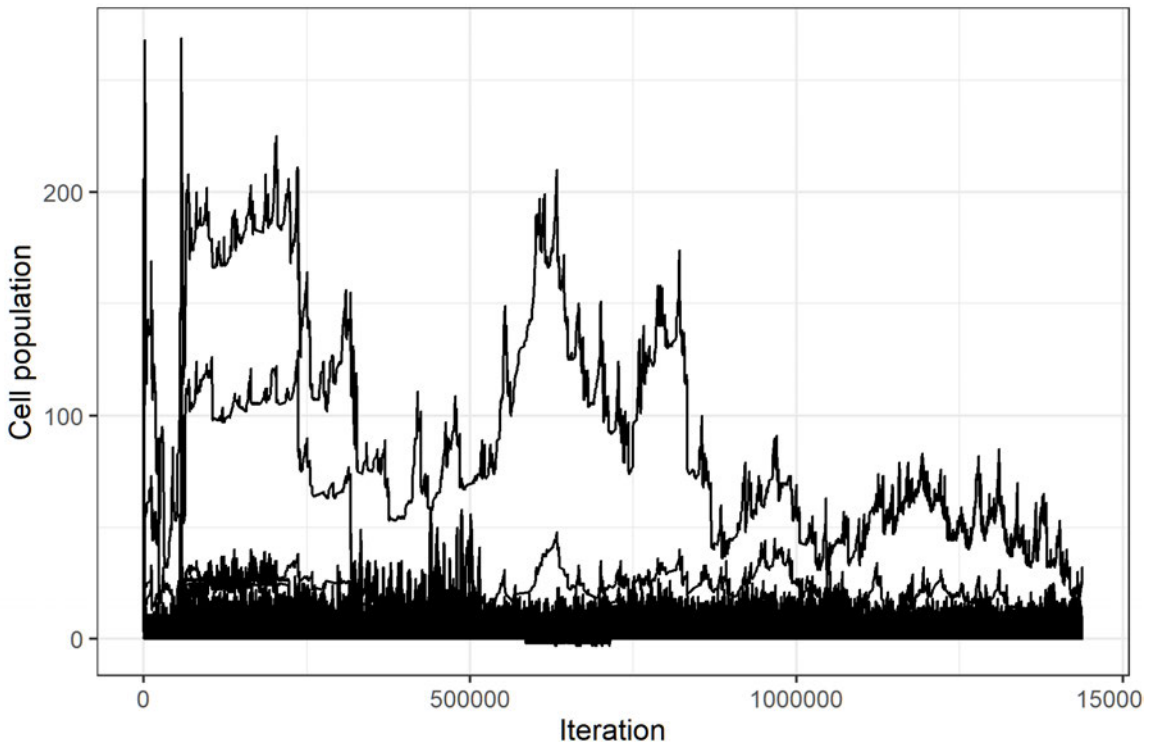


Figure 3.15: Cell populations estimated from the vehicle tracking algorithm.

time categories may be unreliable.

In this chapter, we derived and implemented estimates for the demand for e-scooters across the Adelaide CBD. The demand has been estimated for discrete, fixed locations arranged in a grid across the service area, and for time categories spanning months, the day of the week and the time of day to account for time-based variation in the demand. These estimates form the core component of the simulation models in Chapter 4, which then inform the deployment optimisation performed in Chapter 5.

Chapter 4

Simulation of E-Scooters in Adelaide

4.1 Introduction

In this chapter, we describe and implement a model to simulate the movement of e-scooters across the e-scooter service areas in the Adelaide CBD.

There are several components to construct before developing the simulation model. The first is simulating the hire process in each cell, which directly builds upon the demand estimation performed in Chapter 3. We also consider simulating the destinations of trips given the starting locations, and the duration of a trip between the two endpoints. In Section 4.3 we describe the overall simulation model, as well as an extension of the model that we then use in the optimisation process in Chapter 5. In Section 4.4 we present results from the simulation model and analyse behaviours present.

4.2 Components of Simulation Model

4.2.1 Hire Simulation

The first component in the simulation model is the simulation of hire processes based on the demand estimates from Chapter 3. Following the assumptions made earlier, we assume that hires follow independent Exponential distributions in each cell, with rate parameters varying between each cell and by discrete categories in time. The hire process does not depend on the number of scooters in the cell. Instead, after generating a hire, we consider the hire *successful* if a scooter is available, and *unsuccessful* if a scooter is not available at the time of the hire. The time variation introduces some complexity to the hire time distribution, as the rate of hires changes at known, fixed times.

We use a recursive approach to simulate a hire time that occurs past a change in demand. Consider times when the demand changes $t_1, t_2, t_3, ..., t_k$, corresponding demands $\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{k-1}$ where λ_j denotes the demand between t_j and t_{j+1}, and some time

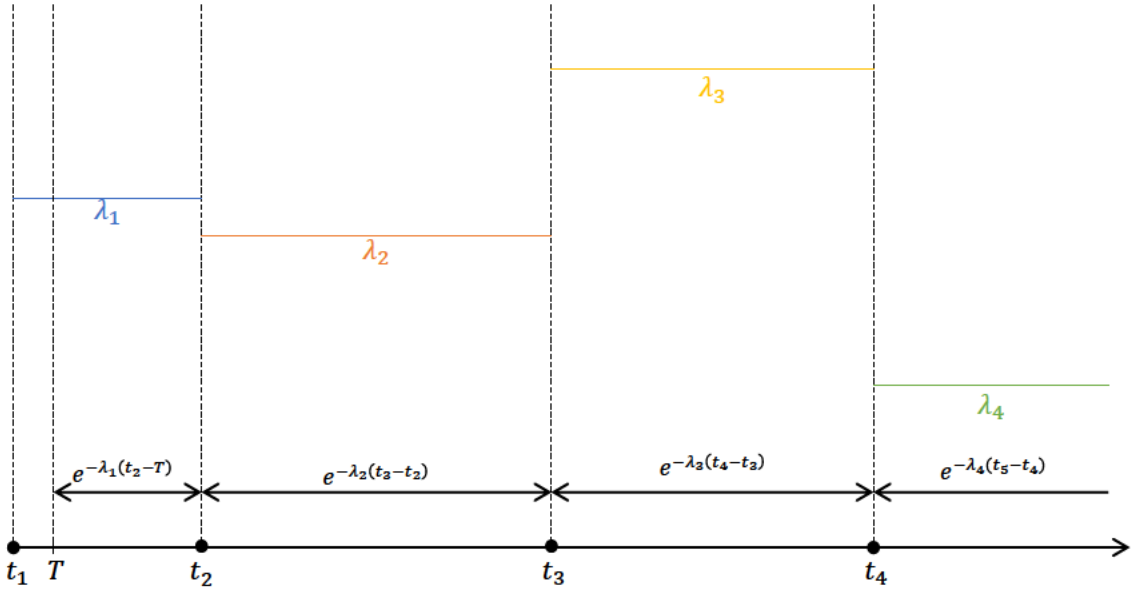


Figure 4.1: Diagram showing the generation of a hire time S from the current time T. The rate λ changes at times t_1, t_2, t_3, and t_4, and the time until the hire is Exponentially distributed with those rates. The probability of the hire occuring in an interval, given it is at least the lower value of the interval, is shown.

$T \in (t_1, t_2)$. Let the hire time be $S > T$. If $S < t_2$, then the hire time S follows a truncated Exponential distribution with rate λ_1 and maximum value $t_2 - T$. However, with probability $\exp(-\lambda_1(t_2 - T))$ we have $S > t_2$. In this case, the local demand is λ_2 and the hire time is then also distributed by an Exponential distribution with rate λ_2 and maximum value $t_3 - t_2$. This process repeats until the hire time does not exceed a hire boundary. This process is shown in Figure 4.1.

If the demand estimate in the cell is zero, then the hire time is set to the far future, i.e. $T > t_{max}$, so that no hires occur in the cell within the time window of the simulation.

4.2.2 Trip Destination Simulation

The next component in simulating an e-scooter ride-sharing system is the destination chosen for a trip. We assume that the destination only depends on the origin cell, and does not change through time. We denote the probability of a trip moving from cell i to

cell j as $p_{i,j}$, for all i, j. To estimate these probabilities, we take the proportion of trips that end in cell j from all the trips that start in cell i, i.e. the estimate, $\hat{p}_{i,j}$ is

$$\hat{p}_{i,j} = \frac{\text{number of trips from } i \text{ to } j}{\text{trips from } i \text{ to any cell}}.$$

For some source cells i, no trips occur. In these instances, we take $\hat{p}_{i,j} = 1/n^2$, where n^2 is the total number of cells, for all destination cells $j \in 0, 1, ..., n^2 - 1$; this forms a uniform distribution of destinations across the entire service area. Using these probabilities, we can simulate the destination cell for a trip by sampling from the distribution of destinations from the source cell.

4.2.3 Trip Duration Estimation

The final component of simulating an e-scooter ride-sharing system is the time needed to take a trip. While we have data regarding the lengths of all trips in the system, we use a grid-based system for simulation. This causes issues when attempting to simulate trips, as we do not know the exact distance travelled in a trip; rather, we know the two endpoint cells. To solve this, we measure the speed of scooters in the units of cells per second rather than metres per second. In doing so, we assume that travelling across a cell in the north-south direction takes the same amount of time as travelling in an east-west direction, which may not be accurate due to the cells being approximately a 50m×70m rectangle, where the cell sizes vary slightly by latitude. The speed of traversing a cell may also depend on the location of the cell, as travelling through a busy intersection in the centre of Adelaide is likely to take longer than a path in a quieter section of the service area. However, we assume that this effect is minor, and taking the average speed over all trips provides a sufficient measurement of the speed of scooters.

To measure the distance travelled in a trip in the grid framework, we consider the Manhattan distance of the trip. Compared to a straight line approximation of distance between the two endpoints, the Manhattan distance estimate better approximates the grid layout of the Adelaide CBD. The Manhattan distance, $d_{i,j}$, is defined for a pair of points (x_i, y_i) and (x_j, y_j),

$$d_{i,j} = |x_i - x_j| + |y_i - y_j|,$$

i.e. the distance in the x-axis added to the distance in the y-axis (Dong et al. (2018)). In terms of cells, the coordinates are the indices of cells in each axis. For example, cell 1732 has latitude index 33 and longitude index 18, so the coordinates are (18, 33). We calculate the Manhattan distance in cells for all trips that occur in the Beam Trip Data. To check that this metric is a reasonable estimate of the distance travelled in trips, we plot the estimated Manhattan distance against the distance recorded by the vehicle in Figure 4.2. If the estimates are accurate, then we expect the Manhattan distance to be

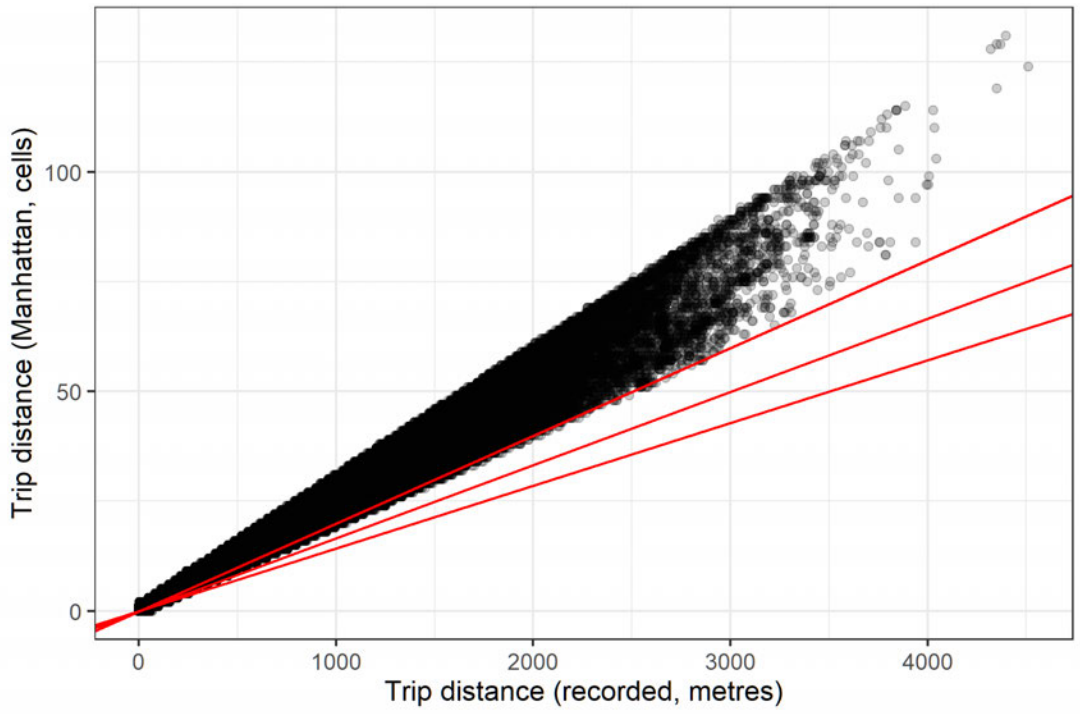


Figure 4.2: Comparison of trip distance recorded in data to the estimated Manhattan distance using binned location data. Also shown are lines with slope $1/50, 1/60$, and $1/70$, corresponding to the expected relationship.

linearly proportional to the trip distance, with a scaling factor of approximately $1/50$ to $1/70$ depending on the direction of the trip. However, we see that the estimated distances are higher than this in general. This indicates that trips that take place in the system do not occur entirely in cardinal directions, with some segments taking place on diagonals. Another possible cause for this difference is e-scooter users not taking the shortest possible path; they may take a slightly longer path depending on the topography of the area. However, as the relationship appears to be linear the Manhattan estimate is sufficient for estimating the speed of scooters.

Using these distance measurements, we can calculate the average speed of scooters during trips in the Beam system. Figure 4.3 shows the distribution of these speeds, measured in cells/second. The distribution is bimodal, with a large peak at 0 and 0.04. The mean of the speeds is 0.0367 cells/second. Converting to kilometres per hour by multiplying by 60m, the approximate width of a cell, we find that the average speed is 6.5km/hr. This is reasonable, since the maximum speed reported by vehicles in the Beam system is 13.5km/hr.

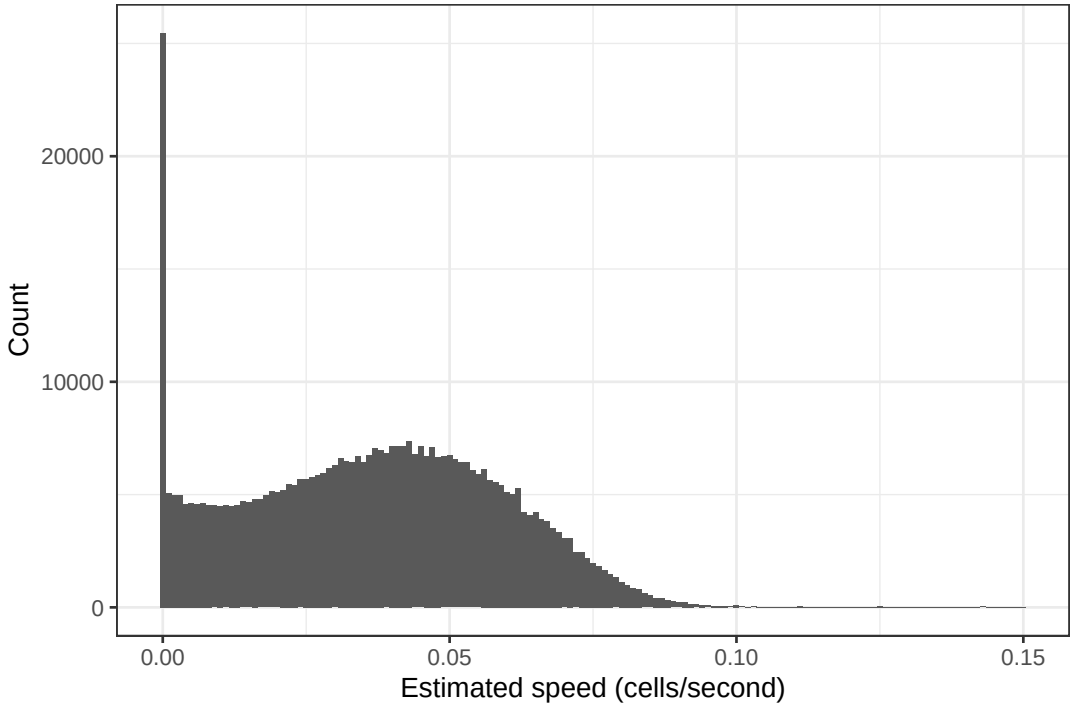


Figure 4.3: Distribution of the estimated speed of e-scooters during trips, measured in cells/second. Each cell is approximately 50m×70m, so the mean of 0.03 cells/second corresponds to approximately 6.5km/hr.

4.3 The E-Scooter Simulation Model

In this section, we describe the overall algorithm used to simulate the hires in an e-scooter ride-sharing system. We use the following notation:

- The current time in the simulation, t, from 0 to T.
- The maximum time to simulate to, $T > 0$.
- The next hire time in cell k, $h_k > 0$ for $k = 0, 1, ..., n^2 - 1$.
- The current population of scooters in cell k at time t, $N_k(t)$.

The process to simulate e-scooters is as follows:

1. Start: $t = 0$. Each cell k has a population of scooters $N_k(t) \in \mathbb{N}_0$.
2. Simulate initial hire times: Generate h_k, for all cells k, from time t.
3. Hire event: We set $t \leftarrow \min_{k=0,...,n^2-1}\{h_k\}$, the time of the first hire in the system. We denote the cell this hire occurs in u, so $h_u = t$.
4. Trip parameters: We first generate a destination cell v. From the pair u, v, we calculate the trip duration and hence arrival time s of a trip.
5. Scooter availability: If a scooter is not present in cell u, $N_u(t) = 0$, we consider the trip to be *unsuccessful*. We simulate a new hire time h_u starting from t, and go to step 3. If at least one scooter is present, $N_u(t) > 0$, the trip is *successful*. In this case, we set $N_u(t) \leftarrow N_u(t) - 1$, reducing the number of scooters in u by 1. We add the scooter to the destination cell v at the arrival time, incrementing $N_v(s)$ by 1. Finally, we simulate a new hire time h_u starting from t, and go to step 3.

This algorithm is repeated until $t > T$. We record all trips within this time, with the status of the trip (successful vs. unsuccessful), the time of the trip, the source and destination cells, and the duration of the trip.

A useful characteristic of this algorithm is that it can be separated into two phases – the *trip generation* stage, and the *scooter movement* phase. The first of these is the stochastic process of generating all trips that may occur in the system, including the origin cell and hire time, and destination cell and arrival time. Each of these trips is then defined to be either *successful* or *unsuccessful* depending on the second stage of the algorithm, where the location of all scooters in the system are tracked according to the trips generated in the first phase. This behaviour allows us to investigate both the trips that occur in the simulated system as well as the trips that fail due to a lack of available vehicles. We also exploit this characteristic in Chapter 5 to increase computational efficiency, as we can generate a large set of trips in advance then determine scooter movement given a proposed

initial deployment. Multiple metrics for system efficiency are then available, including the revenue generated and proportion of successful trips. In Section 4.4 we investigate the efficiency of the Beam system given a naïve initial deployment.

4.4 Results

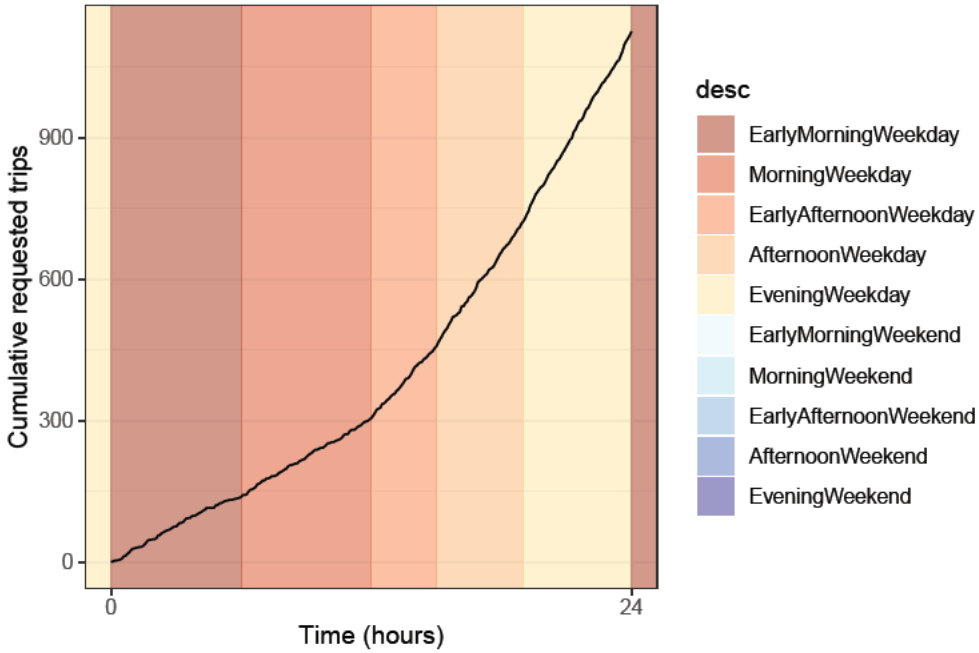
Prior to running any simulations, we require an initial deployment of scooters. As discussed in Section 2.4.2, there are 1415 unique scooters present in the Beam system over the entire time window of the data. Accordingly, we assume that there are 1400 scooters available in the simulated system. To distribute these, we sample with replacement from the list of all cells, weighted by the number of times a scooter is hired in the cell. This provides a deployment that is somewhat close to the expected optimal solution, as scooters are generally placed in areas where a large number of trips occur.

Another small modification to the simulation model is reporting the status of the system at regular time points, to allow for analysis of the evolution of the system over time. We save the cumulative number of successful trips, number of missed trips, the revenue, and the current proportion of scooters that are in use, calculated as the proportion of scooters that are inbound at the time of the report out of all scooters in the system.

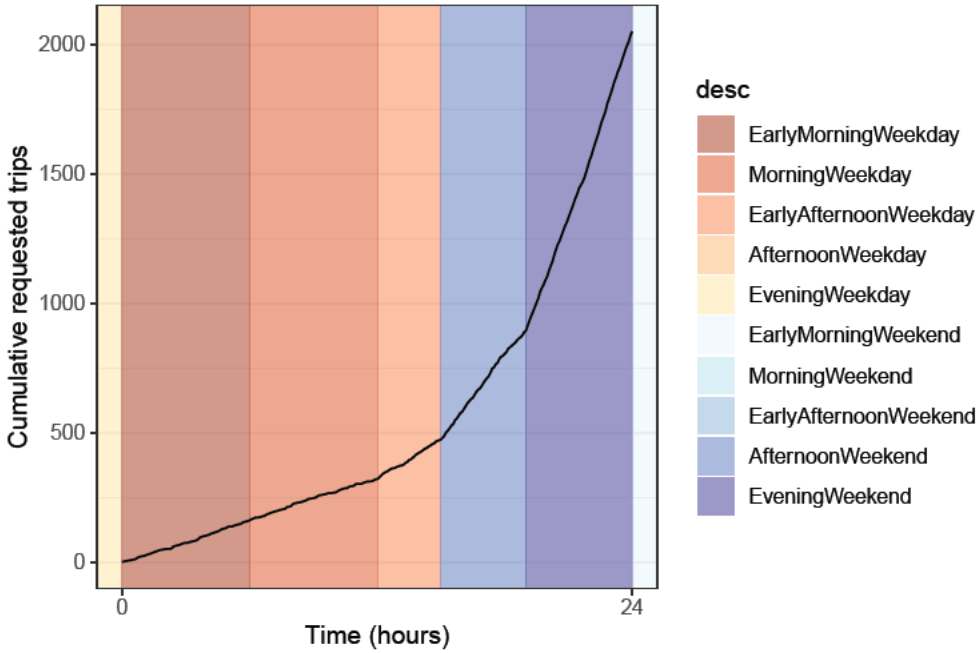
There are multiple parameters to choose for running the simulations. We assume that the revenue from a trip is equal to \$1 for the unlock fee and \$0.45 per minute in the trip, matching the fees Beam charged in April 2022 (Beam (2022*b*)). We take the time step for regular reporting to be 300 seconds, or 5 minutes, and the final time to be 24 hours after the initial time. The final parameters are the starting time and the month the simulation occurs in, which we vary to investigate the behaviour of the system.

We consider running simulations using the demand estimates from March 2021, the month with the highest estimated overall system demand. We consider how the behaviour of the simulated hires change through time. We first investigate the hire process, and the differences in simulated hires throughout the course of a week. To cover all categories of time, we run simulations starting at midnight for the four days of the week between Thursday and Sunday. Figure 4.4 shows the total number of requested hires – the number of hires that are either successful or unsuccessful – for each day. The simulation for each day is presented in an individual panel, and the time categories are shaded. We see that in all simulations, the number of hires is highest during the afternoon and evening time categories towards the right of the plots, caused by the higher system demand during these times. We also see that the demand during weekdays is lower than the demand during weekends, with 1119 trips generated on a Thursday, shown in Figure 4.4a, compared to just over 2311 trips on a Weekend day as shown in Figure 4.4c. However, the total number of requested trips on Friday, shown in Figure 4.4b is close to the number of trips on Saturday, with 2049 requests, indicating that the lower early morning weekday demand is compensated by a higher morning and early afternoon demand compared to the demand

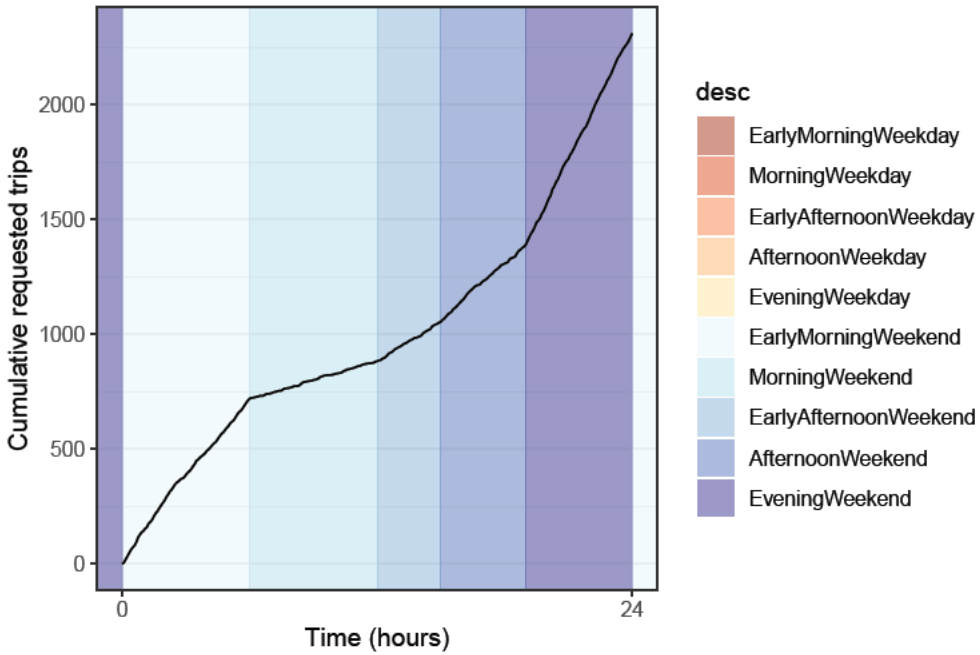
on weekends. This is also shown by the steady behaviour of the hires on weekdays in the early morning and morning, whereas on weekends the demand decreases in the morning and early afternoon times.



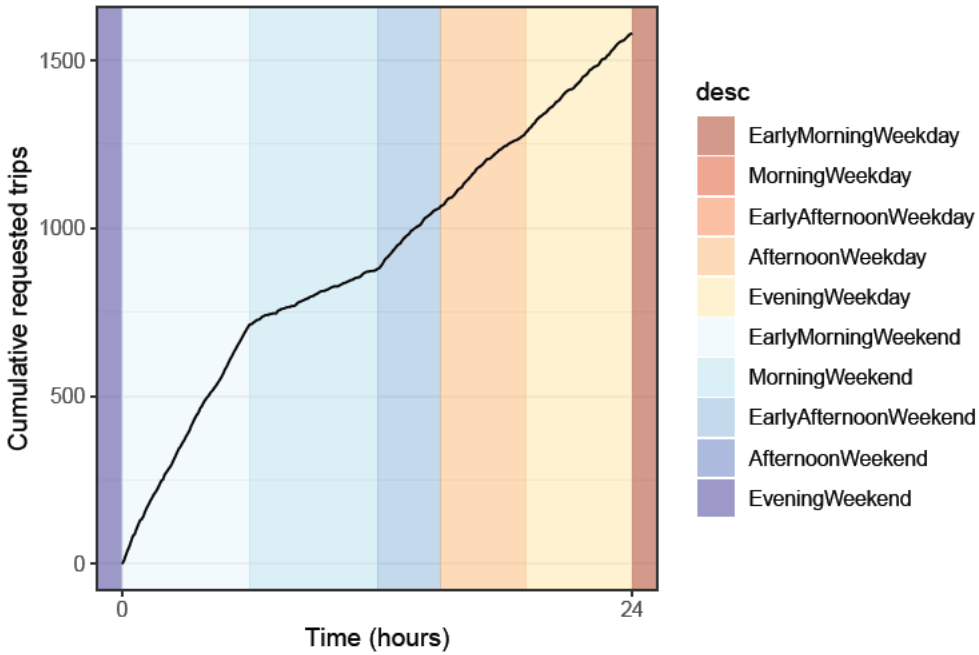
(a) Simulated cumulative number of trips, Beam system, March 2021 demand estimates, Thursday. All times are considered weekdays.



(b) Simulated cumulative number of trips, Beam system, March 2021 demand estimates, Friday. Times until 15:00 are considered weekdays, then after 15:00 times are weekends.



(c) Simulated cumulative number of trips, Beam system, March 2021 demand estimates, Saturday. All times are considered weekends.



(d) Simulated cumulative number of trips, Beam system, March 2021 demand estimates, Sunday. Times until 15:00 are considered weekends, then after 15:00 times are Weekdays.

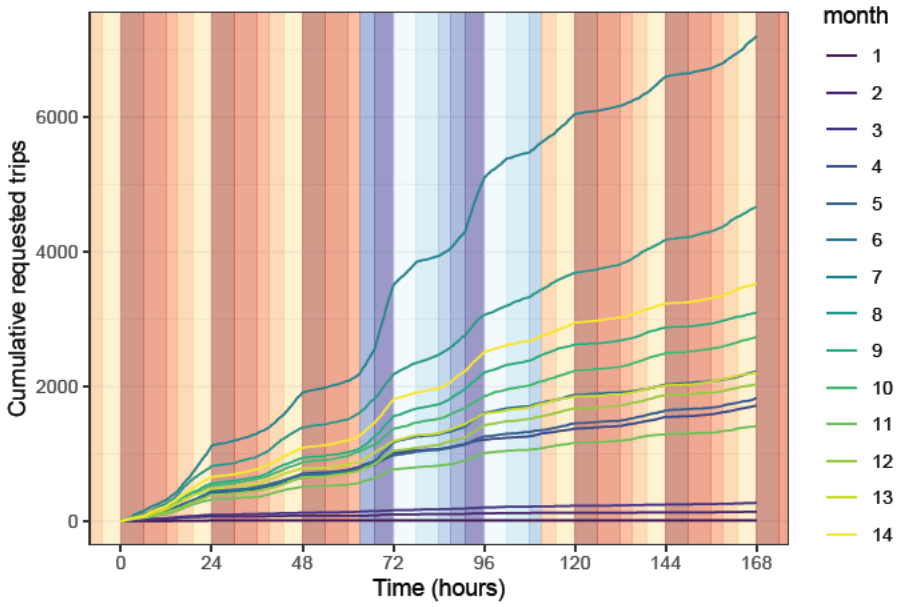
Figure 4.4: Cumulative number of simulated requested hires in the Beam system using demand estimates from March 2021, which were the highest on average. Simulations were run for 24 hours from midnight, for each day from Thursday to Sunday to cover all combinations of time categories. Vertical scales vary between plots due to the weekend demand being higher than Weekday demand. One simulation was used in each case to represent the behaviour of the model.

We next consider generating trips using demand estimates for different months. Figure 4.5a plots the cumulative hires generated in simulations across all months in the data, for a single Wednesday-Wednesday week to show all time categories. Figure 4.5b focuses on simulations in two months in particular, month 6 and 13, which correspond to February and September 2021 respectively. These two months have a similar number of requested trips in the week-long simulations. However, we see that during month 9, corresponding to May 2021, the demand is more consistent between Weekdays and Weekends, whereas during month 10, June 2021, the demand increases during the weekend time periods, allowing the overall hire total to be similar despite a lower Weekday demand. Similar trends appear in other pairs of months, with the effects of Weekends and times of day differing slightly month to month.

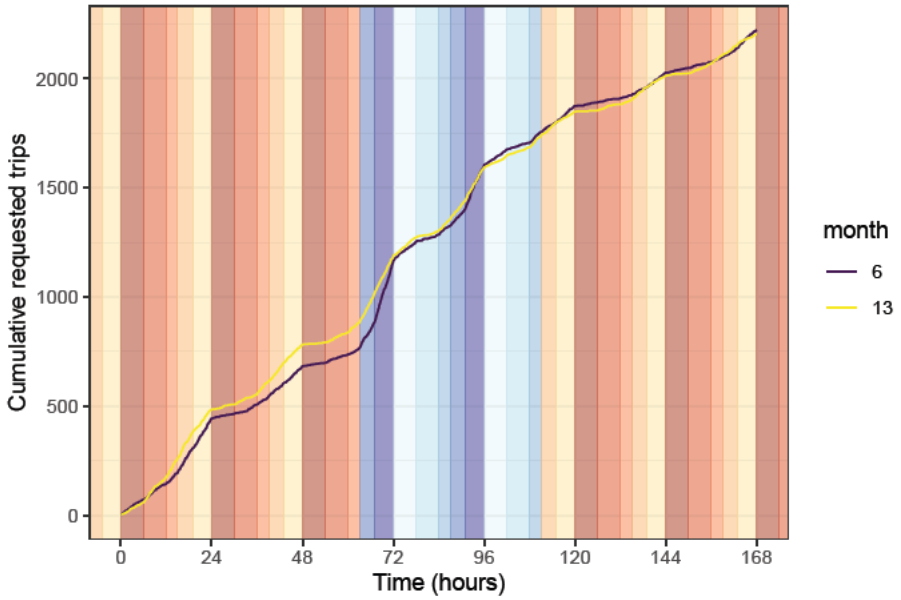
In the simulation model we check if scooters are present in the cell where trips are requested, to designate the trip as successful or unsuccessful and record this result. This provides several potential metrics for measuring the efficiency of a vehicle deployment. We first consider the number of successful and unsuccessful trips during simulation.

Figure 4.6 shows the cumulative number of successful and unsuccessful trips taken place in one week-long simulation. Initially, the number of successful trips is approximately equal to the number of unsuccessful trips, indicating that with a naïve initial deployment of vehicles about half of the requested trips in the system are fulfilled. The frequency of successful trips decreased significantly over time, however. After 72 hours of the simulation, the number of successful trips reached 960. However, between 72 and 168 hours only 11 further trips succeeded, indicating that after the first three days trips were very unlikely to be successful. Due to this, periodically redistributing vehicles to areas with greater demand, where missed trips are requested, is likely to improve the system efficiency by allowing for more successful trips to occur. We further investigate this effect in Chapter 5. As the rate of successful trips decreases, the number of unsuccessful trips approaches the total number of requested trips. We plot the locations of these missed trips in the simulation in Figure 4.7. The locations shown correspond primarily to the areas with highest demand, as shown in Section 3.6, including the Adelaide Railway Station on North Terrace and the ends of Rundle Mall. However, we also see that the corner of Rundle Street and East Terrace has a particularly high number of missed trips. As the initial deployment has sampling weights based on the total number of hires in each area, this inflated number of missed trips may be due to the cell not containing scooters as often in the data compared to other hotspots, and hence reducing the number of hires in the data and in turn reducing the number of scooters initially allocated to the area prior to simulation.

The final metric we consider when analysing the efficiency of a deployment using simulation is the revenue generated from the simulated trips. Here we only consider revenue from successful trips, as missed trips do not contribute to the overall revenue of the e-scooter ride sharing system. Figure 4.8 shows the cumulative revenue over the

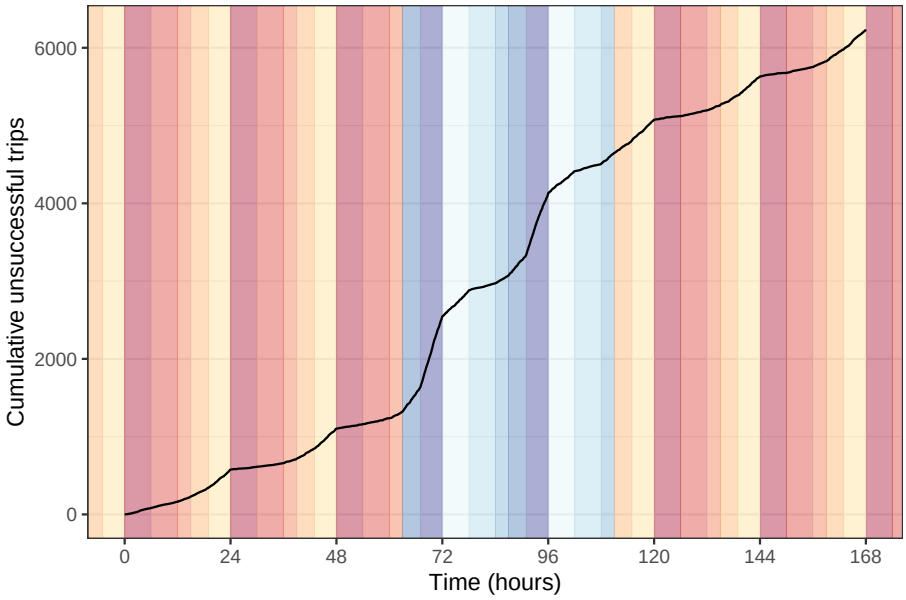


(a) Simulated cumulative requested trips in the Beam system using demand estimates from all months. Simulations are started at 12:00am Wednesday and run for one week.

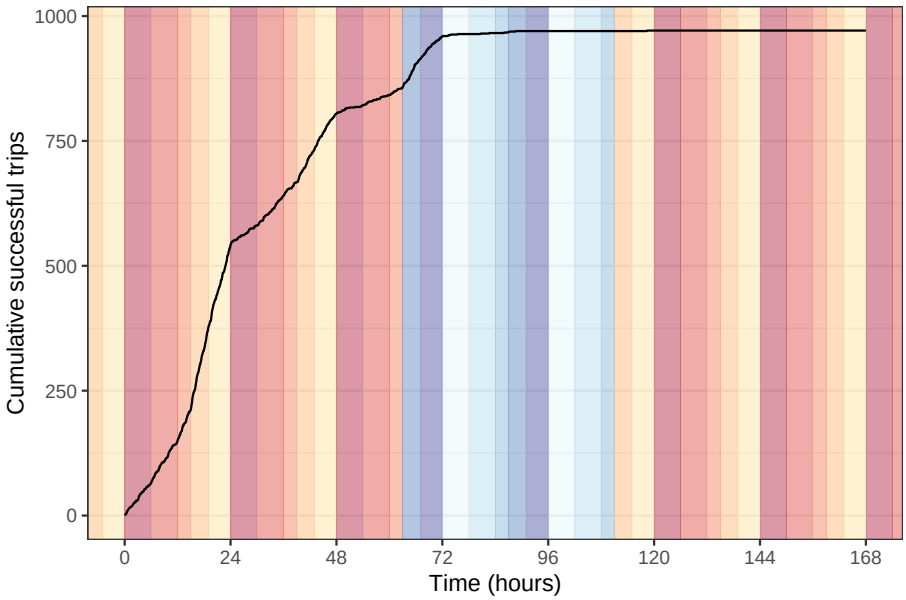


(b) Simulated cumulative requested trips in the Beam system using demand estimates from February (month 6) and September (month 13) 2021. Simulations are started at 12:00am Wednesday and run for one week. The two months have a similar number of requests in the week, however month 10 has a relatively higher Weekend demand and a lower Weekday demand.

Figure 4.5: Simulated cumulative requested trips using the Beam demand estimates, by month. All months show similar trends in the number of hires, with higher demand during the Evenings, especially during Weekends. One week-long simulation, starting on Wednesday at 12:00am, is shown for each month.



(a) Simulated cumulative unsuccessful trips in the Beam system using demand estimates from March 2021. Simulations are started at 12:00am Wednesday and run for one week. Unsuccessful trips are trips that are requested in a cell where no scooters are available.



(b) Simulated cumulative unsuccessful trips in the Beam system using demand estimates from March 2021. Simulations are started at 12:00am Wednesday and run for one week. Successful trips are trips that are requested in a cell where scooters are available. The frequency of successful trips greatly decreases over time due to scooters being taken to areas with lower demand.

Figure 4.6: Simulated cumulative successful and unsuccessful trips using the Beam demand estimates from March 2021. A naïve initial deployment, weighted by the total number of hires in each cell in the data, is used to generate the locations of scooters at the start of the simulation.

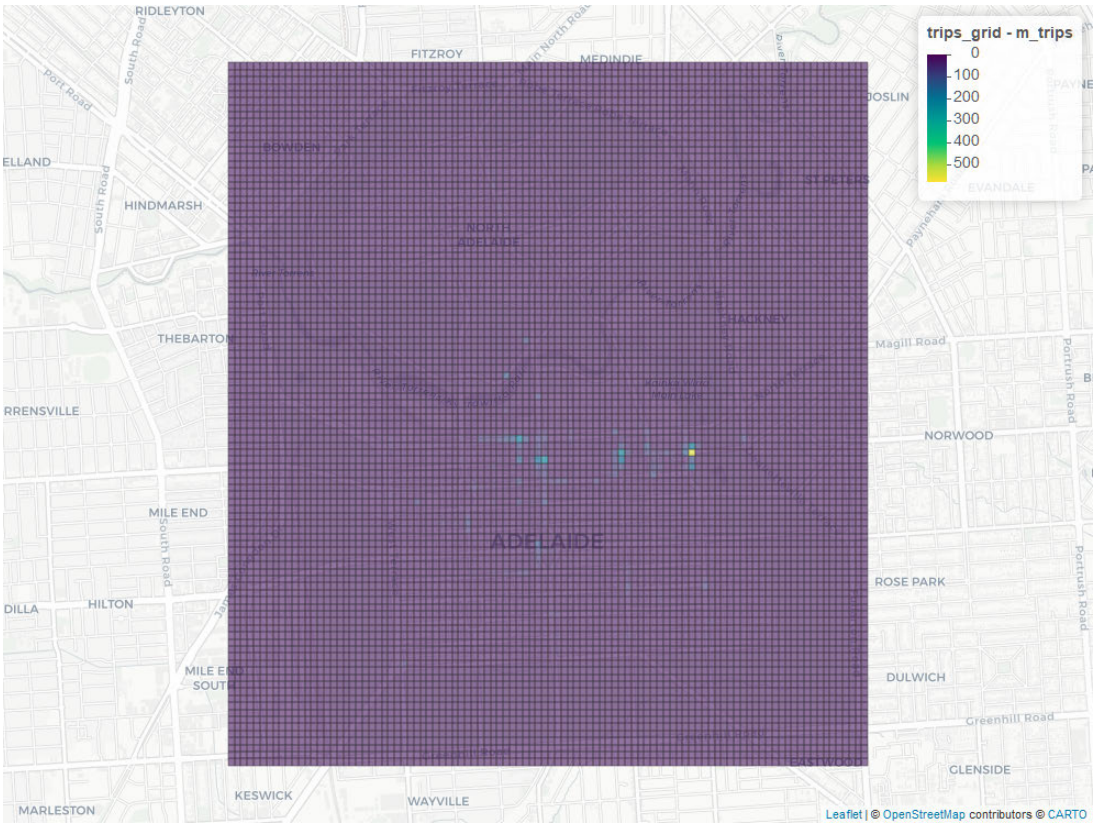


Figure 4.7: Map of the number of simulated trips that are unsuccessful. Demand estimates from the Beam system in March 2021 are used. The locations shown are similar to the demand hotspots, however the corner of Rundle Street and West Terrace has a relatively high number of missed trips, indicating that the naïve estimate is underdeploying vehicles to this location.

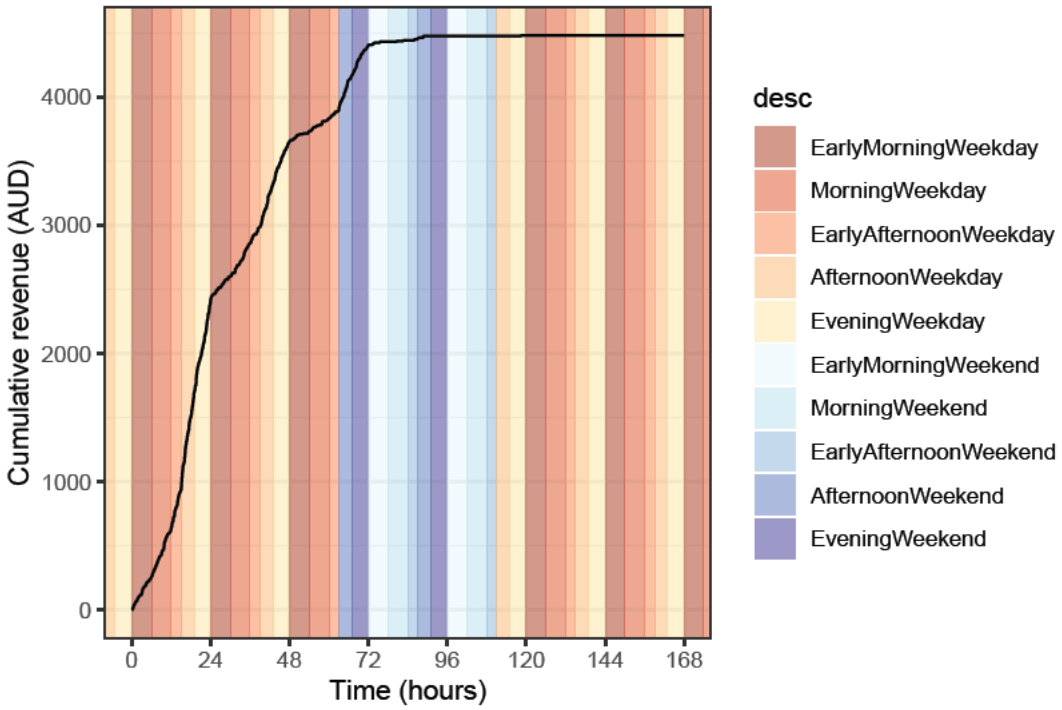
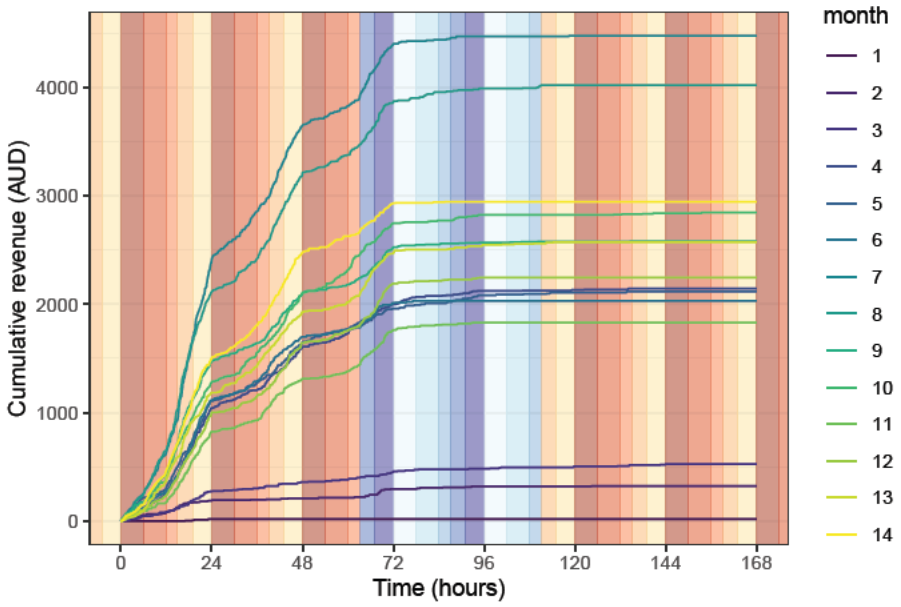


Figure 4.8: Simulated revenue from successful trips in the Beam system using March 2021 demand estimates and a naïve initial deployment.

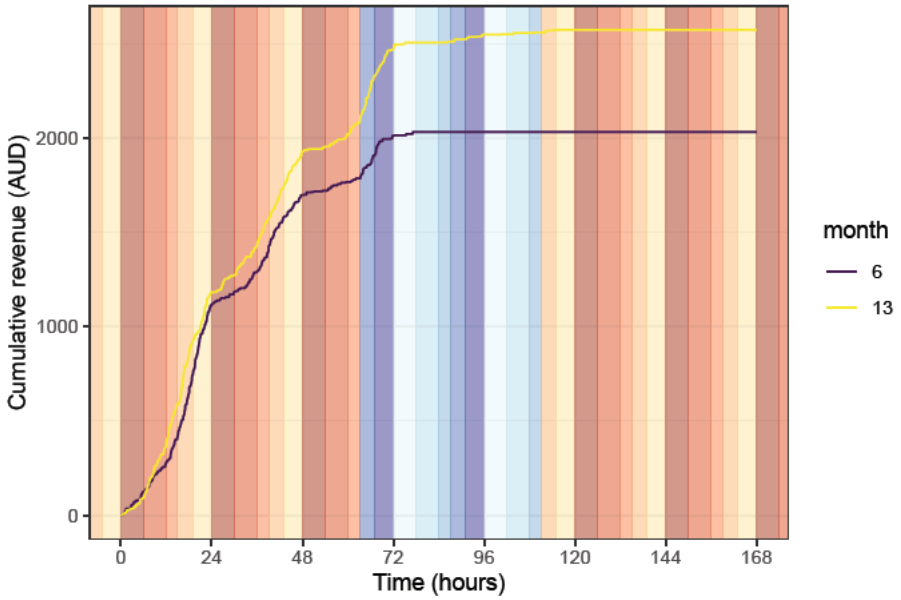
course of the week-long simulation using March demand estimates. We see a similar trend to the cumulative number of successful trips; the revenue quickly approaches a plateau at approximately $4390 after 72 hours, and only a small increase in revenue is recorded after this time due to the lack of successful trips. The difference between the revenue and successful trips is due to the variance in trip duration, and hence cost, between origin cells for trips.

We next compare the difference in revenue during simulations for all months. Figure 4.9a shows the cumulative revenue through the week-long simulation in each month. We see that the revenue follows a similar pattern between all months, increasing in the same general way as the number of requested trips before reaching a constant maximum value after about 72 hours. However, in Figure 4.9b we see that the revenue generated during months 6 and 13, which previously showed very similar numbers of requested trips in Figure 4.5b, have differing amounts of total revenue with the same initial deployment. Month 6, corresponding to the demand estimates from February 2021, reached a total revenue of $2032, which is lower than the $2572 generated in the simulation for Month 13, corresponding to September 2021. This indicates that the number of successful trips and revenue are not solely based on the total number of requested trips.

Finally, we investigate the variability of results from the above simulations. We con-



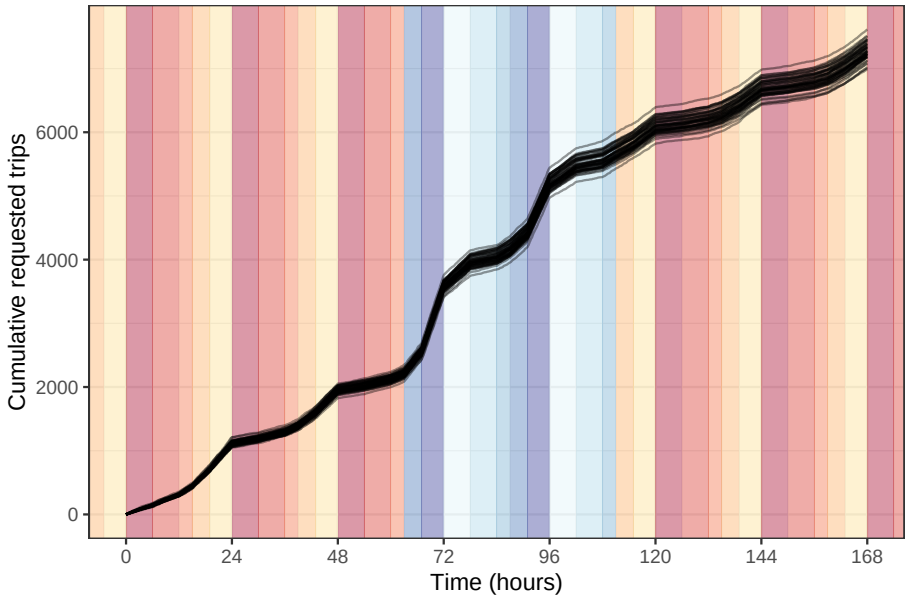
(a) Simulated revenue from successful trips in the Beam system across all months, using the same naïve initial deployment for each simulation.



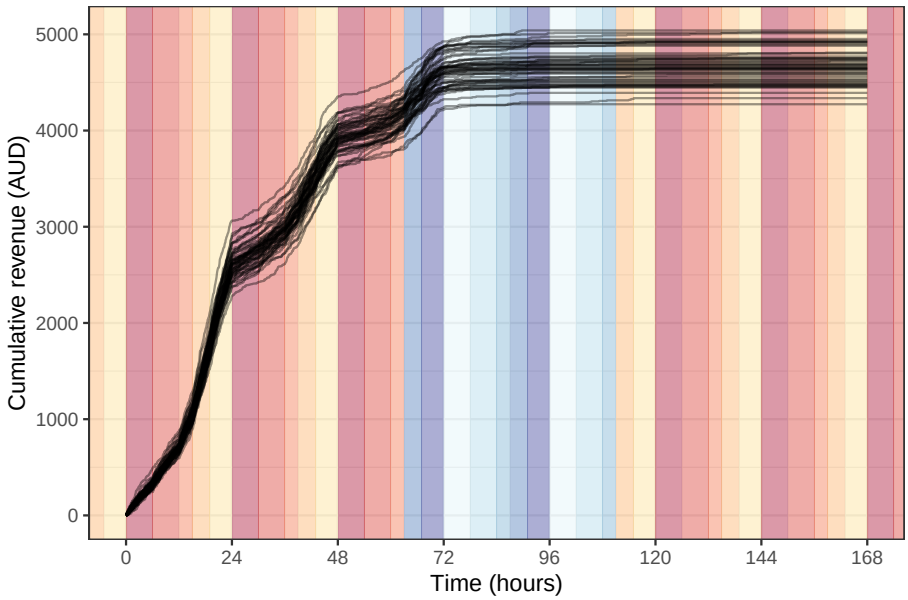
(b) Simulated revenue from successful trips in the Beam system using demand estimates from February (month 6) and September (month 13) 2021, using the same naïve initial deployment for each simulation.

Figure 4.9: Simulated cumulative successful and unsuccessful trips using the Beam demand estimates from March 2021. A naïve initial deployment, weighted by the total number of hires in each cell in the data, is used to generate the locations of scooters at the start of the simulation.

sider both the cumulative requested trips and revenue generated in several week-long simulations during March 2021, denoted Month 7, the month with the highest overall system demand and number of hires in the above simulations. Figure 4.10 shows the variance in both of these values throughout 50 simulations. For each simulation a different seed is used, causing the initial deployments to vary as well as the trips generated. We see that both metrics vary between all simulations, throughout the simulations. The total number of requested trips was between 6990 and 7614, with a median of 7271 trips. The sample standard deviation of the final number of requested trips is 136 trips. The relative variance in the total revenue is higher, with a range of $4273 to $5041 and a median of $4659, resulting in a standard deviation of $187. This increased variance emerges in the simulations earlier than in the cumulative requested trips, with clear differences appearing within 24 hours of the start of the simulations.



(a) 50 simulated sample paths of requested trips generated in the Beam system using demand estimates from March 2021, with a one-week simulation period.



(b) 50 simulated sample paths of revenue generated in the Beam system using demand estimates from March 2021, with a one-week simulation period.

Figure 4.10: Simulated cumulative successful and unsuccessful trips using the Beam demand estimates from March 2021. A naïve initial deployment, weighted by the total number of hires in each cell in the data, is used to generate the locations of scooters at the start of the simulation.

4.5 Discussion

In this chapter, we built upon the demand estimation performed in Chapter 3 to construct a digital twin of an e-scooter ride-sharing system in Adelaide. This two-stage simulation model allows us to investigate in detail the behaviour of trips that occur in the system, and implement alternative strategies to manage the system. We further investigate the deployment and redistribution of vehicles in such a system in Chapter 5, which uses the simulation model as part of an optimisation regime.

We also developed several metrics to assess the efficiency of an e-scooter ride sharing system from the simulation results. The simplest of these metrics is the total requested trips, which depends only on the demand in each section of the service area over time. However, the revenue generated from successful trips accounts for vehicle positions through time, and is hence a more uiseful and accurate measurement for determining the optimal deployment of vehicles across the service area. We use the revenue metric in the optimisation algorithm in Chapter 5.

In developing the simulation model, we assume that all vehicles have sufficient battery charge to make any potential trips. This is supported by the battery data analysis performed in Section 2.5, which shows the battery usage during trips to be relatively low compared to the average charge of idle scooters. However, recharging must occur during the operation of an e-scooter system. The recharging problem provides an opportunity for new research into the optimal recharging strategy for free-floating ridesharing systems such as the e-scooter systems. This is not performed in this work as incorporating battery information into the simulations in a meaningful way increases computational time, and has only a minimal impact on the short time-scales we use for analysing redistribution requirements.

Chapter 5

Deployment Optimisation

5.1 Introduction

In previous chapters, we discussed the patterns of demand estimates across the service area of e-scooter ride sharing systems in Adelaide, and developed a digital twin of the physical system to estimate the effectiveness of a deployment using the revenue generated. These provide the foundations of the optimisation algorithm we develop in this chapter to find the best deployment of vehicles for various situations.

The problem of deploying vehicles or stations in a ride-sharing system has been the target of several studies in the past, with numerous optimisation algorithms implemented. Brandstätter et al. (2017) propose an algorithm for optimising station deployment based on identifying possible routes between stations and iteratively adding these to a network graph, subject to capacity and budget constraints. In a pre-determined network of stations, Correia et al. (2014) use a mixed integer program to find the deployment between stations that maximises the daily profit for a ride-sharing operator. In a free-floating system, Masoud et al. (2019) mixed-integer linear program to optimise the assignment of "chargers", people responsible for recharging and redistributing e-scooters, to improve the efficiency of the recharging process for an e-scooter ride-sharing system. Due to the high dimensionality of the deployments we consider, and the variability in the estimate of expected revenue, we use stochastic optimisation techniques to optimise the deployment. Our problem is similar to that posed by Oudani (2021), who implemented a simulated annealing algorithm to solve their mixed-integer linear program for optimising inter-modal transportation. For these reasons, we use a simulated annealing algorithm to find the e-scooter deployments that maximise the profit for an e-scooter operator.

In Section 5.2 we discuss the background of simulated annealing, the optimisation algorithm we use to optimise the deployment of vehicles across the service area. In Section 5.3 we detail the components and parameters of the simulated annealing algorithm that we choose to assist in the the optimisation we perform. Finally, in Section 5.4 we

discuss the results of the optimisation, including the convergence and variability properties of the expected revenue of a deployment as the algorithm progresses, and the distribution of vehicles across the service area in an optimal deployment.

5.2 Simulated Annealing

We consider the vehicle deployment $\boldsymbol{d}$ as an integer vector containing the number of vehicles in each cell;

$$\boldsymbol{d} = (d_0, d_1, ..., d_{n^2-1}), \tag{5.1}$$

where d_i is the population of cell i, for $i = 0, 1, ..., n^2 - 1$, and n^2 is the total number of cells. We impose the restriction that a finite number of scooters, N, is distributed, so we have

$$\sum_{k=0}^{n^2-1} d_k = N.$$

As discussed in previous chapters, the chosen grid we consider contains 10,000 cells with $n = 100$, and following the simulations in Chapter 4 we use $N = 1400$. Due to the high dimension of the space of possible deployments, which form the solution space for optimisation, we implement simulated annealing to maximise the expected revenue. Simulated annealing also allows for the algorithm to leave a local minima that may exist in the complicated and highly variable solution space (Schneider & Kirkpatrick (2006)).

Our construction of the simulated annealing optimisation algorithm follows the Homogeneous Simulated Annealing Algorithm proposed by Diekmann et al. (1993). Firstly, we define the expected revenue of a deployment $\boldsymbol{d}$ to be $f(\boldsymbol{d})$, the function we intend to maximise. This function is discussed further in Section 5.3. We also denote the deployment at iteration k of the algorithm to be $\boldsymbol{d}_k = (d_0^{(k)}, d_1^{(k)}, ..., d_{n^2-1}^{(k)})$, where $d_i^{(k)}$ is the number of scooters placed in cell i during iteration k, for all cells i. As a stopping criterion, we take the best deployment generated in the algorithm after a fixed number of iterations, K, which we typically take to be 40,000. Later we demonstrate that this value is sufficient for a sufficiently accurate final function value in most cases.

One of the important components of a simulated annealing implementation is the annealing schedule, which controls the temperature at each iteration, which in turn adjusts the probability of accepting a decrease in function value as the algorithm runs. In our implementation, we use an exponential annealing schedule as described by Schneider & Kirkpatrick (2006). We let the initial temperature be T_0, and the temperature at iteration k be denoted T_k. Then, we define

$$T_k = T_0 \alpha^k$$

for some $\alpha \in (0, 1)$. This definition leads to the property that $T_k = \alpha T_{k-1}$, so the temperature at each iteration is the previous temperature reduced by a factor of α. We choose this cooling schedule to allow for frequent jumps in the initial iterations of the algorithm, followed by small refinements in later iterations while maintaining a chance of leaving local maxima.

The final major component of the simulated annealing algorithm is the generation of new potential solutions. This is a critical process within the algorithm, as it allows for a search across the set of potential solutions. There are several desired properties in such this generation algorithm. The primary of these is that the newly generated potential solutions must be *neighbours* of the previous solution; a new solution must be a small distance away from the previous solution to allow for a steady improvement in the function value between iterations. As we have a sense of space in this work, we use this to define what a neighbour means for e-scooter deployment. We consider a solution to be a neighbour of another solution if it can be reached by moving some or all scooters in the deployment to another cell that is adjacent to the cell it is in in the previous solution. Hence, to generate a neighbouring deployment to the previous, we can move some subset of the scooters to an adjacent cell. To move a vehicle to an adjacent cell, we define an *offset vector*, of length 2, where each element is independently distributed with a 0.25 chance of being -1, 0.25 chance of being 1, and 0.5 chance of being 0. These simulated values can then be added to the vehicle's coordinates in the grid to shift it to one of the adjacent 8 cells, or to stay stationary. If a component of the offset vector would move the vehicle outside of the grid, that component is set to zero.

To generate a new solution $\boldsymbol{d}_k$ from the previous deployment $\boldsymbol{d}_{k-1}$, we introduce the parameter R_k, the number of vehicles to move to generate deployment $\boldsymbol{d}_k$. We then use the following algorithm to generate new deployments:

1. Choose a number of vehicles to move, $R_k \sim \text{Exp}\left(\frac{T_0}{T_k R_0}\right)$. If $R_k = 0$, then we set R_k to 1 to ensure at least one scooter moves. If $R_k > N$, the number of vehicles in the system, we set R_k to N, meaning all scooters can move between iterations.

2. Sample a set of vehicles to relocate, denoted $i_1, ..., i_{R_k} \in \{1, 2, ..., N\}$. Each of these vehicles has a current location (cell number), which we denote $c_i^{(k)}$.

3. For each vehicle to relocate, we simulate the offset vector to relocate the scooter to an adjacent cell within the grid.

By introducing the parameter R_k, we allow for the number of vehicles to vary as the optimisation progresses. Initially, R_k will typically be high, allowing a majority of the vehicles to shift every iteration. However, this behaviour is not desired in the later stages of the algorithm, as shifting all vehicles when already at a near-optimal solution is likely to result in a deployment that has a lower expected revenue, and so the algorithm would be inefficient in converging to the optimal deployment. By introducing a dependency

between R_k and T_k, we simultaneously reduce the amount of variability between iterations and reduce the probability of moving out of a local maximum expected revenue.

Following the annealing schedule and neighbour generation components, we can define the acceptance criterion based on the iterative improvement of the objective function $f(\boldsymbol{d}_k^*)$ compared to $f(\boldsymbol{d}_{k-1})$ and the current temperature T_k, where $\boldsymbol{d}_k^*$ is the newly generated solution that we may accept or reject. The acceptance criterion is designed to accept any deployment that improves upon the solution, and probabilistically accept a deployment that decreases the function value, with probability based on the temperature. This allows for local optimal values to be avoided in the early stages of the algorithm, so that the global optimum value can be found. The acceptance criterion we use is defined as follows, at step k:

- *Accept* the new solution $\boldsymbol{d}_k^*$ if $f(\boldsymbol{d}_k^*) > f(\boldsymbol{d}_{k-1})$. In this case, we set $\boldsymbol{d}_k \leftarrow \boldsymbol{d}_k^*$. Otherwise,
- *Accept* the new solution $\boldsymbol{d}_k^*$ with probability $\exp\left(\frac{f(\boldsymbol{d}_k^*)-f(\boldsymbol{d}_{k-1})}{T_k}\right)$. In this case, we set $\boldsymbol{d}_k \leftarrow \boldsymbol{d}_k^*$. Otherwise,
- *Reject* the new solution, and set $\boldsymbol{d}_k \leftarrow \boldsymbol{d}_{k-1}$.

This criterion has several important properties. Firstly, if $\boldsymbol{d}_k^*$ improves upon $\boldsymbol{d}_{k-1}$, we always accept it. Additionally, if the new solution does not improve upon the previous, the probability of accepting the solution is higher if the difference between function values is small, and is also higher for higher temperatures.

In addition to the construction of the simulated annealing algorithm above, we require an initial deployment of vehicles. As we use the demand estimates obtained from a Bayesian framework in the simulations, we use the naïve demand estimates as the basis for the initial deployment. This allows for the initial deployment to be a reasonable approximation for the final deployment, as we expect this optimal solution to match the demand estimates, adjusted for potential differences in trip duration based on location. The particular initial deployment we use is to sample the location of all 1400 scooters weighted by the naïve estimate in each cell, matching the process done to set the initial deployment for simulations in Section 4.4.

5.3 Implementation

The objective function $f(\boldsymbol{d})$ – the function we want to maximise using simulated annealing – is the expected revenue of a given deployment, over a given time period. Due to the high amount of complexity in the e-scooter ride-sharing system, we use the simulation model constructed in Chapter 4 to generate samples of the revenue to obtain an estimate of this expected revenue. However, as discussed in Section 4.4 we showed that due to the high

amount of variability shown in the simulated revenue, a large number of samples is required for accurate estimation of the expected revenue. However, this introduces significant issues in running the optimisation due to the high amount of computational work that needs to be done for every iteration. To solve this, we utilise the two-phase property of the simulation model, where the trip generation can be separated from the vehicle movement. As the trip generation does not depend on the scooter locations, we can generate a large number of trips in advance for all iterations of the optimisation algorithm, then apply the e-scooter movement stage of the simulation algorithm to calculate the revenue from each simulation. This reduces the amount of time spent during the optimisation algorithm each iteration, however may introduce bias into the estimates, as the same simulations are used for all iterations. This would cause the optimisation to approach the optimal solution for that specific set of simulations, rather than all possible situations. This effect can be reduced by using a large number of simulations, to provide a sufficient sample size for estimating revenue. To balance this bias and variability with the required computational time, we use 100 simulations for our implementation of simulated annealing. By using a large number of independent simulations, we allow for parallelisation to be used to increase the speed of the algorithm.

Due to using a fixed set of simulations to estimate the expected revenue of any deployment has the limitation that the sample mean revenue has a maximum value. Consider a deployment that allows all trips to be fulfilled in all 100 simulations. Then, no matter how the deployment is adjusted, as long as all trips are successful there can be no improvement, even though it is possible that if the deployment was implemented in the physical system that a trip may be unsuccessful. Any deployment such that all trips in all simulations are successful would provide the maximum value for the estimated expected revenue, which is equal to the mean over all simulations of the total revenue generated by all trips. If this limit is attained, or approached, during the optimisation algorithm, then the deployment cannot be improved any further given the set of simulations. This effect is minimised in our implementation by using a large number of trips, as it becomes extremely unlikely for there to be a deployment of 1400 vehicles that completely satisfies all trips simulated across 100 individual simulations, so there should always be trips that are unsuccessful. However, the limit can still be calculated, and hence provides a useful measurement for the efficiency of a deployment and assessing the convergence of the algorithm in Section 5.4.

There are also several further parameters we tune to allow the simulated annealing algorithm to work efficiently. The parameters for the annealing schedule, the initial temperature T_0 and the cooling rate parameter α, were chosen to be 100 and $T_0^{-4/n_{iter}}$ respectively, where n_{iter} is the total number of iterations of the simulated annealing algorithm we perform. The value of T_0 was chosen arbitrarily, whereas the value of α was tuned to provide a large opportunity for the deployment to move around freely for a large number of iterations with a high probability of leaving a local maximum value. We have

that for the given value of α, the temperature at iteration k is given by

$$\begin{aligned} T_k &= T_0 \alpha^k \\ &= T_0 \times T_0^{-4k/n_{iter}} \end{aligned}$$

such that when $k = n_{iter}/4$, $T_k = 1$, or 1% of the initial temperature. As we define the number of vehicles to move R_k to be a function of the current temperature T_k, this means that after a quarter of the iterations, we expect 1% as many vehicles are moved compared to the initial iterations, allowing for the algorithm to more precisely find an optimal deployment.

5.4 Results

5.4.1 Convergence and Variability

The first component of the optimisation results we investigate is the long-term behaviour and convergence of the algorithm. To do this, we consider one of the busiest periods for the system in terms of demand: Friday afternoons starting 6pm during April 2021. The time window we consider is 6 hours, so the simulations run from 6pm to 12am, covering the Weekend Evening time category. The algorithm was run for 70,000 iterations, recording the proposed and cumulative best function values at each iteration. The maximum value of the expected revenue, given by the mean total revenue over all simulations assuming that all trips are successful, is \$824. Figure 5.1 shows the evolution of the deployments as the algorithm runs by plotting both the current function value at that iteration, $f(\boldsymbol{d}_k^*)$, in black, and the running maximum function value in red. We see several important features in the plot. The most prominent of these is that for the first 20,000 iterations, the function value appears to not increase, rather it fluctuates wildly. This is due to the relatively high temperature, causing the acceptance probability to be close to 1 for all deployments, including ones that decrease the expected revenue. However, there is still a bias towards better solutions as the probability is not equal to 1. After 20,000 iterations, the running maximum value appears to follow a smooth curve, increasing rapidly at first then flattening out, with a small increase at the end of the 70,000 iterations. This trend implies that an optimal solution exists, however the algorithm has not converged to an optimal deployment yet. The final value reached is \$741, which corresponds to the deployment shown in Figure 5.2. While this is significantly higher than the expected revenue given by the initial deployment of \$291, it is still short of the theoretical maximum value for the set of simulations, which is \$824. The difference of \$83 is likely due to a combination of the lack of convergence, and there not being enough scooters in the system to allow all trips requested in the system for the 6-hour period possible.

We also consider the vehicle deployment generated by the simulated annealing algorithm by mapping the deployment that provided the maximum expected revenue, \$741,

for the case described above. Figure 5.2 shows the location of all 1400 vehicles using cell populations across the grid. Upon first inspection, the deployment of vehicles appears similar to the Bayesian demand estimates, as shown in Section 3.6, which is to be expected as these estimates provide the basis of the hire process used to estimate revenue. However, we see that at most 4 scooters are deployed to any cell in the service area, despite the demand in the hotspots being more than 4x the demand in quieter areas. Hence, the optimal solution given a 6-hour time window is relatively more spread out than the demand estimates. There are also a large number of cells in outlying areas, which have very low estimated demand, that are allocated one scooter each. These are likely outliers in the deployment; they may be vehicles that get stuck outside of the busy areas, and have little effect on the expected revenue. These may eventually move back into areas where they are likely to be used if the algorithm is allowed to continue for a larger number of iterations. We note that this is not yet the optimal solution, as the algorithm had not converged, however the expected revenue generated by this deployment is higher than the expected revenue generated by a naïve deployment using only the demand estimates.

As discussed previously, the optimisation algorithm in this case has not yet converged as the function value is still increasing, and has not hit the theoretical maximum value yet. However, it may be sufficient for the problem of scooter deployment to stop the algorithm before reaching an optimal solution. Here, we consider the change in the running maximum function value – the red line in Figure 5.1 – every 1000 iterations, for all 70,000 iterations. That is, the value $f(\boldsymbol{d}_k) - f(\boldsymbol{d}_{k-1000})$ for $k = 1000, 2000, ..., 70,000$. These differences in expected revenue are plotted against the number of iterations in Figure 5.3. There is a clear spike in the first few iterations, corresponding to the early rise from the naïve deployment in the plot of the revenue generated. After a long period of no change, corresponding to the random fluctuation in function value due to a high temperature, we see another increase in the change, which slowly decays as the algorithm continues. By iteration 40,000, we see that the change per 1000 iterations stays at less than \$5, with a majority of the changes being less than \$2. However, we showed in Section 4.4 that the variability in revenue between simulations is substantial. For the set of simulations used in this optimisation, the sample standard deviation of the total revenue in each simulation is \$67, much higher than the difference in estimated expected revenue per 1000 iterations. This indicates that changes in expected revenue during the optimisation algorithm after the first 40,000 iterations are less than the variability in the simulations. Hence, to reduce the computational time required to obtain deployments, we stop the algorithm after 40,000 iterations are reached, in all cases. This limit can be increased, along with increasing the number of simulations used, to generate more accurate solutions to the problem of vehicle deployment.

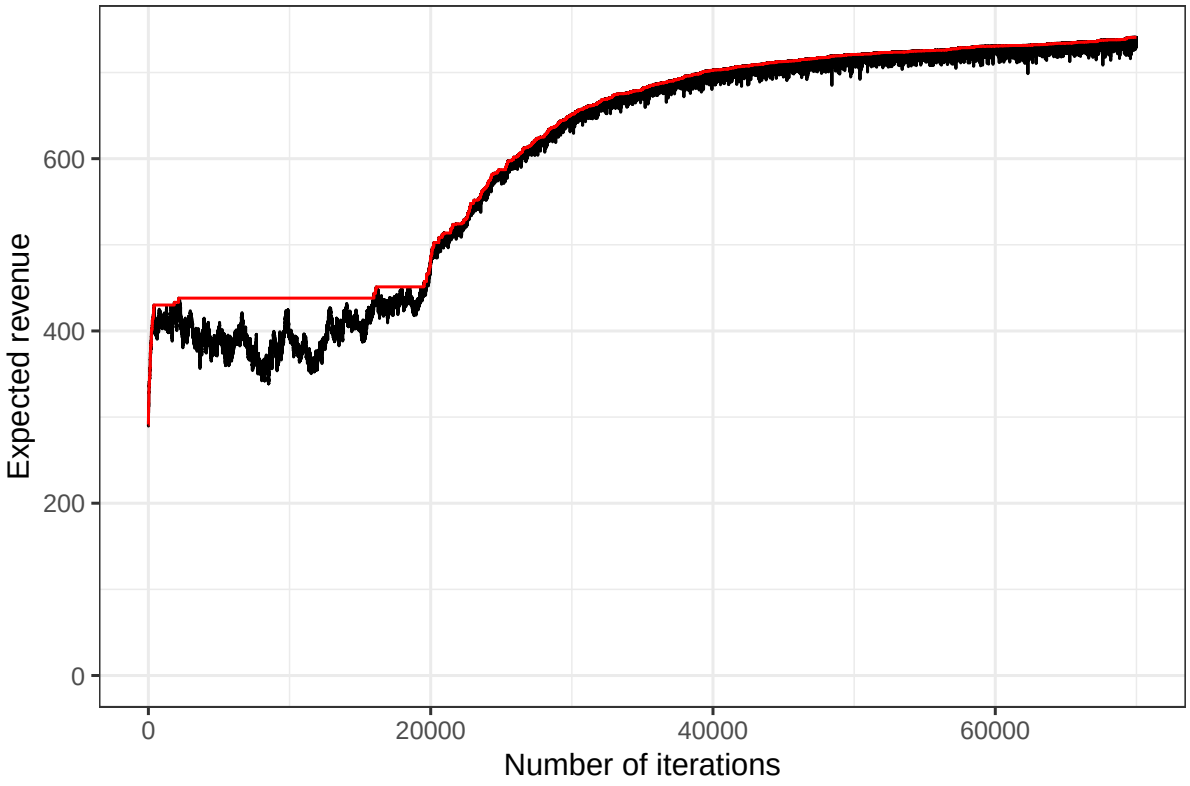


Figure 5.1: Long-term evolution of the expected revenue from vehicle deployments during the simulated annealing algorithm. The time window used to estimate revenue is 6pm to midnight on Weekends, during April 2021. The high variability in the first iterations is due to the high temperature, allowing many solutions to be accepted despite these solutions typically reducing the function value. This effect is not seen after about 20,000 iterations, as the temperature is low enough that it is rare for a deployment to be accepted if it is worse than the previous deployment.

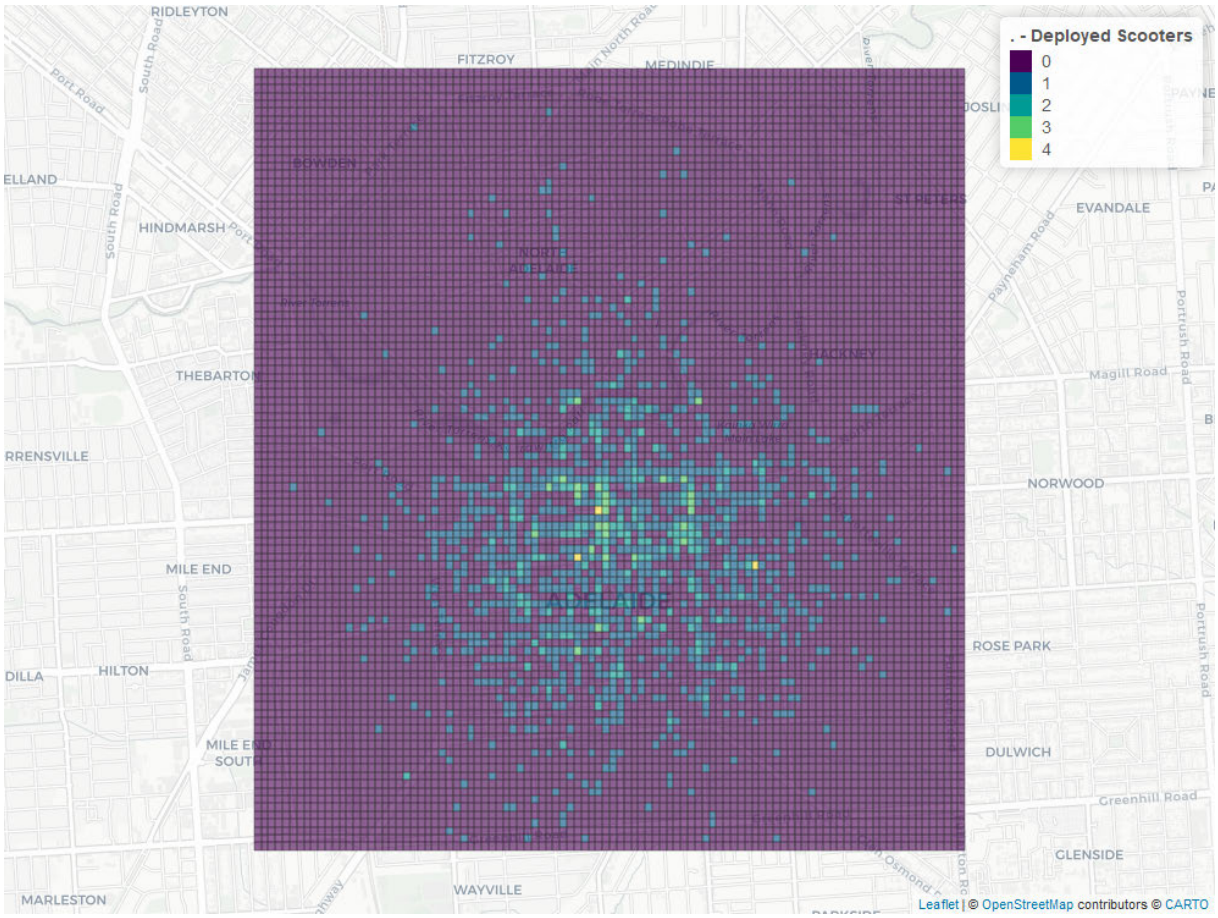


Figure 5.2: Best deployment of scooters across the service area after 70,000 iterations of the simulated annealing algorithm, for the 6 hours between 6pm and midnight on a weekend during April 2021.

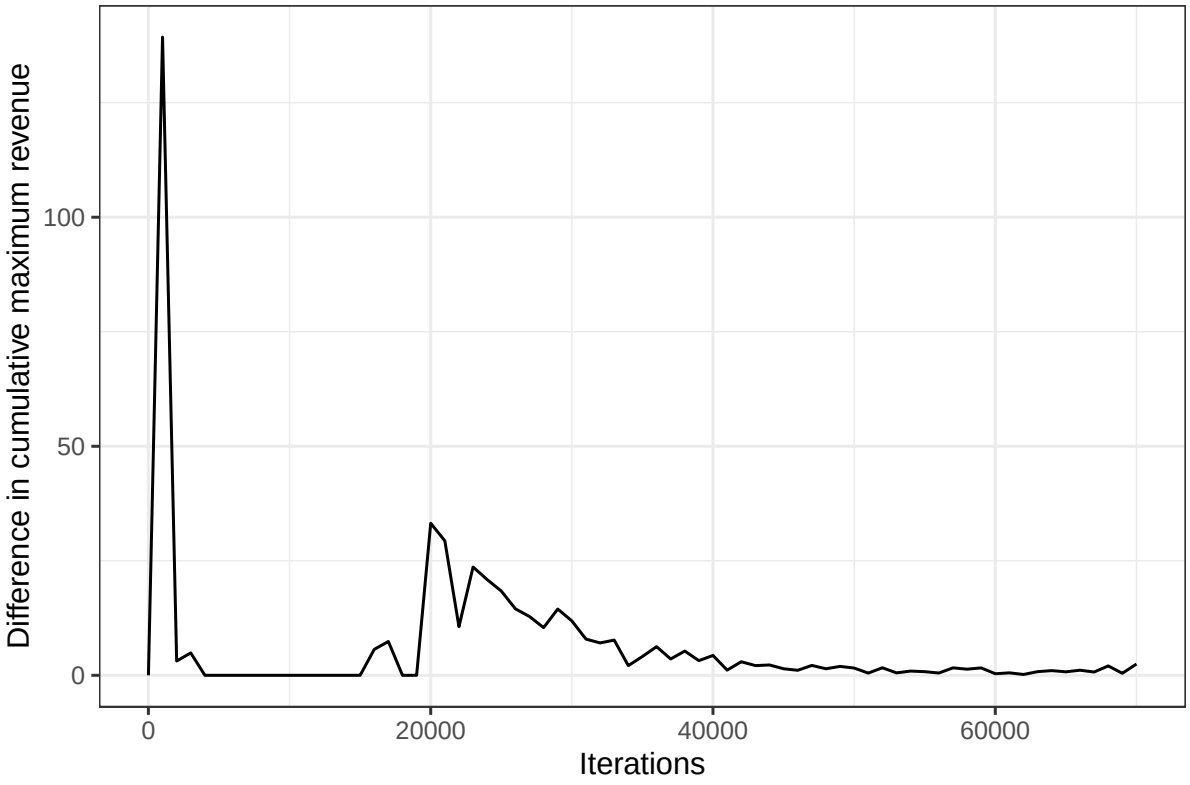


Figure 5.3: Change in running maximum expected revenue per 1000 iterations throughout the simulated annealing applied to vehicle deployment for the Weekend Evening time category during April 2021. The change appears to be insignificant after about 40,000 iterations, changing by at most $2 per 1000 iterations after this time.

5.4.2 Optimal Deployments by Month

In this section, we apply the simulated annealing algorithm to optimise the deployment across all months in the data. We consider only months 5 to 14, corresponding to months January 2021 to October 2021 as these months have high overall system demand. For each month, we use that month's demand estimates both to generate an initial deployment and in the simulations. All instances of the simulated annealing algorithm are performed independently, with 100 independent simulations in each case. We also use the Weekend Evening time category for all simulations, as this is the time with the highest demand for most months.

Following from Section 5.4.1, we use 40,000 iterations for all months. However, due to computational time constraints the algorithms for Month 7 and Month 14 did not reach 40,000. Instead, partial results are used from iteration 32,000 for Month 7 and 36,000 for Month 14.

Figure 5.4 shows the evolution of function values during the simulated annealing algorithm, with both the current and running maximum function value for all iterations for all months. We see similar behaviour between all months, with fluctuations in the function values for the early iterations followed by a steady climb after the first 10,000 iterations. The convergence behaviour varies by month, with paths corresponding to months with a lower maximum expected revenue, such as Months 11 and 5, appearing to converge more readily than those with higher revenues. This is likely due to the total number of unique trips taken across the simulations being lower, so the probability of the algorithm reaching a deployment where a majority of trips are successful is higher. Conversely, for the months with highest demand such as Months 7, 8, and 14, the expected revenue appears to still be increasing at the end of the iterations. This shows that, when using our implementation of the simulated annealing with 100 simulations, an optimal value is obtained more readily for cases with a lower overall demand than for cases with a higher overall demand, where more iterations are required. Due to this, when using the simulated annealing approach to optimising the deployment of vehicles, a threshold-based termination metric should be used rather than a fixed number of iterations.

We also plot the change in expected revenue between the initial deployment and best deployment in Figure 5.5. This shows the increase in expected revenue after running the optimisation algorithm. In all cases, the revenue increases by at least 130%, with the lowest increase occuring in Month 6, corresponding to February 2021, with an increase of 132%. Months 9 and 10 show the highest improvement, increasing the expected revenue by 169% and 172% respectively. This indicates that revenue can be improved across all months by using simulated annealing to optimise the vehicle deployment, however the amount of increase can change depending on the characteristics of the month's demand and usage patterns.

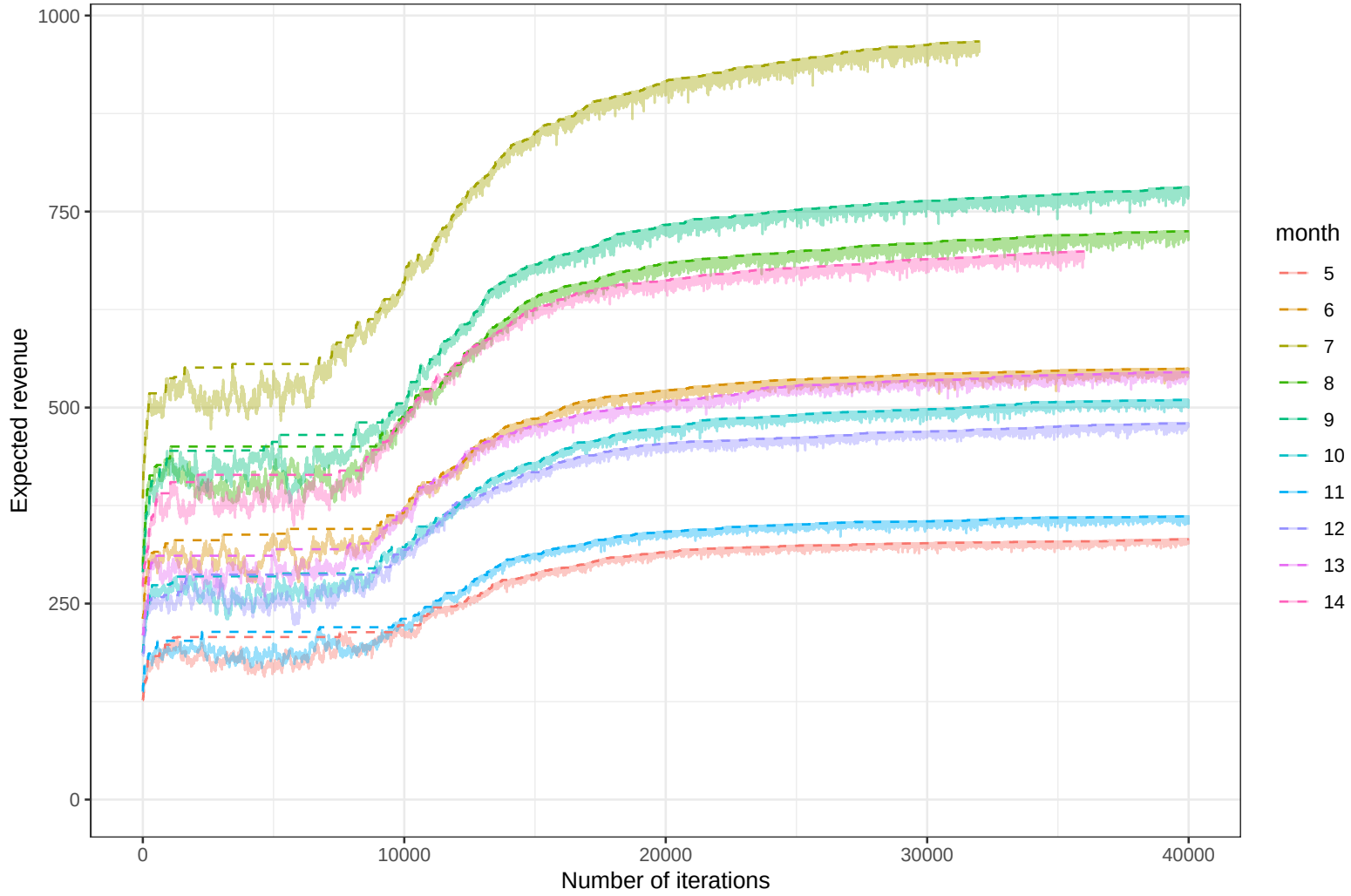


Figure 5.4: Progression of function values during the simulated annealing algorithm applied to months between January 2021 (month 5) and October 2021 (month 14). The same parameters were chosen for all cases, however the initial deployments and simulations are independently sampled. 40,000 iterations were used in all cases except for months 7 and 14, which terminated due to time limits.

We also investigate the differences in the optimal deployments between particular months. Here, we consider Months 5, 6, and 7, corresponding to Janurary to March 2021, and Month 11, corresponding to August 2021. The first three months demonstrate the optimal deployment relatively early during the system operation and how the optimal deployment changes for the Adelaide Fringe festival during February-March 2021, whereas the August deployment demonstrates a typical deployment. Figure 5.6 shows the deployments for all four months.

In all cases, we see a large amount of vehicles deployed to areas with little demand in the outskirts of the service area. Similarly to the outliers shown in Figure 5.2, it is likely that these are due to the objective function evaluating the entire deployment, allowing some vehicles to be poorly used. This effect would likely be reduced by running the algorithm for a longer period of time, allowing all scooters to move to locations where they can be more effectively utilised. Another feature of the deployments that is common between all deployments is the positioning of vehicles along major roads in the CBD such as North Terrace and King William Road, with major hotspots on the corners of King William Road and Rundle Mall, King William Road and North Terrace, and outside the Adelaide Railway Station on North Terrace. The placement of a large number of vehicles here is expected as these are the locations with the highest estimated demand, leading to a large number of requested trips during the simulations. However, in general the cell populations in all optimised deployments are relatively more even across the service area when compared to the distribution of demand. We see a maximum of 7 vehicles distributed to any cell, with cells in busy areas typically receiving 2-4 vehicles, compared to the demand estimates where the demand reaches a peak of around 1 hire/hour in the hotspots, and 0.2-0.4 hires/hour in similar cells.

There are also some differences in the deployments between months. During March 2021, the deployment map appears to differ from the deployments in other months, with an additional hotspot on the east side of the CBD, likely due to the Adelaide Fringe festival generating additional demand for vehicles in this area. The deployment distribution for August also appears to differ slightly compared to the earlier months, with more vehicles deployed to streets on the eastern side of the service area. This change is likely due to the evolution of e-scooter commuter patterns over the 5 months between March and August 2021.

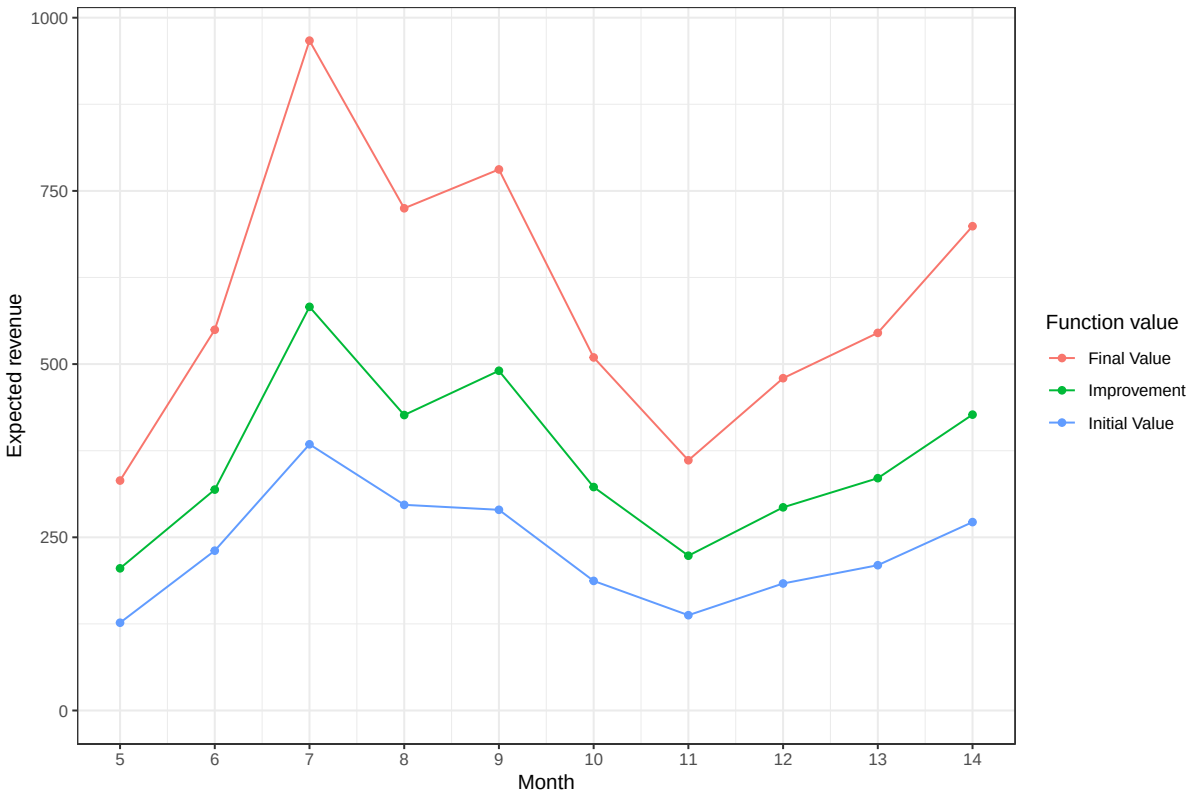
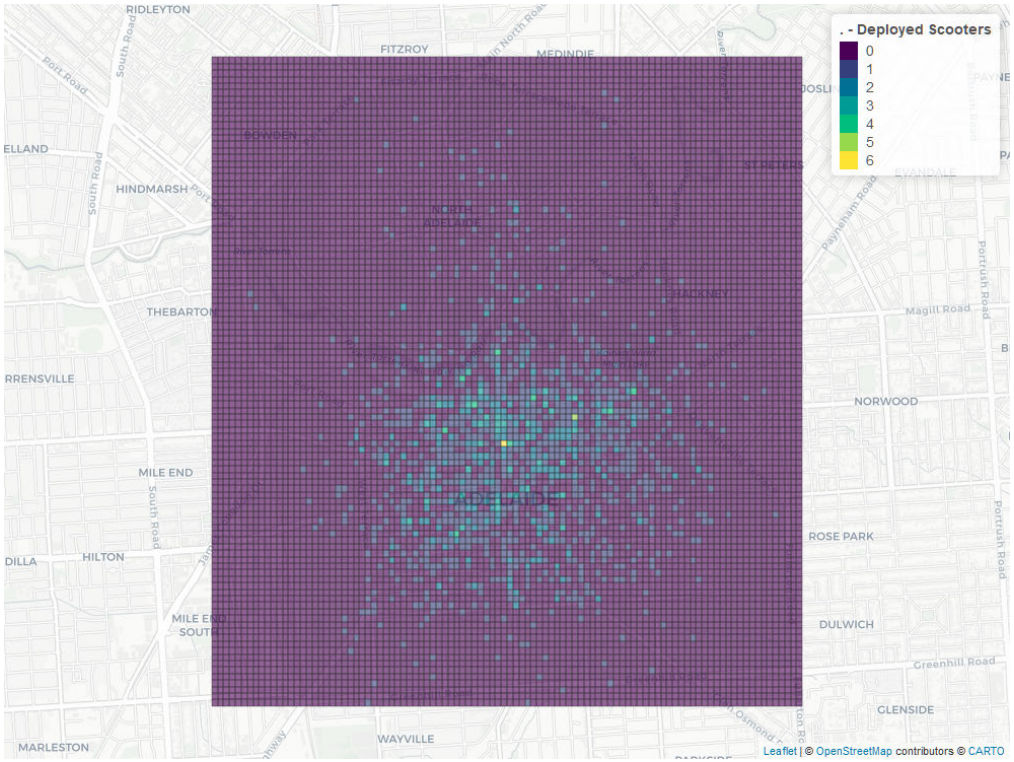
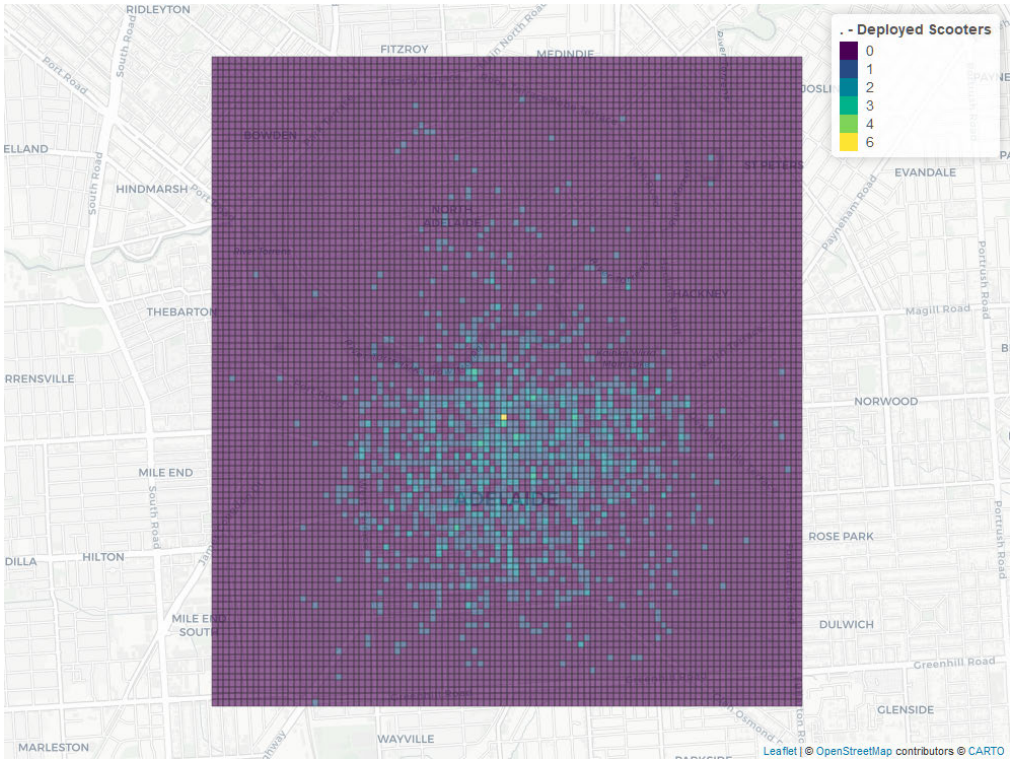


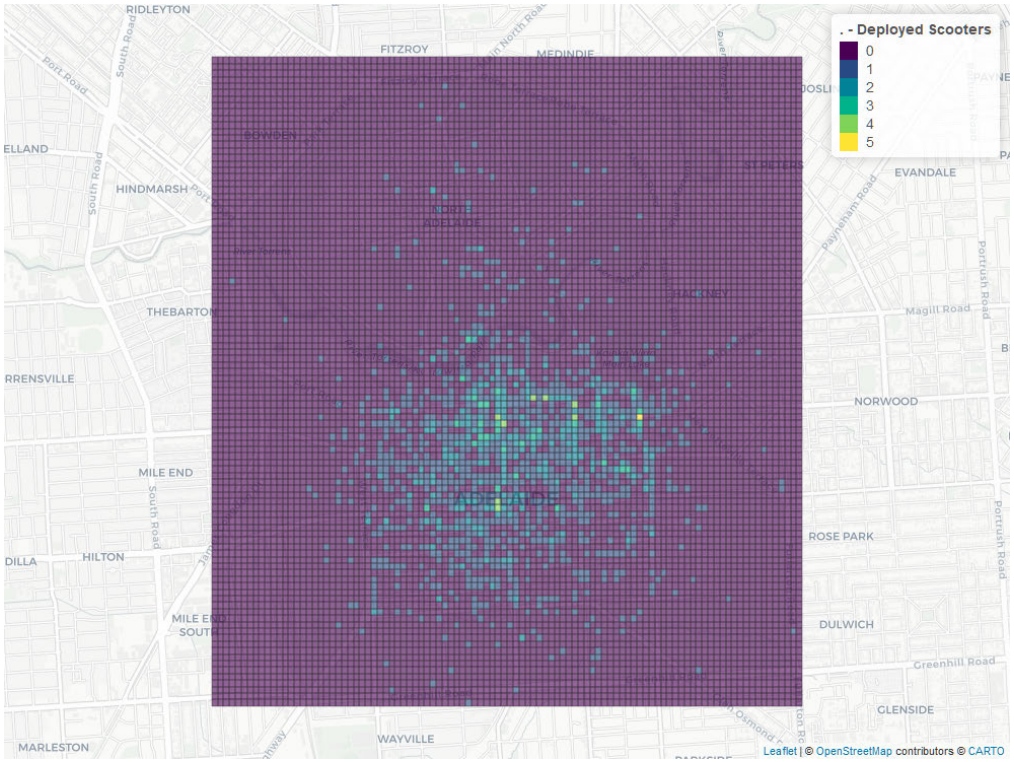
Figure 5.5: Change in expected revenue between the initial and best deployments per month. In all cases the revenue is increased by more than 130% from the initial deployment. The best improvement is shown in months 9 and 10, which have an increase in 169% and 172% respectively.



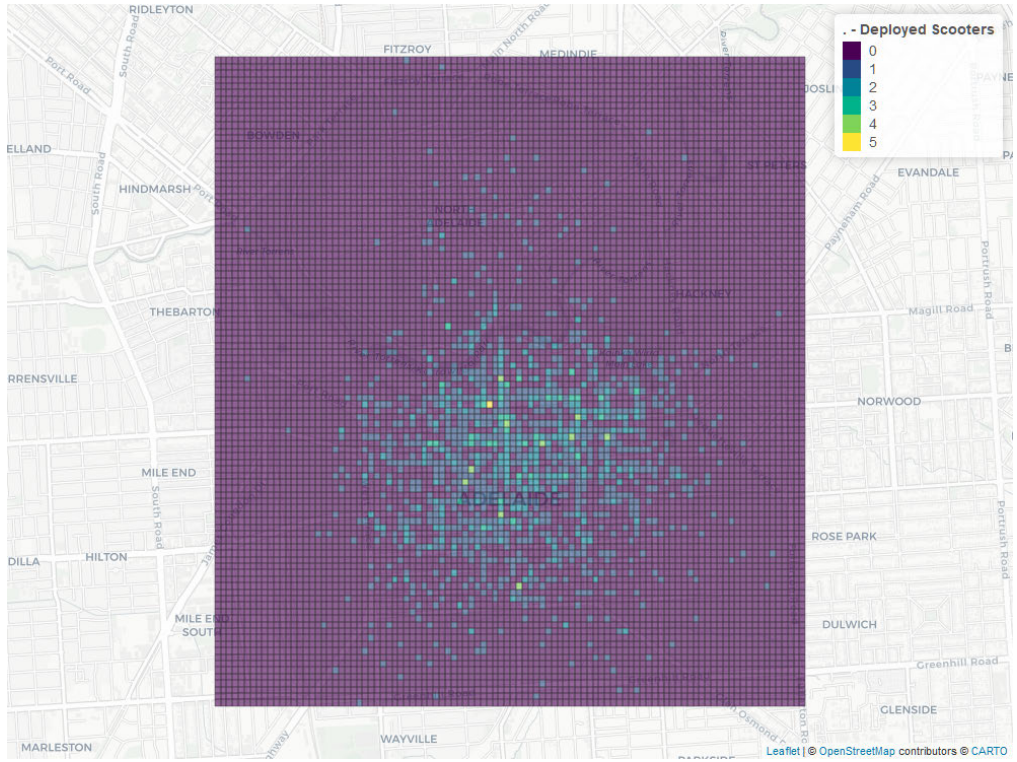
(a) Vehicle deployment for Weekend Evenings during January 2021.



(b) Vehicle deployment for Weekend Evenings during February 2021.



(c) Vehicle deployment for Weekend Evenings during March 2021.



(d) Vehicle deployment for Weekend Evenings during August 2021.

Figure 5.6: Optimal vehicle deployments for various months during 2021. All deployments are for Weekend Evenings. In each case, 100 independent simulations are used, along with up to 40,000 iterations of the simulated annealing algorithm. The deployments generally follow the demand estimates, however are more evenly spread across the entire service area.

5.4.3 Optimal Deployments by Time Category

The final results we investigate are how the optimal deployment changes over the time of day and the day of the week. We do this by applying the optimisation algorithm for all categories of time, as defined in Chapter 2. We maintain the 6-hour time window for simulations, and start the simulations at 12:00am, 6:00am, 12:00pm, and 6:00pm on both a Saturday and a Tuesday to cover all time categories. This method allows for comparison between the times of day, although the time categories do not match the cases.

To investigate the difference in the optimisation algorithm between the time categories, we again consider the change in function value as the algorithm progresses. Figure 5.7 shows the function values for all eight time categories, with EarlyMorning corresponding to the simulations starting at 12:00am, Morning at 6:00am, Afternoon (combining the Early Afternoon and Afternoon time categories) at 12:00pm, and Evening at 6:00pm. Similar patterns are apparent between the time categories as seen in the by-month optimisation, with times experiencing lower overall demand and expected revenue reaching a seemingly optimal value faster than the time categories with higher demand. However, the expected revenue does not depend solely on the overall system revenue during the time categories. The time categories with the lowest expected revenue are the EarlyMorningWeekday, EveningWeekday, and EveningWeekend categories. While the EarlyMorningWeekday demand is the lowest of all time categories, the low revenue generated by both evening time categories indicates that the trips taken during this time are much shorter than those taken during other times, in turn generating less revenue.

We also plot the change in expected revenue between the initial deployment and best deployment for each time category. As shown in Figure 5.8, we again see a large improvement in the revenue after applying simulated annealing. However, in the time categories of EarlyMorningWeekend and both Afternoon categories, we see an increase of less than 100%, whereas the minimum increase in the monthly deployments was 132%. This indicates that while an improvement can still be made by optimising the deployments for these time categories, the effect is less than the effect due to the month. Some time categories still receive a high increase in revenue, with Weekday Evenings showing a 171% increase, followed by Weekend Mornings with 161% and Weekday Mornings with a 145% increase.

Figure 5.9 shows the optimal deployments from the four times considered on a Weekday. There are slight differences in the patterns in the deployments between all times. However, due to the optimisation algorithm terminating early in the Afternoon time category, this deployment is likely not accurate, and instead is primarily based on the initial deployment. In the other three deployments, we see that the deployments are overall similar to the deployments generated previously, where vehicles are generally placed according to the demand estimates, but more evenly spread across the service area. The Evening deployment appears to be the most homogeneous across the grid, with a maximum of four vehicles placed in a small number of cells. While the Morning deployment also has a

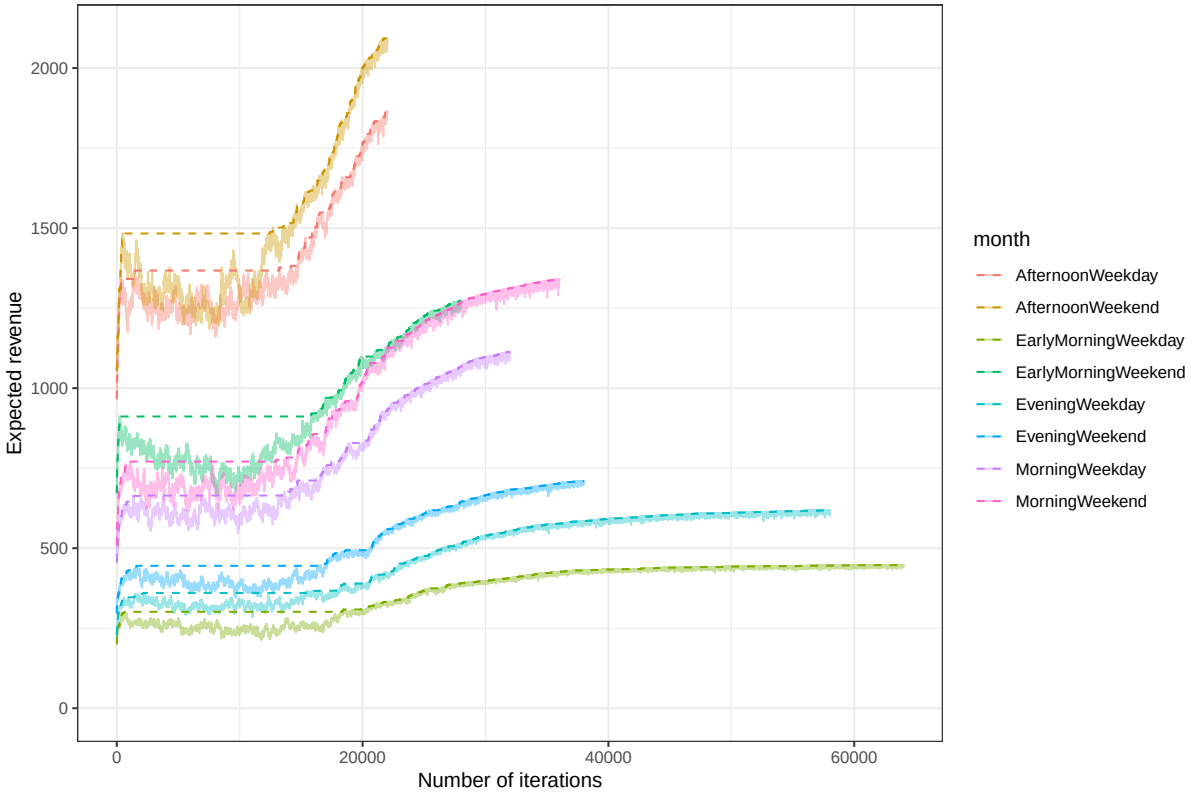


Figure 5.7: Progression of function values during the simulated annealing algorithm applied to different times of day for April 2021 demand estimates. The differences in total number of iterations are due to the algorithms terminating after a two-day runtime, which severely limits the number of iterations, especially in time categories with a large number of trips in the simulations.

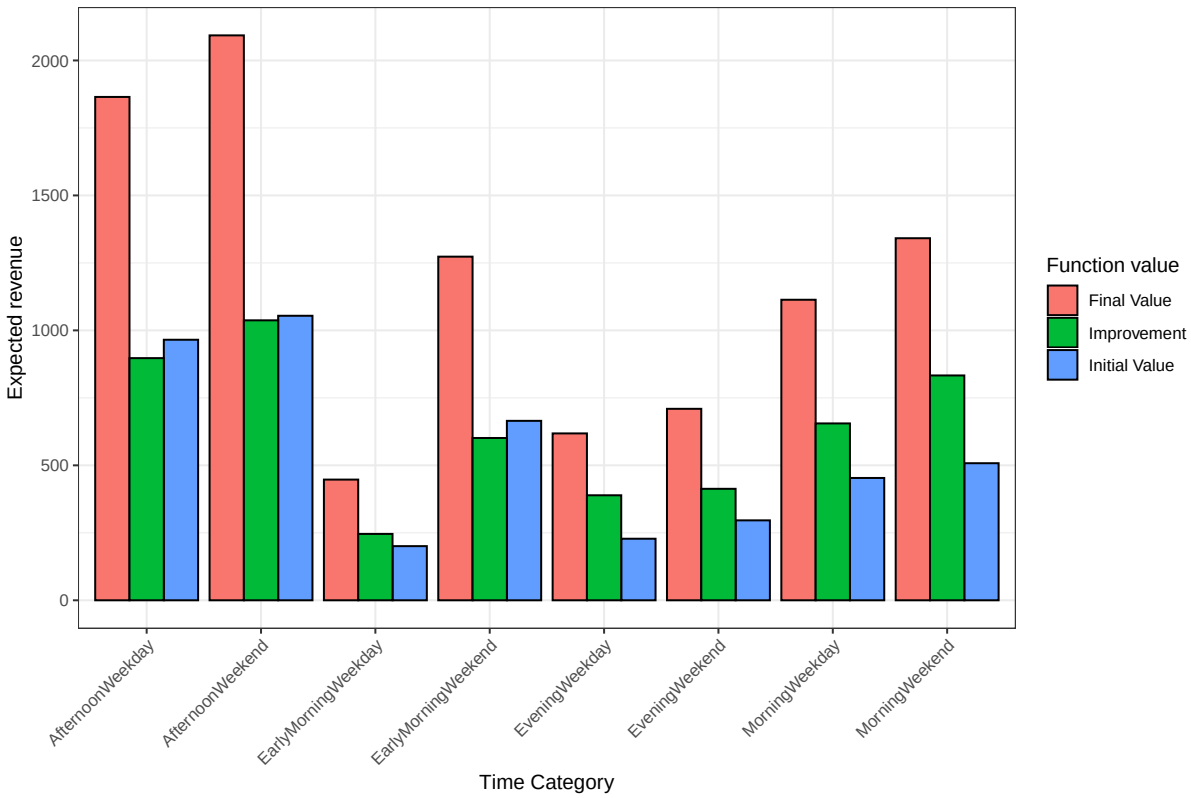
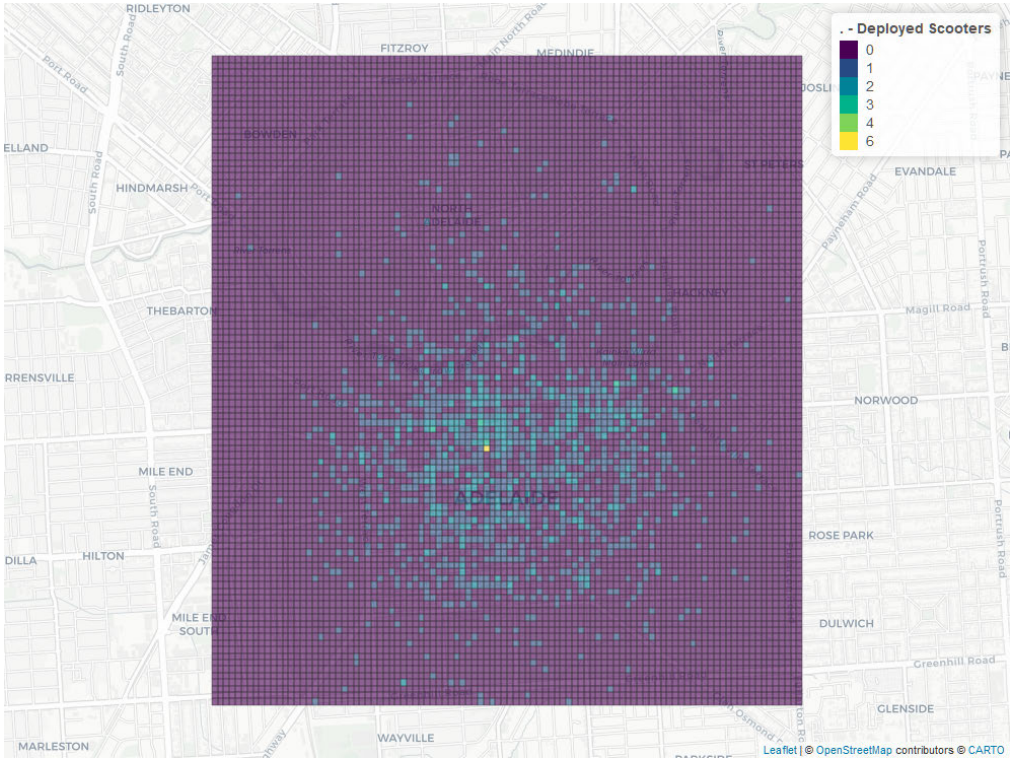
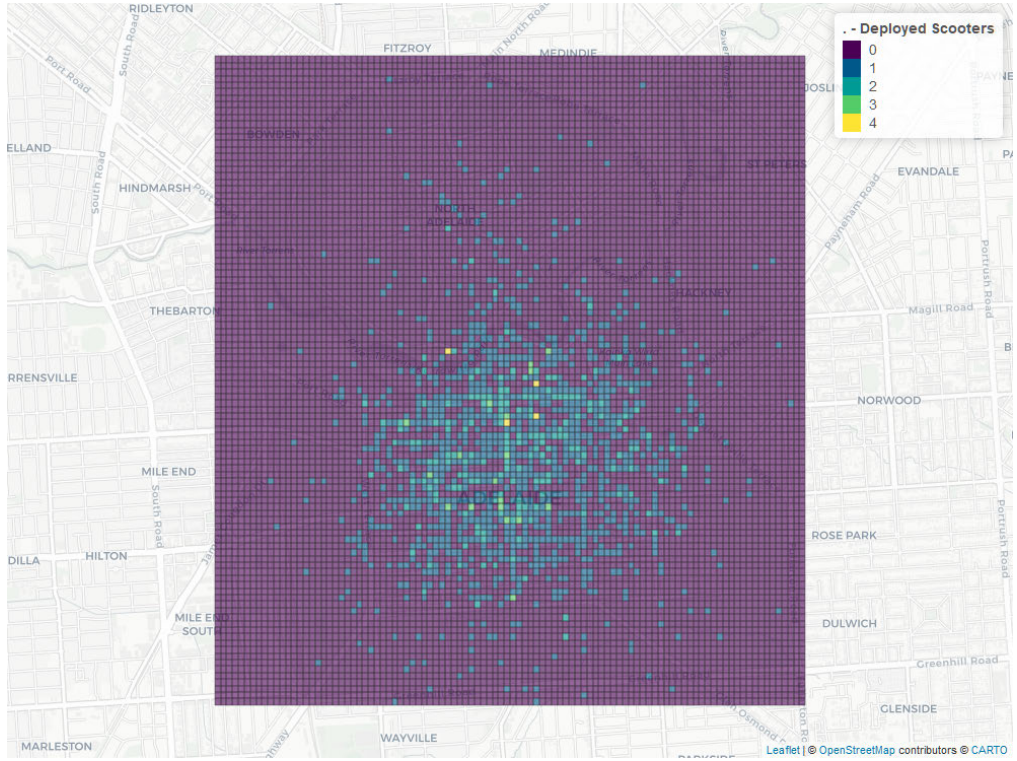


Figure 5.8: Change in expected revenue between the initial and best deployments in each time category. The best improvement is shown in the Weekday Evening time category, which has an increase of 171%, whereas the Weekday Afternoon category shows the lowest improvement at just 93%.

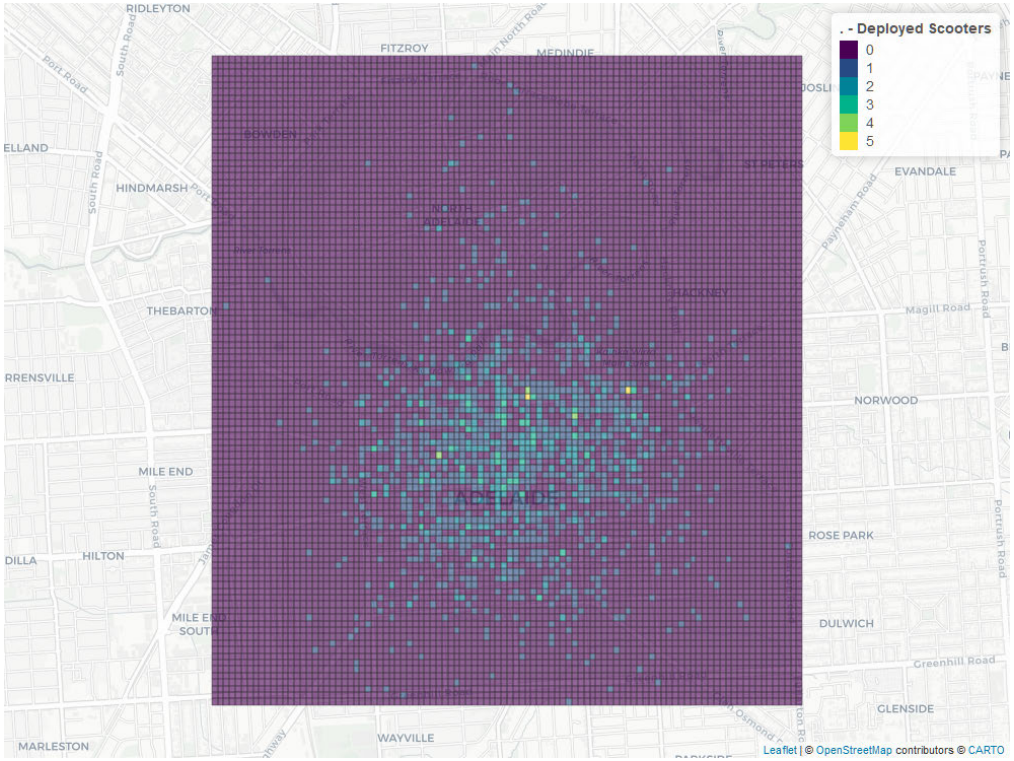
maximum of four, more cells contain four vehicles in the deployment, indicating that the deployment is more concentrated around specific hotspots. The Early Morning deployment is the most concentrated, with a maximum of six scooters allocated to a location on King William Road.



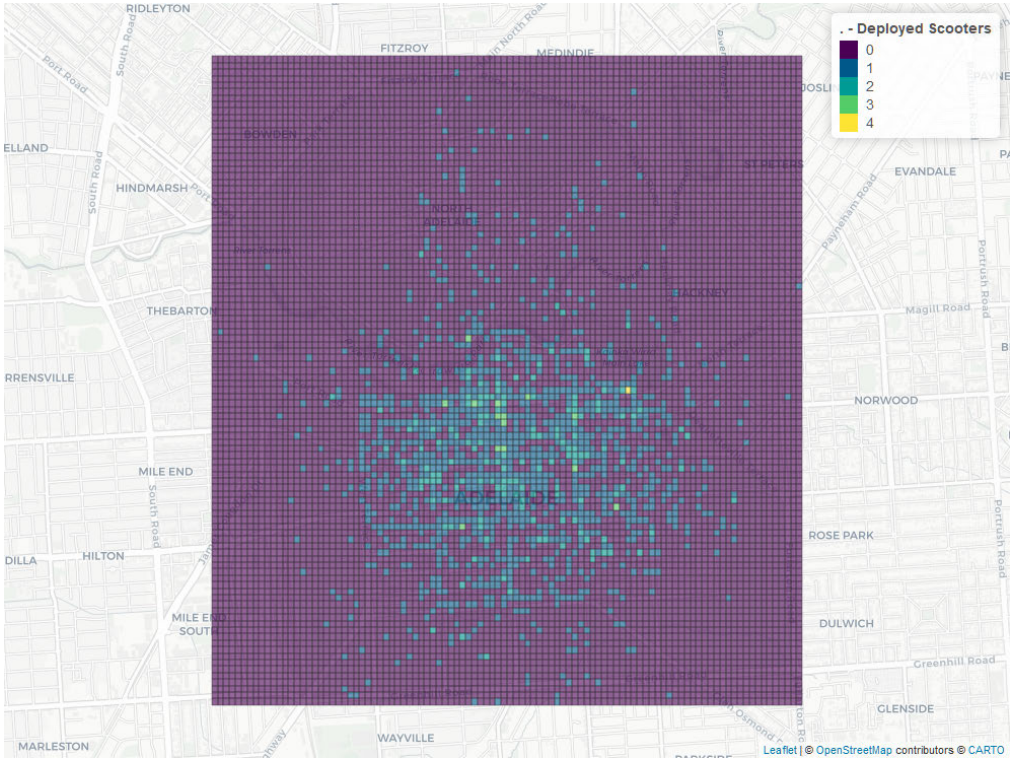
(a) Vehicle deployment for Early Mornings (12:00am-6:00am) on Weekdays during April 2021.



(b) Vehicle deployment for Mornings (6:00am-12:00pm) on Weekdays during April 2021.



(c) Vehicle deployment for Afternoons (12:00pm-6:00pm) on Weekdays during April 2021.



(d) Vehicle deployment for Evenings (6:00pm-12:00am) on Weekdays during April 2021.

Figure 5.9: Optimal vehicle deployments for times on Weekdays during April 2021. In each case, 100 independent simulations are used. Similarly to the monthly results, the deployments generally follow the demand estimates.

5.5 Discussion

In this chapter, we developed and implemented a framework for optimising the deployment of vehicles in a free-floating ride-sharing system, such as e-scooter ride-sharing systems in Adelaide. We use two algorithms to achieve this; the simulation algorithm developed in Chapter 4 provides a method of estimating the revenue generated by a deployment of vehicles, and simulated annealing is used with the estimated expected revenue to optimise the deployment of vehicles. When applied to the Adelaide e-scooter systems, we find that the optimal deployments generally follow estimates of the demand for the vehicles, however are spread out to allow more trips to occur across the entire system. This effect is observed when optimising both for each month with a fixed time, and for various times of the day for a fixed month. Similarly to the demand estimates, we see that the deployment of vehicles is more concentrated to main roads. In all cases, the deployments have a maximum of 7 vehicles deployed to any individual cell, indicating that for a 6-hour deployment time window at most 7 vehicles should be placed at one location.

There are several possible improvements to be made to the optimisation results and implementation. In a large number of cases above, we find that the algorithm has not yet converged after many iterations. Due to time constraints, we terminate the optimisation algorithm prior to convergence, once a sufficient expected revenue is reached. However, to find the true optimal deployment, many more iterations are needed. The cause for the high number of required iterations is the combination of a large solution space, and a naïve method of neighbour generation. As we are distributing 1400 vehicles across 10,000 cells, there are potentially $1400^{10,000}$ possible solutions that the algorithm may need to test, which is far more than feasible. By restricting the new solutions to being neighbours of previous solutions, we limit the scope of the algorithm to only solutions that may provide an increased expected revenue, in exchange for reducing the possible amount of change between iterations. However, the optimisation may be able to be made more efficient at determining the best deployment by introducing a bias into the neighbour generation algorithm. Examples of this include weighting cells to move to by the demand in the cell, or by increasing the probability of moving a vehicle that is poorly utilised in the current deployment. Alternatively, relaxing altogether the restriction of new solutions being neighbours of the previous may increase the rate of convergence. Convergence may also be achieved by using a different optimisation algorithm, such as the simulation-optimisation procedures described by Jian et al. (2016).

Additionally, in this chapter we consider optimising the deployment over a fixed 6-hour time window. An analysis of optimal deployments over shorter time scales, for example a one-hour time window, may yield interesting results, in particular during times of peak system usage. However, the approach of using a fixed time window may result in less applicable deployments as any system behaviour after the window is not considered. Incorporating information from an extended time window, similar to the work done by Cerruto et al. (2022), may improve the efficiency of deployments if applied to a physical

system.

In all of the above results, we use simulations to estimate the expected revenue of a deployment. By using simulations, we introduce several assumptions that may not apply to all situations. The main restriction is that we set the time window of the simulations to six hours. This means that we are optimising the deployments for only the next six hours, without accounting for any future trips requested. By changing the time window of the simulations, it is likely that the optimal deployment would change, such as by concentrating more vehicles in cells with high expected trip lengths, which therefore generate more revenue for the operator. Hence, when implementing the algorithm in order to inform a physical vehicle deployment, this parameter should be tuned to best match the time between deployments. We also only use 100 simulations for each case shown in the above results. Due to the high variability in the simulations, more simulations should be used to decrease the volatility of the optimal deployments, at the cost of additional computational time.

Incorporating a cost to moving vehicles may also improve the practicality of the deployments generated by the optimisation algorithm. Currently, the best deployment does not take into account the current location of the vehicles, so may require all vehicles to be moved, which is costly and time consuming for the operator. Adding a cost for moving vehicles would allow the existing deployment to influence the optimal redistributions. However, in order for this to be done in a useful way, data regarding the cost of moving scooters would need to be collected.

Chapter 6

Summary and Future Work

In this thesis, we performed an investigation into the patterns in usage of electric scooters in Adelaide. We then used the results to develop a digital twin of an e-scooter ride sharing system and to optimise the deployment of vehicles across the service area.

In Chapter 3 we derived an estimate for the demand for vehicles, discretised in both time (creating *time categories*) and spatial dimensions (creating *cells*), using a Bayesian framework. This model accounted for issues due to limited data, and the issue of vehicles not being present in all locations at all times. We found that the locations with the highest demand are similar between the two e-scooter systems we investigated, Beam and RIDE, with five of the top ten cells shared between the systems. The overall demand in the Beam system was much higher with around ten times higher than the estimated demand in the RIDE system. Due to the limited time window of the RIDE data along with the lower demand, we used the Beam demand estimates for the further analysis.

In Chapter 4 we presented the construction of a digital twin of an e-scooter ride sharing system. This simulation model is primarily based on the demand estimates derived in Chapter 3, with the estimated demand driving the hire process, where hires are simulated independently in each cell. Our implementation of the simulation model allows for the system to be split into two phases; the hire process, and the vehicle location process, where we track the movement of vehicles through time and across the service area. We then assign a value to each requested trip from the hire process, indicating whether the trip is successful or unsuccessful, which then allows us to calculate the revenue generated over time.

These simulations are used to estimate the expected revenue generated by an e-scooter system given a particular initial deployment of vehicles, which forms the objective function used to optimise the deployment of vehicles in Chapter 5. We use simulated annealing as it provides a relatively simple way to optimise the high-dimensional integer programming problem. To increase the efficiency of calculating the objective function in each iteration of the algorithm, we utilise the two-phase design of the simulation model by generating a set of requested trips in advance, then calculating the revenue generated by that set of trips

in every iteration. We find that the optimised deployments are generally similar across all months in the data, and across all times of day, with the deployments suggesting to place vehicles according to the demand estimates, but more spread out. There are some notable deviations from this trend, including the Early Morning optimisation indicating that a more concentrated deployment provides an increased expected revenue. This optimisation framework, combined with the simulations, would allow for the optimisation of similar free-floating ride sharing systems.

There are several avenues for future research stemming from this work. In the demand estimation, we use only 14 months of data, where the first few months likely experience lower demand due to the slow adoption of e-scooter rental in Adelaide. Using a larger data set, especially one spanning multiple years, would allow for demand trends to be analysed year-by-year, for example the effect on warmer weather in summer on the demand for e-scooters. Increasing the amount of data would also introduce the possibility of a more precise model, for example one that estimates the demand continuously in spatial dimensions such as kernel density estimates. While this increases the complexity of the analysis of demand trends, it may allow for finer detail in investigating hotspot locations across a service area.

Further improvements could also be made in the deployment optimisation methods. Firstly, simulated annealing was used as the optimisation algorithm in this thesis as it is both simple to implement and general enough to be applicable in the optimisation of e-scooters. Other algorithms, such as genetic algorithms, which were previously used by Chen et al. (2019) to optimise the placement of electric bicycle sharing stations. Additionally, within the simulated annealing framework several improvements can be made to the implementation in Chapter 5. A more sophisticated neighbour generation procedure, such as one that is more likely to move less-utilised vehicles. This would likely increase the probability of a new solution improving on the previous, especially in the later iterations of the simulated annealing algorithm where a majority of vehicles are highly utilised. Another improvement that may be made is to implement an adaptive cooling schedule rather than the exponential schedule, allowing for the current progress of the algorithm to inform the behaviour of the cooling schedule. Alternatives to traditional optimisation algorithms may also be implemented such as machine learning procedures to improve the generated deployments.

The methods described in this thesis are general enough such that they may be applied to many last-mile services. However, some features of e-scooter systems may be explored further to improve the demand and simulation models. As a mobile application is used to hire e-scooters, it may be possible to incorporate app usage data into the models by estimating the rate of missed hires, where a potential customer opens the app and is unable to locate a vehicle nearby. This can also inform the binning process, as an estimate of a reasonable search distance can be calculated. This would lead to a grid or other cell layout that reflects the behaviour of commuters using an e-scooter system.

In general, incorporating more data from an e-scooter operator, including experimental results, would likely improve the results by allowing for more accurate demand estimates or more efficient deployment optimisation.

Appendix A

Algorithm for Demand Estimation with Vehicle Availability Correction

Let:

- n_{cells} be the number of cells in the grid;
- $t_1, t_2, \ldots, t_K$ be the times at which the scooter populations are known;
- T be the total length of time spanned by the data;
- $N_k(t)$ be the scooter population in cell k at time t, and $\boldsymbol{N}(t) = (N_1(t), ..., N_{n_{cells}}(t))$, for all $k = 1, ..., n_{cells}, 0 \leq t \leq T$.

Then, the following algorithm infers the scooter populations over time, and calculates the demand estimates described in Section 3.5.2.

Algorithm 1 Calculation of Demand Estimates with Vehicle Availability Correction

Require: $t_0 < t_1 < t_2 < \ldots < t_k$ ▷ Times with known cell populations
  $T_k \leftarrow 0, k = 1, ..., n_{cells}$ ▷ Cumulative time with scooters present in each cell
  $n_k^{hires} \leftarrow 0, k = 1, ..., n_{cells}$ ▷ Cumulative number of hires in each cell
  for $j = 1, ..., k-1$ **do** ▷ For each interval between known times
    $N_k \leftarrow N_k(t_j), k = 1, ..., n_{cells}$ ▷ Initial starting populations in the interval t_j, t_{j+1}
    $s_k \leftarrow t_j, k = 1, ..., n_{cells}$ ▷ Start times of intervals with scooters present
    $P_k \leftarrow \mathbb{1}_{\{N_k(t_j)>0\}}$ ▷ Indicator for if each cell contains at least 1 scooter
    for i in status data set entries in $[t_j, t_{j+1})$ **do**
      $t \leftarrow t_i^{status}$ ▷ time of event
      $c \leftarrow c_i^{status}$ ▷ cell number of event
      type $\leftarrow$ type$_i^{status}$ ▷ type of event
      change $\leftarrow$ change$_i^{status}$ ▷ change of cell population during the event
      if $N_c = 0$ **then** ▷ If the scooter count leaves 0
        $s_c \leftarrow t$
        $P_k \leftarrow 1$
      else
        if $N_c = 1$ and change $= -1$ **then** ▷ If the scooter count hits 0
          if *type* is a user pickup **then**
            $n_c^{hires} \leftarrow n_c^{hires} + 1$
          end if
          $T_c \leftarrow T_c + (t - s_c)$
          $P_k \leftarrow 0$
        else ▷ If the cell population neither leaves nor reaches 0
          if type is a user pickup **then**
            $n_c^{hires} \leftarrow n_c^{hires} + 1$
          end if
        end if
      end if
    end for
    $\lambda_k^j \leftarrow \frac{n_k^{hires}}{T_k}, k = 1, \ldots, n_{cells}$
  end for

Bibliography

Andrenacci, N., Ragona, R. & Valenti, G. (2016), 'A demand-side approach to the optimal deployment of electric vehicle charging stations in metropolitan areas', *Applied energy* **182**, 39–46.

Baek, K., Lee, H., Chung, J.-H. & Kim, J. (2021), 'Electric scooter sharing: How do people value it as a last-mile transportation mode?', *Transportation research. Part D, Transport and environment* **90**.

Beam (2022*a*), 'Beam - Shared Electronic Scooter and Bike Rentals', `https://www.ridebeam.com/`. Accessed: 2022-02-01.

Beam (2022*b*), 'Beam Australia - Frequently Asked Questions', `https://www.ridebeam.com/faq/au-faq-escooter-groupride-universal`. Accessed: 2022-04-05.

Bierlaire, M. & Sharif Azadeh, S. (2016), 'Demand-based discrete optimization'.
URL: *http://infoscience.epfl.ch/record/217396*

Biondi, E., Boldrini, C. & Bruno, R. (2016), Optimal deployment of stations for a car sharing system with stochastic demands: A queueing theoretical perspective, *in* '2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)', IEEE, pp. 1089–1095.

Boysen, N., Fedtke, S. & Schwerdfeger, S. (2020), 'Last-mile delivery concepts: a survey from an operational research perspective', *OR Spectrum* **43**(1), 1–58.

Brandstätter, G., Kahr, M. & Leitner, M. (2017), 'Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand', *Transportation research. Part B: methodological* **104**, 17–35.

Buehler, R., Broaddus, A., Sweeney, T., Zhang, W., White, E. & Mollenhauer, M. (2021), 'Changes in Travel Behavior, Attitudes, and Preferences among E-Scooter Riders and Nonriders: First Look at Results from Pre and Post E-Scooter System Launch Surveys at Virginia Tech', *Transportation Research Record* **2675**(9).

Cerruto, G., Vassio, L., Mellia, M. & Giordano, D. (2022), 'Modelling relocation strategies for shared mobility system management'.

Chen, Z., Hu, Y., Li, J. & Wu, X. (2019), 'Optimal Deployment of Electric Bicycle Sharing Stations: Model Formulation and Solution Technique', *Networks and spatial economics* **20**(1), 99–136.

Ciociola, A., Cocca, M., Giordano, D., Vassio, L. & Mellia, M. (2020), E-Scooter Sharing: Leveraging Open Data for System Design, *in* '2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)', pp. 1–8.

City of Adelaide (2019), 'E-Scooters — City of Adelaide', `https://www.cityofadelaide.com.au/transport-parking/public-transport/e-scooters/`. Accessed: 2021-03-01. Available via: `https://web.archive.org/web/20190719030819/https://www.cityofadelaide.com.au/transport-parking/public-transport/e-scooters/`.

City of Adelaide (2022), 'E-Scooters — City of Adelaide', `https://www.cityofadelaide.com.au/transport-parking/public-transport/e-scooters/`. Accessed: 2022-07-18.

City of Norwood, Payneham & St Peters (2021), 'E-scooter trial to kick off in NPSP — City of Norwood, Payneham & St Peters', `https://www.npsp.sa.gov.au/article/view/1678`. Accessed: 2022-07-18.

City of West Torrens (2022), 'Coastal Path e-Scooters City of West Torrens', `https://www.westtorrens.sa.gov.au/Community/Lifestyle-and-inclusion/Coastal-Path-e-Scooters`. Accessed: 2022-07-18.

Correia, G. H. D. A., Jorge, D. R. & Antunes, D. M. (2014), 'The Added Value of Accounting For Users' Flexibility and Information on the Potential of a Station-Based One-Way Car-Sharing System: An Application in Lisbon, Portugal', *Journal of intelligent transportation systems* **18**(3), 299–308.

Cui, S., Ma, X., Zhang, M., Yu, B. & Yao, B. (2022), 'The parallel mobile charging service for free-floating shared electric vehicle clusters', *Transportation research. Part E, Logistics and transportation review* **160**.

Diekmann, R., Lüling, R. & Simon, J. (1993), *Applied simulated annealing*, Lecture Notes in Economics and Mathematical Systems, 396, 1st ed. 1993. edn, Springer-Verlag, Berlin.

Dong, Y., Wang, S., Li, L. & Zhang, Z. (2018), 'An empirical study on travel patterns of internet based ride-sharing', *Transportation Research Part C: Emerging Technologies*

86, 1–22.
URL: *https://www.sciencedirect.com/science/article/pii/S0968090X17302954*

Gong, D., Tang, M., Buchmeister, B. & Zhang, H. (2019), 'Solving Location Problem for Electric Vehicle Charging Stations-A Sharing Charging Model', *IEEE access* **7**, 138391–138402.

Grolemund, G. & Wickham, H. (2011), 'Dates and Times Made Easy with lubridate', *Journal of Statistical Software* **40**(3), 1–25.
URL: *https://www.jstatsoft.org/v40/i03/*

He, S. & Shin, K. G. (2020), Dynamic Flow Distribution Prediction for Urban Dockless E-Scooter Sharing Reconfiguration, *in* 'Proceedings of The Web Conference 2020', WWW '20, ACM, pp. 133–143.

Jian, N., Freund, D., Wiberg, H. M. & Henderson, S. G. (2016), Simulation optimization for a large-scale bike-sharing system, *in* '2016 Winter Simulation Conference (WSC)', IEEE, pp. 602–613.

Keane, D. & Opie, R. (2021), 'SA's COVID-19 lockdown will hit businesses hard, but the Premier is promising support', *ABC News* . Accessed: 2022-02-28.
URL: *https://www.abc.net.au/news/2021-07-20/sa-lockdown-to-impact-business-amid-covid-threat/100306692*

Lesh, M. C. (2013), Innovative Concepts in First-Last Mile Connections to Public Transportation, *in* 'Urban Public Transportation Systems 2013', American Society of Civil Engineers (ASCE), pp. 63–74.

Masoud, M., Elhenawy, M., Almannaa, M. H., Liu, S. Q., Glaser, S. & Rakotonirainy, A. (2019), 'Heuristic Approaches to Solve E-Scooter Assignment Problem', *IEEE access* **7**, 175093–175105.

Moreau, H., de Jamblinne de Meux, L., Zeller, V., D'Ans, P., Ruwet, C. & Achten, W. M. (2020), 'Dockless E-Scooter: A Green Solution for Mobility? Comparative Case Study between Dockless E-Scooters, Displaced Transport, and Personal E-Scooters', *Sustainability (Basel, Switzerland)* **12**(5).

Oudani, M. (2021), 'A Simulated Annealing Algorithm for Intermodal Transportation on Incomplete Networks', *Applied sciences* **11**(10), 4467–.

Pantuso, G. (2022), 'Exact solutions to a carsharing pricing and relocation problem under uncertainty', *Computers & operations research* **144**, 1–23.

Pinna, M., Vassio, L., Giordano, D. & Mellia, M. (2022), 'Demand Model Generation from Traces: Adaptive KDE Data-Driven Optimization'.

Poulson, J., Mabin, S. & Schmidt, A. (2020), 'As tough SA COVID-19 lockdown begins, travellers report 'sea of cars' on outback highway into NSW', *ABC Broken Hill* . Accessed: 2022-02-28.
URL: *https://www.abc.net.au/news/2020-11-19/rush-into-nsw-as-sa-covid-lockdown-starts/12899612*

R Core Team and contributors worldwide (2021), 'R: K-means Clustering', `https://search.r-project.org/R/refmans/stats/html/kmeans.html`. Accessed: 2021-12-14.

RIDE (2020), 'RIDE Scooters — Let's Ride!', `https://ride.co/`. Accessed: 2022-02-01.

Sanders, R. L., Branion-Calles, M. & Nelson, T. A. (2020), 'To scoot or not to scoot: Findings from a recent survey about the benefits and barriers of using E-scooters for riders and non-riders', *Transportation research. Part A, Policy and practice* **139**, 217–227.

Schmöller, S., Weikl, S., Müller, J. & Bogenberger, K. (2015), 'Empirical analysis of free-floating carsharing usage: The Munich and Berlin case', *Transportation research. Part C, Emerging technologies* **56**, 34–51.

Schneider, J. & Kirkpatrick, S. (2006), *Stochastic Optimization*, Scientific Computation, 1st ed. edn, Springer Berlin Heidelberg.

Shaheen, S. & Chan, N. (2016), 'Mobility and the Sharing Economy: Potential to Facilitate the First- and Last-Mile Public Transit Connections', *Built environment* **42**(4), 573–588.

Shu, J., Chou, M. C., Liu, Q., Teo, C.-P. & Wang, I.-L. (2013), 'Models for Effective Deployment and Redistribution of Bicycles Within Public Bicycle-Sharing Systems', *Operations research* **61**(6), 1346–1359.

Sutton, M. (2021), 'Adelaide Fringe 2021 set to open as organisers 'pivot' to manage COVID-19 woes', *ABC Radio Adelaide* . Accessed: 2022-02-28.
URL: *https://www.abc.net.au/news/2021-02-18/adelaide-fringe-2021-to-kick-off-despite-covid-19/13165396*

Wang, H. (2019), 'Routing and Scheduling for a Last-Mile Transportation System', *Transportation science* **53**(1), 131–147.

Weikl, S. & Bogenberger, K. (2015), 'A practice-ready relocation model for free-floating carsharing systems with electric vehicles - Mesoscopic approach and field trial results', *Transportation research. Part C, Emerging technologies* **57**, 206–223.

Yu, S., Liu, G. & Yin, C. (2021), 'Understanding spatial-temporal travel demand of free-floating bike sharing connecting with metro stations', *Sustainable cities and society* **74**.