Master in Management + Analytics

Let's shuffle: Facility Optimal Location for Stations within Bicycle Sharing Systems in the City of Buenos Aires after the pandemic

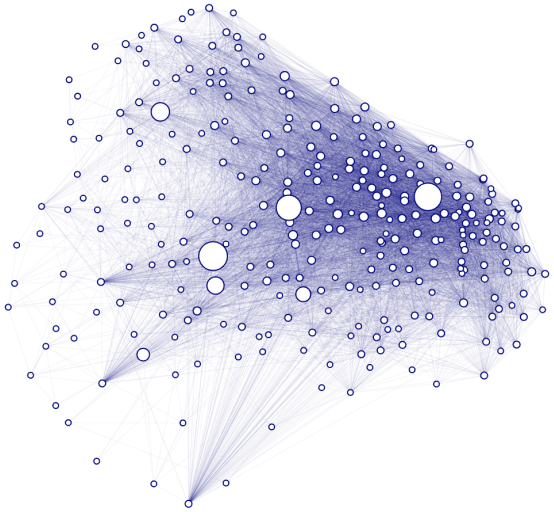
Autoría: Cruces, Nicolás
Fecha de defensa de la tesis: 2023

Universidad Torcuato Di Tella

Master in Management & Analytics Thesis

Let's shuffle: Facility Optimal Location for Stations within Bicycle Sharing Systems in the City of Buenos Aires after the pandemic

---



Author:
Nicolas Cruces

Advisors:
Juan José Miranda Bront & Nicolás García Aramouni

Table of Contents

Index of Tables

Index of Figures

Abstract

People's habits have changed after the pandemic and cycling around the city of Buenos Aires is no exception. This thesis leverages literature on Capacitated Facility Location Problems (CFLP) to build an optimal bike-sharing network to minimize the total system's cost. The objective is to decide which stations should be left open to meet projected demand in the worst-possible cases, ensuring that users do not have to walk more than a predefined distance to the facility that is closest to them. Results suggest that there is an excess of stations in the downtown area and idle capacity that could be relocated in peripheral areas, reflected by a positive load factor increase of 2x after the optimization is done. The solution shows that up to 70% of total costs could be saved after using our optimization model, by closing down facilities while meeting demand. While total cost is estimated as the budget that needs to be invested to ramp up the system from scratch, it is a useful metric that shows us how the network could be optimized taking away stations from overcrowded areas without losing any of the current demand. All of these bike-sharing facilities could be relocated to areas that have a low-density of bikes, improving access to the cycling system in the city of Buenos Aires.

A Momeh, El Abuelo, Dijon, Pupa, La Bobe. No estaría acá sin ustedes.

1. Introduction

1.1 Context, Motivation, Problem Statement

Cycling is an activity that has increased in popularity in recent years: bicycles are a healthy, emission-free means of transportation that are cheaper than cars. However, not all people in the city of Buenos Aires, Argentina, have the luxury of owning a bicycle. Having this in mind, the city government introduced the *Ecobicis* program in 2012, a bike sharing mechanism that sought to provide a new alternative in public transportation systems.

As with any bike sharing system, the administration installed bike stations in different spots within the city, stocking them with bicycles that people could take out for an indefinite period of time only to later return them to another station. The program continued to be maintained by the city government until May 2019, when Ecobicis was sold to Tembici, a Brazilian ridesharing company. The acquisition had to do with ensuring that private capital would support the network, while the Government of the City guaranteed its gratuity by contributing 60M$ in subsidies to the bike-sharing company. The subsidy meant a huge saving for the city, since it only represented half of what they spent to maintain the system at the time.[1]

Since its acquisition, Tembici has expanded coverage of the program within the city. They installed multiple stations outside of the typical tourist spots and observed a mass increase in usage of 3.6x[2]. The pandemic put their growth to a stop, with severe quarantine measures in place in Argentina. The service was completely suspended from March 19th, 2020 until May 12th, 2020, and when it came back online, the total number of daily trips was not at the level it used to be, as seen in Figure 1.1.

[1] Extracted from:
https://www.cronista.com/apertura/empresas/Adios-a-las-bicicletas-amarillas-una-firma-brasilena-operara-Ecobici-20190219-0010.html

[2] Value calculated by comparing average number of bicycle trips before Tembici acquisition (March-May 2019) and after Tembici acquisition (May 2019-Dec 2019).

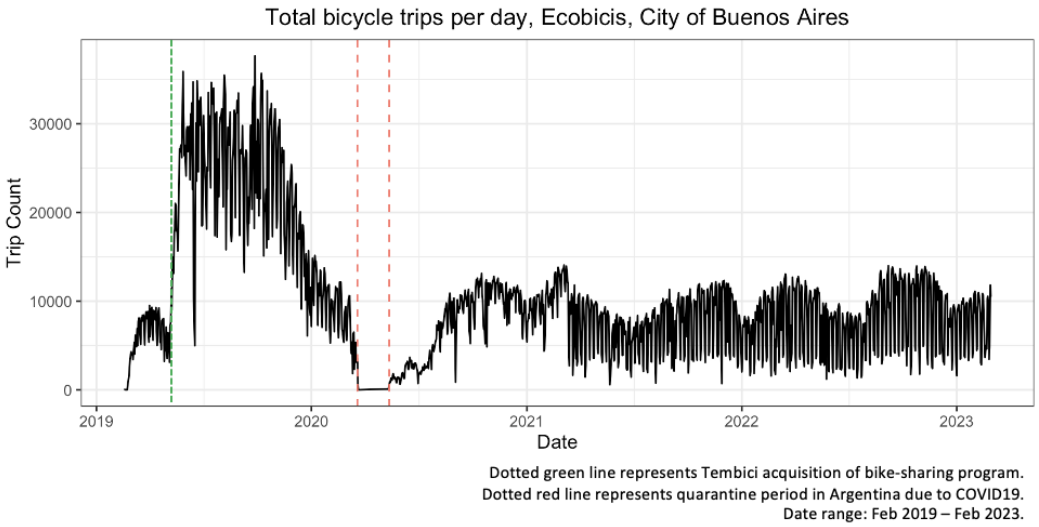


Figure 1.1: Total number of trips per day considering all stations that were active during that day, ranging from Feb 2019 to Feb 2023. Plot was built using information from bike-sharing user trips from Feb 2019 to Feb 2023 in the City of Buenos Aires.[3] Plot was built by the author.

What motivates this thesis is the hypothesis that states that, after the quarantine, people in the city have changed their cycling habits. Working remotely from home, city life does not revolve so much around downtown, where most offices were located. Hence, there is an over investment of stations and retrievable bikes in the downtown area and an under investment along the city's periphery, where people spend a larger relative share of their time, as seen by the current network of stations in the city in Figure 1.2.

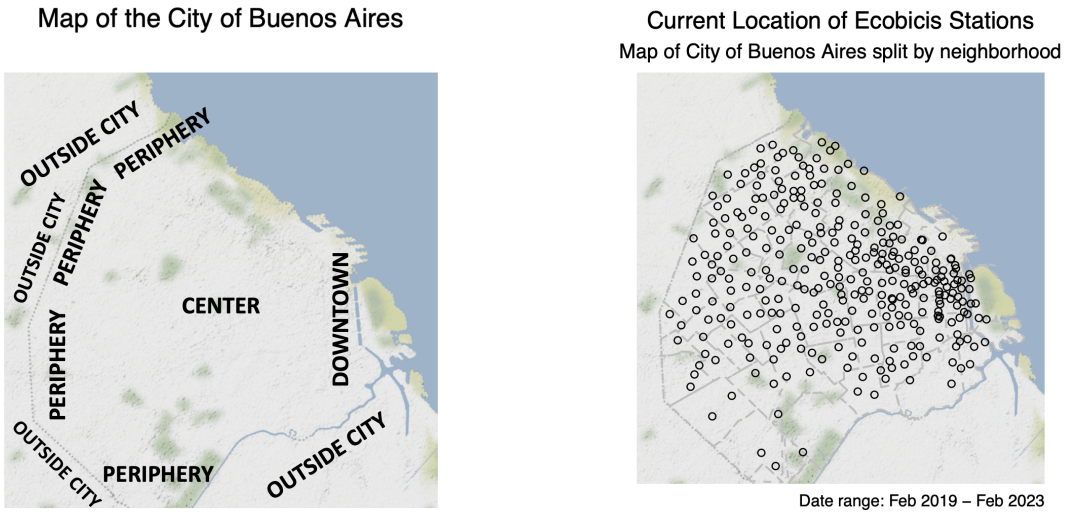


Figure 1.2: Current location of stations within the City of Buenos Aires, last 6 months of data (sept 2022-feb 2023).

[3] Link to extracted information: https://data.buenosaires.gob.ar/dataset/bicicletas-publicas

Our objective is to study the optimal design of the bicycle station network under different conditions, aiming to minimize the system's total cost. Given that we already have the current station configuration, by using different simulation techniques and assumptions, we can estimate the current system's cost and use it to understand the improvement our optimization model can bring. Our solution will involve Linear Integer Programming and can be framed within a subset called Capacitated Facility Location Problems (CFLP). Such types of problems require inputs that depend on where stations are located and where demand is located. Therefore, we will also use the centroids of Voronoi regions around each station to calculate demand locations. We will also predict demands for each center of demand using state of the art time series techniques, ensuring that our predictions reflect worst-case scenarios, making our solution more robust. Finally, we run a ceteris paribus analysis to determine how each of the parameters affect our proposed end result.

1.2 Literature Review

1.2.1 Theory On CFLP

Wolsey (2021) [12] describe Integer Linear Programming (ILP) as a mathematical optimization problem in which some or all of the variables are restricted to be integers. The Linearity in the ILP usually refers to a linear objective function. CFLP's are a typical type of Integer Linear Programming problems.
Snyder and Shen (2019) [10] have a book that does a great introduction to CFLP's. Paraphrasing them, this problem comes from firms that need to decide what is the number and location of factories, retailers or physical facilities. There is a key tradeoff between the quantity of facilities and the customer service they provide. Too many facilities probably imply excellent customer service since most customers are close to a facility, but also make the firm incur in high facility costs to build and maintain them. On the other hand, if the firm installs too few facilities, they will have lower costs, but customers will have to travel great distances to get to a facility.

In the CFLP problem, facilities have fixed costs that represent building each facility, which are independent of the amount of volume that passes through the facility. There is also a transportation cost per unit of product shipped from a facility to a customer. Each facility also has a maximum capacity of product that they can store and there is a single product. The problem is to choose facility locations to minimize the fixed cost of building facilities plus the transportation cost to transport product from facilities to customers, subject to constraints requiring every customer to be served by some open facility.

The authors also go into different types of CLFPs, with other objective functions that we will not cover throughout this thesis. Covering models are worth mentioning, given the nature of the problem we will address here. Our firm could desire that all users are very close to a bike-sharing station and not have to travel far, regardless of the installation cost that this implies. The idea here is that we minimize the amount of stations that are open, while ensuring that every customer will be allocated to a station, without being left out of the system.

While our objective function will minimize total costs, we will introduce restrictions that guarantee coverage for the current level of demand. We will also include restrictions that make solutions practically feasible, meaning that a user does not get allocated to a station that is very far away and to which they would not travel to.

1.2.2 Practical applications of CFLP

Much has been written about these types of problems. Essentially, bike sharing systems depend on optimal decisions in three different types of fronts, some of them more strategic and others more operational. Paraphrasing Nikiforiadis, Aifadopoulou, Salanova Grau, and Boufidis (2020) [8], the three optimization problems to consider are:

- The location and number of bike-sharing stations within the service area.
- The capacity of each station and the number of bicycles available in the system.
- The redistribution process of the bicycles within the day or a given time period.

These problems are sorted from most strategic to more operational. We will discuss all of them briefly, below.

1.2.2.1 Capacity of each station, number of bicycles and redistribution process problems

Freund, Henderson, O'Mahony, & Shmoys (2019) [3] discusses the work they did with *Motivate*, the leading company in the bike-sharing industry in the United States. Their first project had to do with reallocating capacity across stations, based on the station's current utilization, with the end goal of improving the service quality for riders. The second of their projects, described in the same paper, have to do with building the right incentives for users in New York City to make them rebalance the system without any intervention from the owner.

User Dissatisfaction Functions (UDF) are the framework behind this paper. The idea is to associate the number of docks and bikes of each station to the expected number of stockouts that they will have. Each UDF follows a stochastic sampling process of users returning or renting bikes. A dissatisfied user is recorded every time that a user in the sampled sequence wants to rent a bike and no bikes are available, or conversely, when they would like to deposit a bicycle and all docks are full. Then, the UDF is defined as the expected number of users that were dissatisfied from the sample, based on the station capacity and bikes available. In order to avoid having biased estimates of demand, since no foreign actor was intervening in the system at the time their investigation took place, they developed a decensoring method that estimates time-dependent demand for arrivals and returns at each location.

Another paper that works on the tactical Bicycle Redistribution Problem (BRP) is Dell'Amico, Hadjicostantinou, Iori, and Novellani (2013) [2]. Their objective is to decide how the vehicles that replace bicycles in each station should be routed so as to minimize the vehicles total cost. A typical example of when this problem becomes of paramount importance is in cities with hills, where users decide to take a bike from up the hill and deposit it in the station below the hill, but find other means of transport to go back up the hill again. Therefore, you have stations that

have a lot of bikes and stations that run out of bikes, hence the need for a redistribution operator. The authors start by formulating a Mixed Integer Linear Programming version of the BRP, based off of the Multiple Traveling Salesman Problem, where uncapacitated vehicles based out of a central depot have to visit a set of bike-sharing stations, with the constraint that each station is visited exactly once. In order to adapt the typical Salesman Problem to the nature of the BRP, they include additional constraints so as to ensure that demands are met and vehicle capacities are not exceeded. They build different models and test them by collecting data from several bike-sharing websites, and benchmark every one of them by their computational run time.

Finally, Ohana (2021) [9] is also worth mentioning, given that they use a similar dataset to the one being leveraged for this thesis. Their thesis focuses on understanding what is the optimal amount of bikes that should be allocated to each station at the beginning of each day, minimizing the chances of the system not having stock. It estimates demand using a model that incorporates climate variables such as weather, temperature to be able to decompose the estimation between trend and seasonality. If we were to frame it within the three types of decisions from Nikiforiadis [8], this body of work would fit into the following: “define the redistribution process of the bicycles within the day or a given time period”. In an opposite lane but using the same framework, our thesis will focus on a more strategic decision: “define the location and number of bike-sharing stations within the service area”.

1.2.2.2 Location and number of bike-sharing stations problem

This thesis addresses the first of the problems described in Nikiforiadis, Aifadopoulou, Salanova Grau, and Boufidis (2020) [8]: define the location and number of bike-sharing stations in the area of service. There are pre-existing works that have already solved this object of study in different cities around the world, with diverse methods to estimate potential demand.

Nikiforiadis, Aifadopoulou, Salanova Grau, and Boufidis (2020) [8] tackle this topic in the city of Thessaloniki, Greece by constructing an objective function with three sub-objectives that are weighted differently. Said objective function maximizes the amount of demand that is met and the geographical coverage that the system has over the city while minimizing the bicycle redistribution needs. They determine what the potential demand for a new station would be by using a pre-existing bike-sharing system where people could rent a bike without having to go to a specific station. Then, they split the city into quadrants and define their centers of demand as the quadrant centroid and use it to estimate arrival and departure rates.

Martinez, Caetano, Eiro, & Cruz (2012) [7] solved the design of a bike-sharing system in the city of Lisbon, Portugal. Their objective function is net benefit, defined as the difference between income minus bicycle deposit cost and total fleet cost, with the decision variables being the quantity of stations, their location, and the stock of bicycles. Beyond solving the problem for different price configurations, the main contribution of this paper is to incorporate uncertainty into potential demand, with a model of simulated synthetic trips, calibrated for the metropolitan area of this city. With this dataset, they estimate the propensity to make this trip by bicycle with a discrete choice model.

Gonzalez (2017) [5] addresses this problem in the city of Buenos Aires by minimizing total commuting time to work, playing with the pre-existing public transportation network and thinking of bicycles as a complement. He computes areas that are surrounding subway stations and are more than four blocks away and defines them as bikeable but not walkable distances. Within these zones, he uses a GIS location-allocation algorithm incorporating variables such as the inhabitants of each census block, the time each census block takes to reach the center, and the average cost of the trip. He also leverages census parameters as an input for the algorithm to estimate potential demand.

García-Palomares, Gutiérrez, and Latorre (2012) [4] also aims to define the location that stations should occupy in the city of Madrid using a location-allocation model in GIS, but they employ another method to estimate the potential demand. The authors begin by creating a layer of points containing the population and employment associated with each point and a layer of polygons that includes the number of trips that originate and terminate from each transportation zone. To obtain the number of trips for each point, they multiply the ratio of trips generated by the transportation zone by the number of inhabitants in that point. By calculating the average trips per job of each point, they are able to detect zones that represent office locations and adapt their model to incorporate this information.

Liu et al. (2015) [6] determines the optimal location of stations in the city of New York by predicting the demand and balance of the system, incorporating information on people's mobility and proximity to key points as features. They define a Voronoi region around each station, assuming that the centroid of the Voronoi region is equal to the center of demand. For each of these regions, they estimate the demand for the area around a station using a neural network, using the variables described as input. Then, they use a genetic algorithm to select the optimal location of stations out of a randomly defined set of candidate points.

This thesis will leverage the pre-existing literature and will adapt it to the nature of the city of Buenos Aires. This thesis is divided as follows. Section 2 gets into a data review, talking about the sources that the data came from and the exploratory data analysis. Section 3 describes the model being used, the inputs that are necessary and how they are being estimated. Section 4 solves the problem using the problem instance that we define as ideal and walks us through a sensitivity analysis for all the input variables. Finally, Section 5 includes conclusions and potential for future work.

2. Data

2.1 Data Sources, Feature Engineering

There are three main datasets that were used for this project. The first one of them is the same one used by Ohana [9] and holds all of the bicycle trips done by users of the bike-sharing system from Feb 1st 2019 to Jan 31st 2023. These were extracted from the Gov. of the City of Buenos Aires's official webpage[4]. This dataset holds the following information. We will use this as the base for predicting demand per active station, understanding where those stations are located.

Column Name	Description	Variable type
id_recorrido	Unique identifier of trip.	String
duracion_recorrido	Duration of trip in seconds	Integer
Fecha_recorrido (origen/destino)	Timestamp of when the trip occurred. Records when the trip started (origen) and when the trip ended (destino).	Timestamp
Id_estacion (origen/destino)	Unique identifier of bike-sharing station. Records where the trip started (origen) and where the trip ended (destino).	String
Nombre_estacion (origen/recorrido)	Name of the bike-sharing station. Records where the trip started (origen) and where the trip ended (destino).	String
Direccion_estacion (origen/recorrido)	Address of the bike-sharing station. Records where the trip started (origen) and where the trip ended (destino).	String
Long_estacion (origen/destino)	Longitude of the bike-sharing station. Records where the trip started (origen) and where the trip ended (destino).	Integer
Lat_estacion (origen/destino)	Latitude of the bike-sharing station. Records where the trip started (origen) and where the trip ended (destino).	Integer
id_usuario	Unique identifier of user that did the bicycle trip.	String

Table 2.1: Dataset description for bike-sharing trips data extracted from the Gov. of the City of Buenos Aires's official webpage.

The second dataset that was used came from consuming the available Public Transport API also provided by the Gov. of the City of Buenos Aires[5]. There are three endpoints that hold different information per station and that provide the latest up to date information about the system. This dataset will help us associate a capacity to each of the active stations mentioned in the previous dataset, a key part of our optimization problem that will help us understand how much demand each station can meet.

We managed to ping them at regular hourly time intervals for a month, from Nov 27th 2022 to Dec 25th 2022. API query results were saved in a json file that was appended to previous API

[4] Link to extracted information: https://data.buenosaires.gob.ar/dataset/bicicletas-publicas
[5] Link to Public Transport API docs: https://apitransporte.buenosaires.gob.ar/console/

results to accumulate all of them in a single place, and were later exported to a document .csv to be able to merge with other datasets. The information that was later used from these files is described in Table 2.2. Further details on how the API was consumed can be found in the Appendix section.

Column Name	Description	Variable type
station_id	Unique identifier of bike-sharing station.	String
lat	Latitude of the bike-sharing station.	Integer
lon	Longitude of the bike-sharing station.	Integer
capacity	Total capacity of bike-sharing station. This shows how many bicycles the station could potentially hold.	Integer
num_bikes_available	Quantity of bicycles available to use at a certain point in time in the bike-sharing station.	Integer
num_docks_available	Quantity of docks available to leave a bicycle at a certain point in time in the bike-sharing station.	Integer
last_updated	Timestamp in which the API was consumed.	Unix timestamp

Table 2.2: Dataset description for station and bike-sharing system information data extracted from the Public Transport API from the Gov. of the City of Buenos Aires.

The final dataset that was leveraged for this project was the date and name of holidays in Argentina, from 2019 up to 2023[6]. We constructed this dataset based off of the official calendar of holidays form that is published every year by the Argentinian government. This dataset will play a key role when predicting demand, since it will help our model detect outliers.

We also built additional date and time features to make sure we are extracting the largest amount of information from the dataset we can. These new variables are listed in Table 2.3.

[6] Link to example of Argentina National Holiday calendar for 2021: https://www.argentina.gob.ar/interior/feriados-nacionales-2021

Column Name	Description	Variable type
Fecha_recorrido_hora_dia (origen/destino)	Hour of day when the trip occurred. Records when the trip started (origen) and when the trip ended (destino).	Integer (in ART)
Fecha_recorrido_dia (origen/destino)	Date when the trip occurred. Records when the trip started (origen) and when the trip ended (destino).	Date
Fecha_recorrido_mes (origen/destino)	Month when the trip occurred. Records when the trip started (origen) and when the trip ended (destino). Fill in day with 1st day of each month.	Date
Fecha_recorrido_year (origen/destino)	Year when the trip occurred. Records when the trip started (origen) and when the trip ended (destino). Filled in month and date with Jan 1st.	Date
weekday	Day of week when the trip occurred.	String
is_weekend	Dummy variable representing if the day when the trip occurred is part of the work week or the weekend.	String
count_ids	Count of bicycle trips with duplicated id_recorrido. Trips that have multiple trip ids (id_recorrido) will be discarded.	Integer

Table 2.3: Features that were inferred based off of the datasets that were described in Tables 2.1 & 2.2.

Given the clear structural change in the dataset exhibited in Figure 1.1 after the COVID19 quarantine from March to May 2020, we will only use data from May 2020 onwards since it will be more representative of the current scenario. This implies that we will have a total of ~8.2 million observations of trips recorded in this period, with 28 descriptive variables coming from the trips dataset. The datasets that were extracted from the Public Transport API represent an extra 31.5 K observations with 7 variables, which will be merged with the trips dataset file to run the analysis.

2.2 Data Preprocessing

The second step of every data analysis project has to do with making sure that the data makes sense and cleaning it to ensure we are providing insights to inform accurate decisions.

We started by counting the amount of trips per day, and considered a day of activity as one that had at least 100 trips. We chose the 100 trips value because, as seen in Figure 1.1, the average of trips per day after the pandemic is around 5000 trips per day. However, there are also some dates that are holidays and where people did not cycle as much as they usually do. We did not want to count these days as inactive, since there were bicycle rides that occurred. Hence, the 100 trip threshold excludes days that have close to zero trips, and counts as active days with a low amount of trips but still had some. Then, we counted the amount of active days within a month, to determine which month had missing days. We divide that by the amount of days in the month, to make sure we translate this value to a percentage.

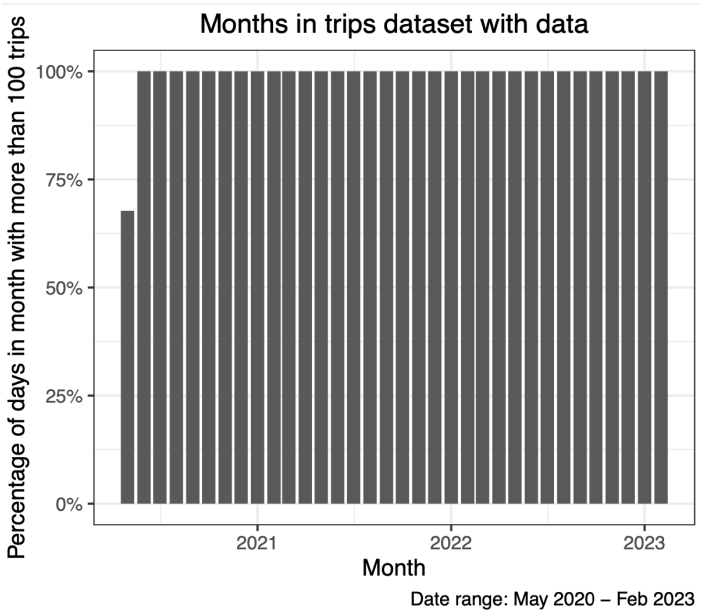


Figure 2.1: Percentage of days in total month that had more than 100 trips per day. Date range: May 2020 - Feb 2023.

As Figure 2.1 shows, we have fewer days than the days in the entire month for May 2020. This is expected given that the bike-sharing system was shut down until May 12th. The rest of the months are at 100% of coverage meaning that there are no clear holes in the dataset when we look at it by date of bike trip.

We also sought to determine the quantity of trips that were duplicated based on the trip unique identifier ("id_recorrido"). Given that each trip should have a single identifier, it makes sense to discard trips that are duplicated in the dataset, given that they could bias results.

Count of repeated ids per trip	Count of trips	% of total observations
1	8,222,296	99.8%
2	7,862	0.1%
4	388	0.005%

Table 2.4: Count of trips that had repeated unique identifiers and percentage of total observations that they represent. Date range: May 2020 - Feb 2023.

As observed in Table 2.4, trips with repeated ids account for 0.2% of total observations. These are discarded from the dataset, given that the data loss is negligible.

Furthermore, we looked at the distribution of trip time, split by weekday or weekend. We compare how the mean and quartile of the distributions looks with the entire dataset, on top, and how the distribution looks when outliers are excluded up to 1%, on the bottom.

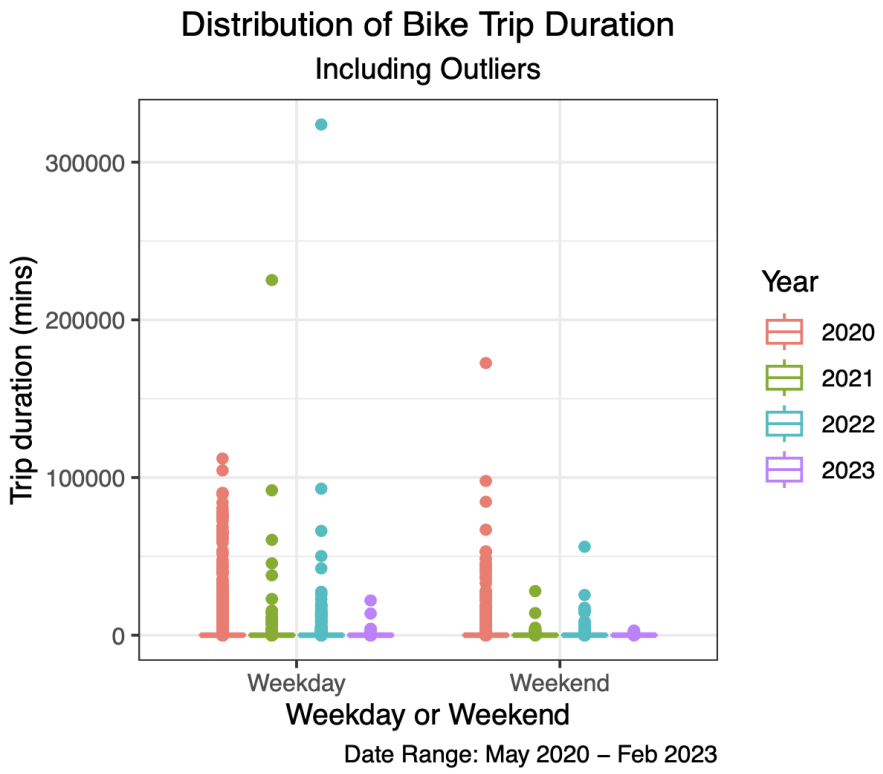


Figure 2.2: Boxplots that shows distribution of trip time during the week and during the weekend. Outliers included.

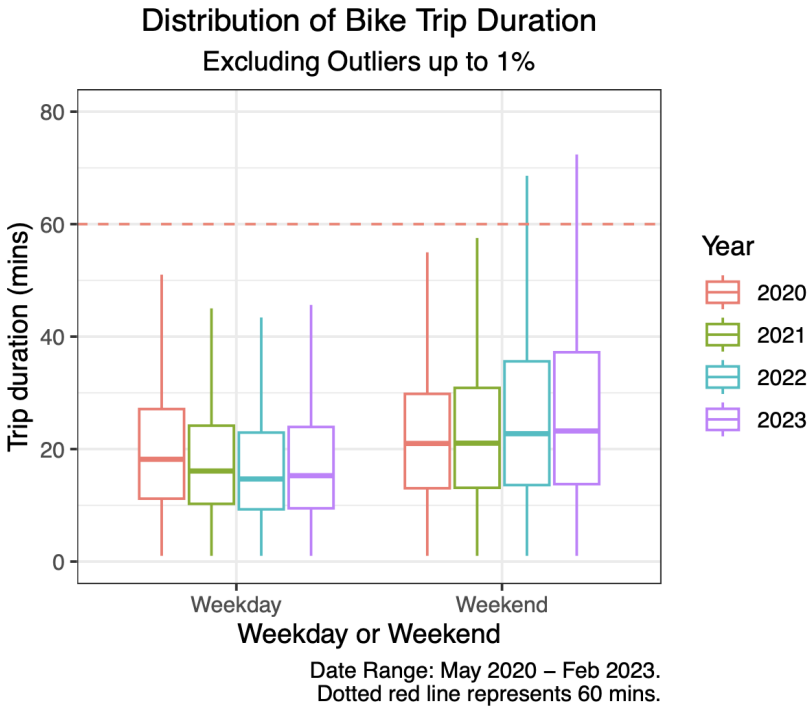


Figure 2.3: Boxplots that shows distribution of trip time during the week and during the weekend. Outliers excluded.

Figure 2.2 that includes the entire dataset shows that there are observations that have a huge gap to the average time per trip, and that take more than 3 hours. Since these don't really make sense given the current bike-sharing system, which limits trips to a duration of 1.5 hours with renewal, we have removed these observations, using a 1% winsorization technique. The result after winsorizing is shown in Figure 2.3. It's interesting to see that the trend between years and weekdays or weekends is reversed. Seems like users had longer durations in previous years during weekdays, but longer bike rides during weekends in the most recent years.

This left us with ~7.6M observations, discarding less than 100k trips that were considered to be outliers. This accounts for 316 stations for which we had both capacity information coming from the Public Transport API and the trips dataset. The following exploratory analysis may hold data from before the quarantine, but the model and the results were built solely on observations that have survived this data preprocessing.

2.3 Descriptive Analysis

After having a healthy dataset which we can trust, we sought to dig deeper to understand the historical location and movement of stations and the demand for them throughout time.

The first point worth mentioning is how close current bike stations are to actual bike lanes. We want to make sure that the station's latest location does not make users have to expose themselves to cycling on streets before finding an actual bike lane.

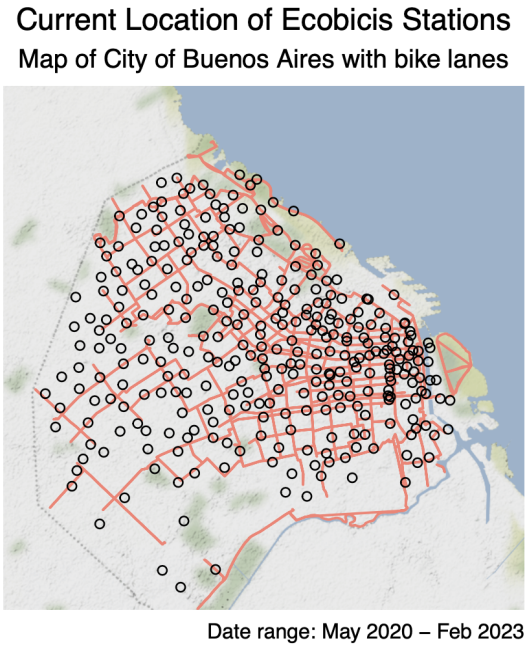


Figure 2.4: Bike-sharing Stations current location and proximity to public bike lanes, represented by the red lines.

As seen in Figure 2.4 of the City of Buenos Aires, stations that are close to the downtown area, in the mid-right hand side of the map, usually have many bike stations that pass near them. There is also a high density of bike-sharing stations around the center area, and a general lower

concentration of stations and bike lanes around the peripheral zones, especially in the southern areas.

The second point worth mentioning is that not all areas of the city have had the same behavior when comparing demand per station before and after the COVID19 quarantine. In order to make a fair historical comparison, we split the city into 1 km2 quadrants, making sure we were not subject to stations moving or going out of commission. Some of the quadrants do not have any stations within them and some have more than one. The segmentation and the amount of stations in each quadrant are shown in Figure 2.5. Leveraging the km2 quadrants, we computed the ratio between the share of trips that originated from each quadrant before and after quarantine and showed that in the heatmap plot, in Figure 2.6.

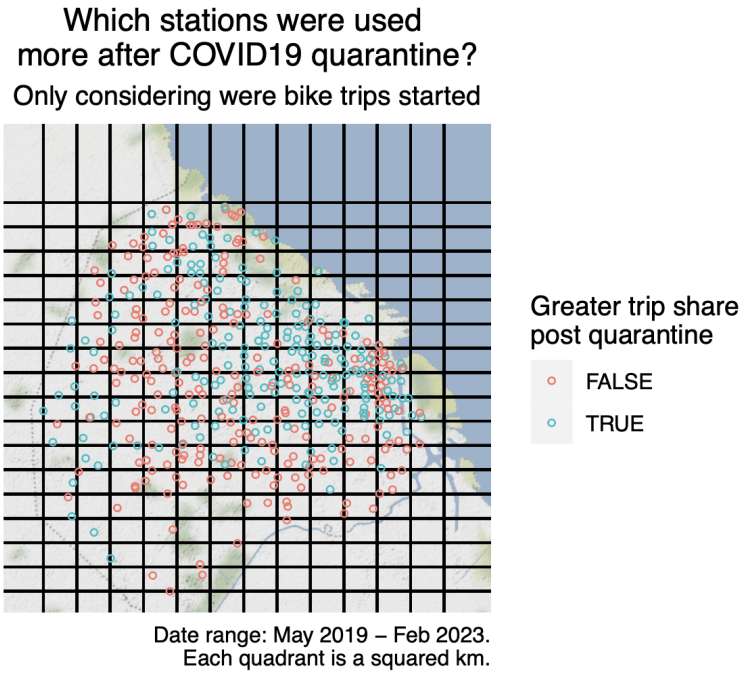


Figure 2.5: Grid of squared km throughout the city, showing the stations that are included within each of them. Color of dots represent stations that had a lower share of trips post quarantine in red (higher share of trips post quarantine = False) and those that had a higher share of trips post quarantine (higher share of trips post quarantine = True).

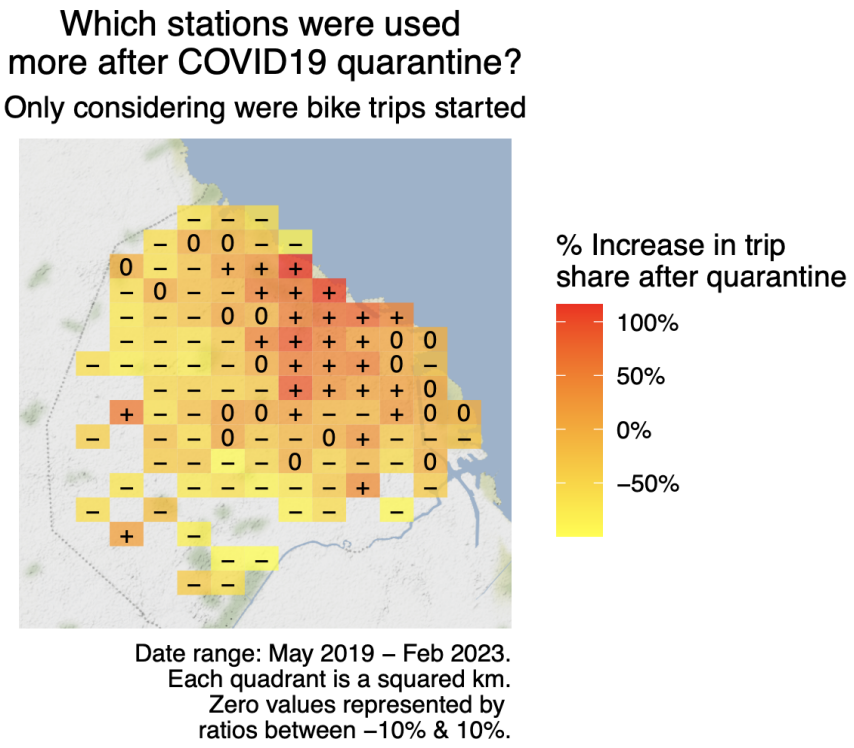


Figure 2.6: Ratio of shares of demand before and after COVID19 quarantine for each squared km.

Quadrants with intense red colors in Figure 2.6 have had a higher relative demand after the quarantine period, and are represented with positive signs (“+”). On the other hand, quadrants with lighter shades of red and yellow have had a lower relative demand after COVID19 hit and are represented by (“-”). Quadrants with zero values mean that the increase was between -10% and 10%. This is good evidence that shows that there is a need for a relocation of stations throughout the city. Places with the largest quantity of stations downtown show lighter colors than places around the city center, specially close to the river and North East area.

The heatmap could be a reflection of new bike-sharing stations being positioned in certain areas, which could make trips go up. In essence, this has to do with causality: does the offer of bikes make people cycle more in that area, or does the saturation of the system in certain areas due to high usage make the owner react and build new bike-sharing stations? As shown in Figure 2.4, some of the stations in common in both periods have been less used. If people were to go where the bikes are, then no stations would be underused throughout time. Also, we believe that a rational owner's decision process to open new stations would have to take into account current area usage of bikes. Hence, for this analysis we will assume that more stations being opened in an area happens due to high existing station usage in that zone.

Figures 2.5 & 2.6 show us where the trips started. To complement this insight, we wanted to understand where users were going to as well, using the dataset from May 2020 onwards. For that reason, we built a directed graph, where each node represents a station. The size of each node represents the quantity of trips that originated at that station. The objective of this visualization is to observe if there is some clear pattern at different points in time. We compare graphs based on aggregate trips for Monday and Sunday at different times of day, going from 7 am to 12 pm and from 1 pm to 8 pm.

The reason behind why we chose single days (Monday's and Sunday's) instead of analyzing the entire work week and weekend has to do with user behavior. When we analyzed these two sets of graphs, we noticed that they were similar. This makes sense, given the volume of trips that was incorporated in this analysis, as mentioned in Section 2.1 and 2.2. Furthermore, we thought that looking at single days and hour ranges provided more granularity to the analysis than looking at entire work weeks or weekends.

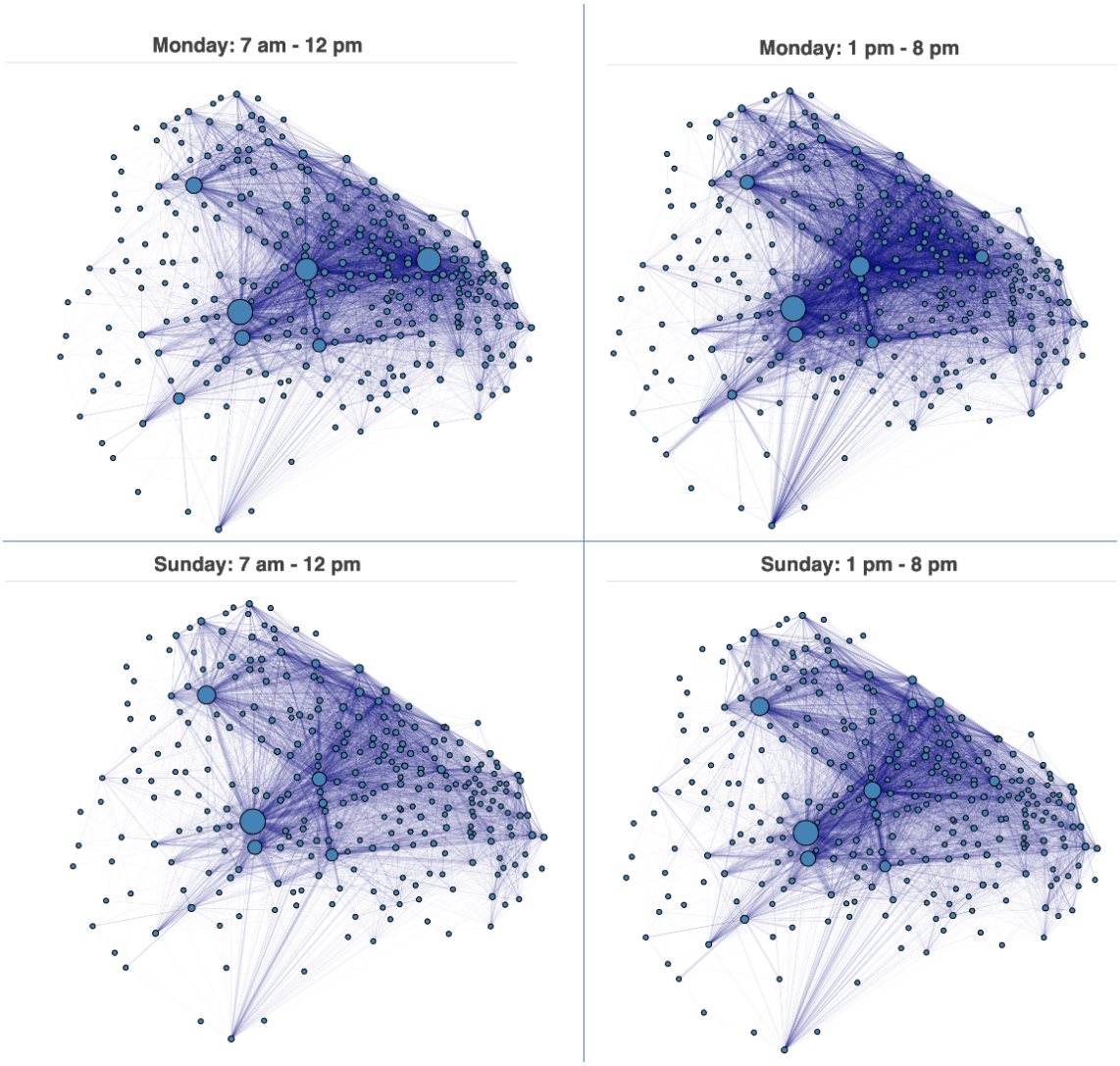


Figure 2.7: Graph using station locations as nodes, size of nodes represent amount of trips that originated from

them. Comparing Mondays and Sundays with different hour ranges throughout the day.

From looking at Figure 2.7, a couple of insights become quite clear. First off, there is more activity around downtown on Mondays than on Sundays. This is expected since there are probably some people commuting to work on weekdays, so it makes sense for there to be larger nodes around that area. Secondly, the rest of the big nodes are pretty common between both days, although there seems to be a larger share of trips in the northern area of the peripheral region on weekends than on weekdays. Furthermore, there don't seem to be considerable differences between the morning and the afternoon of the same day. Finally, the center of the city has quite a lot of activity, both on weekdays and on weekends. These differences in behavior between different days of the week led us to differentiate our demand between weekends and weekdays, but not between hours of the day. This will allow our model to provide a different station network configuration for different moments of the week and could be more efficient. We will provide a deeper analysis of this matter in the next section.

3. Model

As mentioned in our Literature Review section (Section 1.2.1), we will leverage the existing work on CFLP. The standard CFLP minimizes the total cost of maintenance of the system, considered to be the sum of fixed costs of maintaining a facility and transport costs of shipping a product from a facility to a customer. It also includes restrictions that make sure that all of the demand from customers is satisfied, whatever the distance to a facility may be. Therefore, we have two sets of nodes, facilities and customers, and this problem looks to define where stations should be located.

We apply the CFLP to our bike-sharing network. Our bike-sharing stations are the CFLP's facilities. We define centers of demand locations as the centroids of the area of points that are closest to each station, these will be our CFLP consumers. Their demand is predicted based on the usage of bikes in each station. We guarantee that all demand is met by the available bike-sharing stations, that demand cannot exceed the station's capacity and that people from a center of demand cannot walk more than a certain distance to the station where they are allocated to. Our model will be a particular case within the family of CFLP.

3.1 Variables, Objective Function, Restrictions

Paraphrasing Liu et al. (2015) [6], the bike sharing network optimization problem for bike sharing systems can be split into two: demand estimation for each bike station and station network optimization. The former is a prediction problem, where we try to assess what the demand for each station will be like in the future and incorporate that into the model. The latter is a linear programming optimization problem, which we will address leveraging the preconceived literature around CFLP. We define the model and its variables first, and then go into how to estimate each of the inputs that it needs, including demand estimation.

The goal of our model is to minimize the aggregate cost of deploying and building the bike-sharing network. There are two main sources of expenditure that it takes into account: variable costs and fixed costs. Variable costs take into account the distance that users have to travel from the centers of demand to the actual stations. Variable costs are only taken into account when a user from a demand center decides to travel to a station. Fixed costs represent the money that the owner of the bike sharing system needs to spend in order to build a new station. These are only taken into account when a station is opened. In essence, this CFLP objective function incorporates costs that the owner faces but also includes a user perspective, ensuring that they will not have to travel large distances to reach an active station and be able to cycle. Hence, our model objective function becomes:

$$min\, Total\, Cost \;=\; min \sum_{i=1}^{n} \sum_{j=1}^{m} (C_{i,j} * x_{i,j}) + \sum_{j=1}^{m} (F_j * y_j)$$

Going into what each of the variables stand for:

- Centers of demand are represented with the letter "*i*", and go from 1 to the amount of centers of demand ("n"). $G = centers \in \{1;\ldots;\ n\}$.
- Bike-sharing stations are represented with the letter "*j*", and go from 1 to the amount of stations in the system ("m"). $B = bike\ stations \in \{1;\ldots;\ m\}$.
- Variable costs that include distance from a center of demand *"i"* to a station *"j"* are represented by $C_{i,j}$. This is a "*nxm"* matrix that holds the distance for each center of demand to each station, where centers of demand are rows and columns are stations.
- F_j includes the fixed costs of opening each station *"j".* It is a vector of dimension *"m".*
- y_j is a binary variable that indicates whether the station *"j"* is opened or closed. There are as many y_j as there are stations in the model. $y_j \in \{0;\ 1\}$.
- $x_{i,j}$ is a continuous variable that indicates the share of demand from demand center *"i"* that is allocated to station *"j"*, and goes between 0 and 1. $0 <= x_{i,j} <= 1$.

As any model, there are different restrictions that it faces to find an optimal solution. The first restriction worth mentioning is a capacity constraint per station, meaning that the total demand per station is less or equal than the total supply of bikes it has. For simplification purposes, we are assuming that supply is the same as the capacity of each station, which means that there are as many bikes in the system as docks available in each station. The new variables showed here symbolize the predicted demand per center of demand "i" (d_i) and the capacity of each station "j" (k_j). This restriction is repeated for all "j" stations in the system.

$$\text{(1) Capacity constraint: } \sum_{i=1}^{n} d_i * x_{i,j} <= y_j * k_j \quad \forall\, j \in B$$

The next restriction that goes into the model has to do with the continuous share of demand that can be allocated to different stations. Naturally, the shares of the demand cannot add more than 1, because it would mean that we have a demand prediction that is below actual demand.

$$\text{(2) Demand constraint: } \sum_{j=1}^{m} x_{i,j} = 1 \quad \forall\, i \in G$$

The final restriction that will go into the model is about the maximum radius that users can walk from their center of demand to the station they were allocated to. This restriction can affect the model feasibility depending on the walkable distance threshold that we do not want to exceed. A very small threshold could turn the solution of the model non-feasible, given that stations that are so close to a center of demand may not exist. In order to bring this restriction to life, we will update the upper bound of the assignment of center of demand "i" to station "j" to be zero. This means that there is no possibility of the demand center "i" to be associated with station "j" because the distance between them exceeds the maximum distance threshold "T". The only new parameter incorporated here is "T", which symbolizes the walkable distance threshold.

(3) Walkable distance constraint: $x_{i,j} = 0 \ \ if \ C_{i,j} > T$

Putting everything together, the model that will be optimized becomes:

$$min \sum_{i=1}^{n} \sum_{j=1}^{m} C_{i,j} * x_{i,j} + \sum_{j=1}^{m} F_j * y_j$$

$subject\ to$:

$$\sum_{i=1}^{n} d_i * x_{i,j} <= y_j * k_j \quad \forall j \in B \ \ (1)$$

$$\sum_{j=1}^{m} x_{i,j} = 1 \quad \forall i \in G \ \ (2)$$

$$x_{i,j} = 0 \ \ if \ C_{i,j} > T \ (3)$$

$$0 <= x_{i,j} <= 1 \ ; \ y_j \in \{0; 1\} \ \ (4)$$

sets:

$$G \in \{1, 2, \dots n\}$$
$$B \in \{1, 2, \dots, m\}$$

The optimal solution minimizes the total cost, evaluating simultaneously the cost of opening a station (y_j) and the cost of a particular station to serve a particular demand center ($x_{i,j}$). Capacities (k_j) and station locations are given by the current status of the network and are extracted from the data sources described in Section 2.1. This is another assumption of the model: facility location does not change from its initial position. Existing facilities can only be turned on or off. In other words, we assume that the current station location represents the universe of options to locate a station with a certain capacity, and we want to choose which ones to use.

Demand per station will be predicted, variable costs will be constructed based on a notion of current demand centers and fixed costs are assumed to take different values in order to analyze different scenarios. We describe how we are calculating each of these values in section 3.2.

3.2 Calculating Demand

Demand per station is something that depends on user disposition, station capacity and bicycle availability. Each station has a different pattern that we need to quantify in order to provide the most accurate inputs to this model. The reason why demand is being predicted and we are not using former data points is to give some uncertainty to the model. We do not know for sure what

the demand will be in the future when our model recommendation is implemented, so we would like to be as cautious as possible to guarantee that our network solution is robust.

Therefore, we predict the demand up to 180 days going forward, and keep the time period with the maximum prediction per station. This ensures that we are considering a worst-case situation, where demand for each station is as high as possible. We observe data at a daily granularity level to make the prediction. Furthermore, we save the average prediction for that time period as well as the upper bound of the confidence interval of the prediction, to give the model some variability, and will run them through the optimization algorithm.

3.2.1 Uncensoring demand

We considered whether demand per station was censored due to bike availability. The good thing about bike-sharing systems is that there are two sides to the demand. We have to take into account both the bike extractions as well as the bike deposits. Both are important given that if a person does not have a bike to take out, then that demand is censored. However, if a user deposits a bike in the station, then that enlarges daily capacity, since it's a bike that can be used by another user and a new trip can be completed for that day. While we do not know the total number of bicycles available in a station before each trip happens, we can calculate what the accumulated stock of bikes in a station looks like by sorting the extractions and deposits by time.

We start by assuming that the bicycle stock of each station starts at zero. This is a necessary assumption given that we have not included trip data going back further than May 2020, for reasons we have described in Section 2.2. In other words, all stations start out with a balanced bicycle stock that can only be changed by people's extractions and deposits. Then, we add one every time a deposit is made and we subtract one every time a bike extraction is made.

Leveraging this framework of aggregate bicycle stock throughout time, we can think of three types of stations:

- **Stations that keep themselves stable**. These are stations that have a similar number of extractions and deposits throughout time and regulate themself without any intervention.
- **Stations that are prone to extractions**. The natural foot traffic of the station means that users are always prone to take a bike from that station and deposit it somewhere else.
- **Stations that are prone to deposits**. These stations are places where people usually tend to deposit bikes throughout time.

Out of the 316 stations we considered throughout this thesis, we chose one station as an example for each type of station. We plotted the accumulated stock of these three in Figure 3.1, using the trips dataset, from May 2020 to Feb 2023. The red lines represent the station capacity.

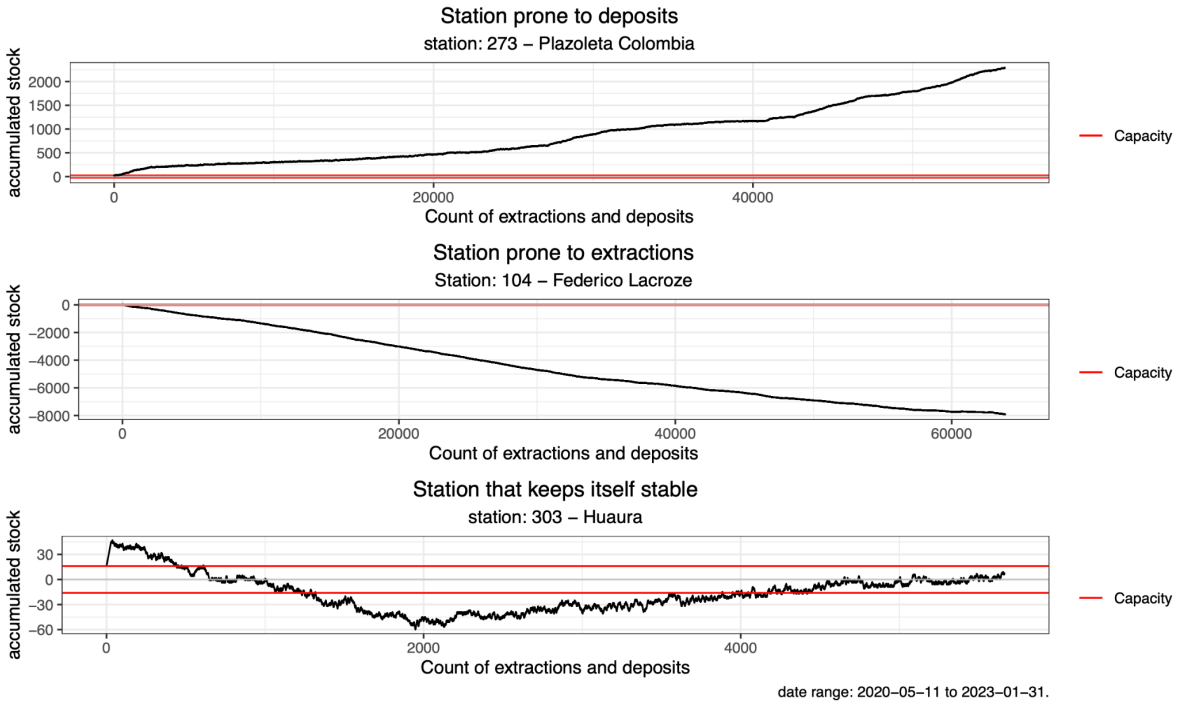


Figure 3.1: Accumulated stock throughout time for different types of stations.

As we can observe from Figure 3.1, stations that keep themselves stable like “303 - Huaura” have an accumulated level of stock that is approximately similar to the station capacity. These stations probably do not have a lot of activity relative to the rest and are quite dependent on people’s movements, choosing to extract or deposit a bike. Note the scale of the y axis that is always showing that the accumulated stock is between -50 and 50, around zero.

However, stations that are prone to extractions like “104 - Federico Lacroze” demonstrate interesting behaviors. Users constantly decide to go on bike rides from bikes in that station and deposit them somewhere else. Essentially, this means that there would not be any more bicycles available until users started depositing them again. This is conclusive evidence of government intervention and bicycle redistribution via central planning.

The opposite problem occurs with stations that are prone to deposits, as shown in the “273 - Plazoleta Colombia” station example in Figure 3.1. Given that the accumulated stock for these does not stop growing, there has to be a physical person taking bicycles that users deposit away from stations prone to deposits into stations prone to extractions that need them.

As we can see from Figure 3.2, this is not an isolated behavior. There are actually many stations that fall into the understocked (prone to extractions) or overstocked (prone to deposits) category. Another visualization can be observed in Figure 3.3, showing that there are both overstocked and understocked stations.

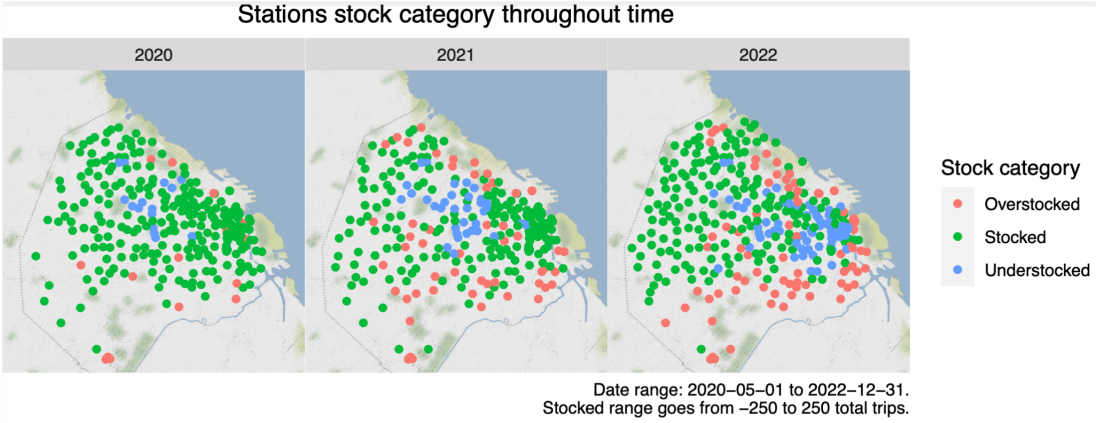


Figure 3.2: Location of stations according to their accumulated stock, from May 2020 to Dec 2022.

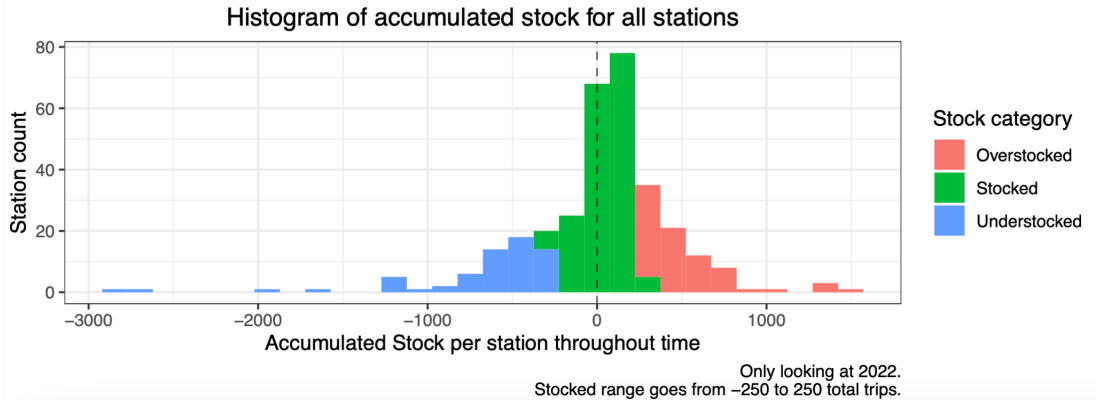


Figure 3.3: Distribution of stations according to their stock from Jan 2022 to Dec 2022.

Based on this insight, our conclusion is that government intervention is prevalent in this system. We will assume that this task is done by minimizing the chances of users wanting to go on a bike ride and not having bikes available in the station they like. Therefore, throughout this work, we assume that demand is uncensored due to government bike redistribution between stations.

Furthermore, since both of these types of stations are necessary for the system to function, we will consider station demand to be the maximum between bike extractions and bike deposits per day. That way we guarantee that a station will be considered as popular when people use it, regardless of whether it is prone to deposits or to extractions. We will use this definition to build the input data we need for the model to predict demand per station. We will call this the demand dataset.

$$d_i = max(bike\ deposits_i;\ bike\ extractions_i) \quad (5)$$

3.2.2 Predicting demand with Prophet

Once we defined what the demand would be, we built the dataset input as a time series, per day and per station. We leveraged the Prophet package to predict what the demand would be for the next 180 days since it is accurate, robust and easy to use.

3.2.2.1 Prophet Methodology

Paraphrasing Taylor and Letham (2017) [11] Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, and weekly seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

The authors propose a practical approach to forecasting that combines configurable models, with a modular regression with interpretable parameters that can be intuitively adjusted by the analyst with specific domain knowledge on the time series. It is essentially an ensemble model, with three main components: trend (g_t), different types of seasonality (s_t) and holidays (h_t), as well as an error term (ϵ_t) that represents any idiosyncratic changes that are not accommodated by the model.

$$y_t = g_t + s_t + h_t + \epsilon_t \quad (1)$$

The trend model uses a nonlinear saturation growth model as a base, which is tied to population growth in natural ecosystems. In our case, since we are modeling trips per station throughout time, the saturation will converge to the total number of bicycles in that station at a certain point in time.

The seasonality model tries to capture effects that are repeated with a certain frequency throughout time. The authors rely on Fourier series to provide a flexible model of periodic effects. For our bike-sharing problem, we could have different seasonalities in place. Weekly seasonality could be in place for a station in areas that are more active throughout the work week, with more activity from Monday to Friday. Yearly seasonality could also be at play, given that users might decide to go on fewer bike rides due to hard weather conditions in the winter, and enjoy cycling more during the summer.

Holidays are included into the model by assuming that the effects of holidays on the time series are independent. They add an indicator function representing the time at which a holiday happened and assign a parameter to each of them, which corresponds to the change in the forecast. Argentina is in the top 20 countries with the most amount of holidays throughout the world. Depending on the station location, holidays may affect the station's activity positively or negatively in terms of trips, as we will see in section 3.2.2.2: "Prophet Application". For instance, a station located in typical recreational areas along the periphery of the city of Buenos Aires

could have a larger amount of trips occurring on holidays, given that people are off work and may like to cycle more in that area.

3.2.2.2 Prophet Application

We started by analyzing a single station and assessing the model's performance.

We chose the "002 - Retiro II" station because it is in a downtown area, typically very active and capturing lots of traffic. In line with the author's mantra about robust estimates with easy implementations for the analyst, we obtained a forecast by activating the trend, yearly and weekly seasonality mentioned above. We also incorporated the Argentinian holidays dataset into the model, and used the demand dataset of trips per day for this particular station as the dependent variable. Immediately, the model finds a good fit based on the pre-existing data with a low Root Mean Squared Error (RMSE) and predicted 180 days forward. Figure 3.3 shows the actual values as gray points and the model's fit in blue with its confidence intervals. The dotted line shows the station capacity.

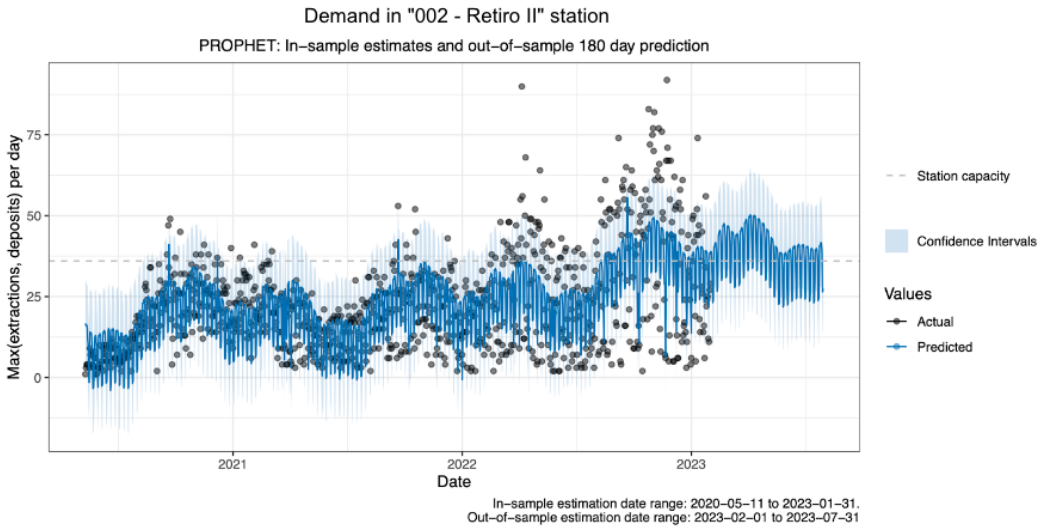


Figure 3.4: Prophet model fit and 180 day prediction for station "002 - Retiro II".

As we can observe, Prophet seems to pick up the trend of data quite well, since it started out low and then grew, even though there is a larger dispersion in the actual values. For the future, it predicts values that are similar to the last ones. Around the time of June-July, there is a downward trend of the demand, which accounts for winter and how bike-sharing demand falls due to the cold weather. It's also interesting to see how Prophet is able to capture certain spikes in the data, that are probably due to special dates, where people changed their cycling habits since they had the day off. We will analyze this further below by decomposing the demand into the model's parameters.

This information can also be seen when we decompose the actual values into trend, weekly and yearly seasonality and holidays components, as we look at Figure 3.5. In order to measure

Prophet's goodness of fit, we also analyzed what the RMSE was for different prediction lengths, ranging from 20 to 180 days, in Figure 3.6.

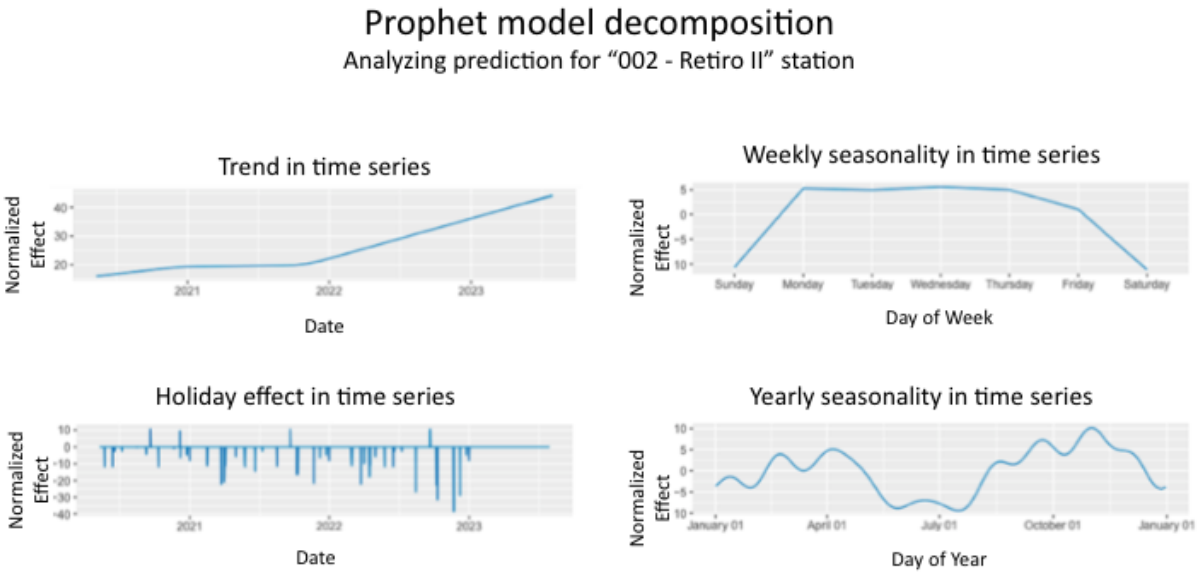


Figure 3.5: Prophet output analysis. Showing decomposition of time series between trend, holidays, weekly and yearly seasonality. Analyzed for "002 - Retiro II" station.

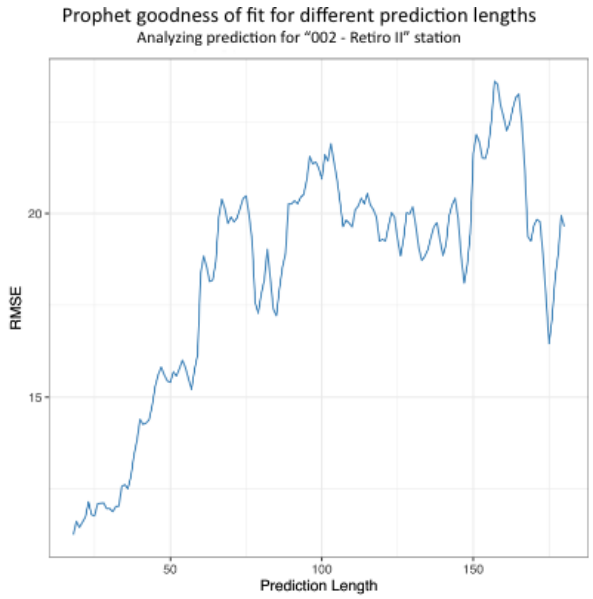


Figure 3.6: Prophet goodness of fit analysis. Showing the RMSE over different prediction horizons. Analyzed for "002 - Retiro II" station.

As we can observe in Figure 3.5, the trend goes up by the beginning of 2022. The model predicts it will continue to grow throughout 2023. Furthermore, looking at holidays and seasonality, some of these components can be extrapolated and others cannot. For example,

the yearly seasonality that shows a lower amount of trips during winter and a higher amount of trips during autumn and spring will probably be seen across all stations. However, given that this is a station located in the downtown area, we observe that demand goes down during summer (from December to February in the yearly seasonality). The same can be said for the days of the week, with a lower demand for Saturday and Sunday and a higher demand on weekdays. Most holidays also account for a smaller demand. This all has to do with this particular area of the city, near offices and businesses. Most users of this station usually do so on weekdays because they go to work. People do not go to work on holidays, on weekends. They usually take time off during the summer. Therefore, this behavior should be considered as particular to this station. Stations in the peripheral areas or center of the city will probably have a different yearly or weekly seasonality.

RMSE behaves as expected in Figure 3.6. It is smaller when the prediction is closer to the latest data points, and higher when the prediction is a few more periods into the future. In order to benchmark our model, we compared Prophet's performance in terms of RMSE with a Single Exponential Smoothing (SES) model, using the same simple dataset. Model fits and predictions for both models are compared below, in Figure 3.7. We use the last 180 days of actual values as a testing set and train the model on all previous days.

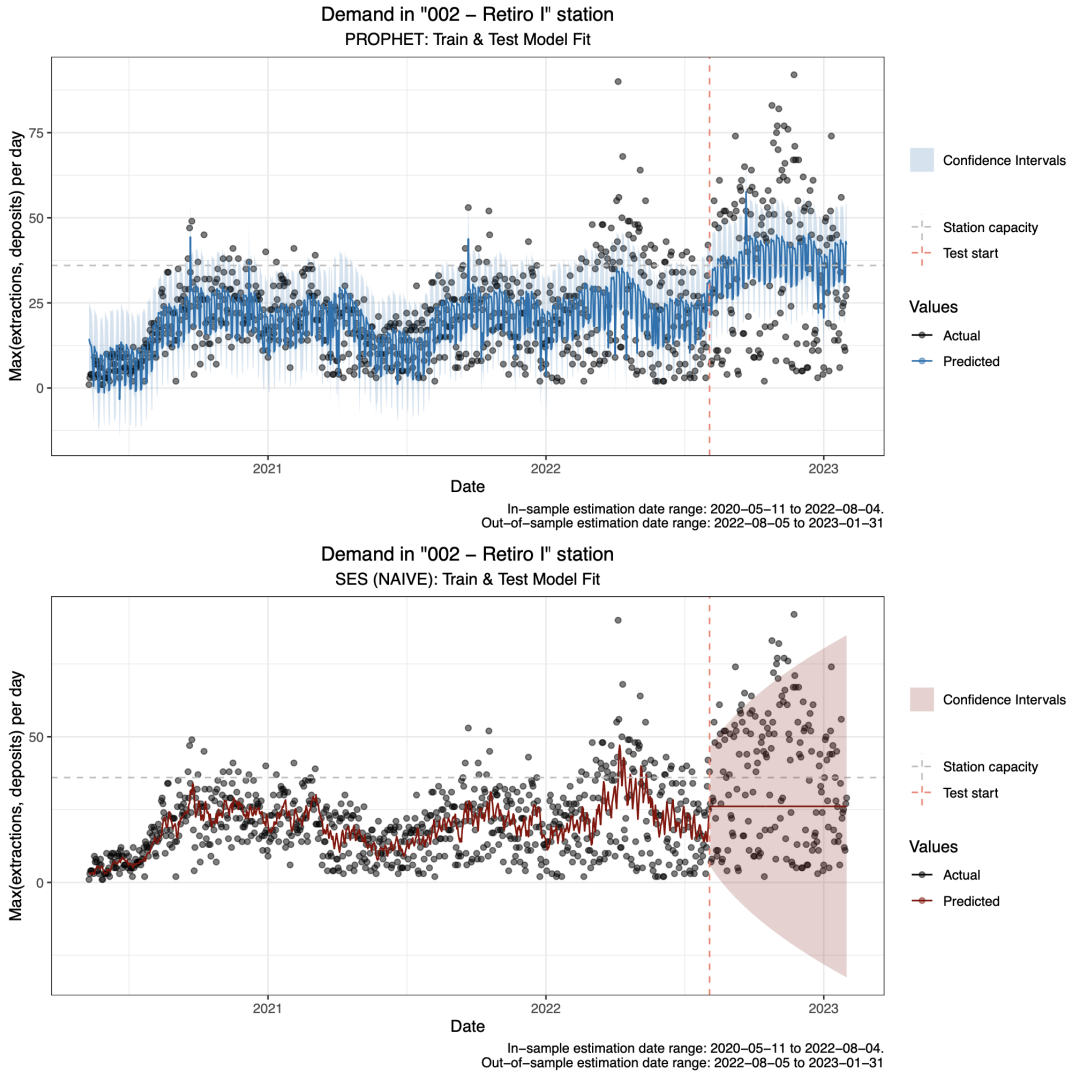


Figure 3.7: Prophet fit and prediction (top) vs SES Naive model fit and prediction (bottom). *Analyzed for "002 - Retiro II" station.*

Comparing Prophet and SES, we can observe that Prophet has a much more precise fit in the testing and training period. This is evidence that it does not overfit and that it is better at predicting the future than the naive model, which only predicts that the future is the mean of previous observations with a confidence interval that enlarges as more days go by.

	Train	Test	All periods
Prophet RMSE	8.48	18.1	10.4
SES RMSE	10.5	23.8	13.3
% Improvement Prophet	19%	24%	21.8%

Table 3.1: Comparing Prophet & SES model in terms of RMSE for training and testing dataset, only in "002 - Retiro II" bike-sharing station. Testing dataset defined as values for the last 180 days. Training dataset defined as periods that are not in testing.

We also added the RMSE in each subset for all models in the "002 - Retiro II" station to quantify how much better Prophet is. As observed in the final row, Prophet achieves an improvement in performance of above 20% in terms of RMSE than our naive model.

We repeat this demand estimation for each station separately. As mentioned previously, we capture the date with the maximum prediction for our demand function for each station, split by weekend and weekday. We also keep the predicted mean of that date and the upper bound of the predicted confidence interval. Therefore, we have a combination of 4 different demand predictions per station to incorporate multiple scenarios:

1. Mean of max predicted demand on weekends.
2. Mean of max predicted demand on weekdays.
3. Upper bound of confidence interval of max predicted demand on weekends.
4. Upper bound of confidence interval of max predicted demand on weekdays.

3.2.3 Hours of activity

The last piece of the puzzle missing is to calculate the hours of activity during each day that each station has. With that we are able to define the demand per hour of activity of each station, dividing the worst-case demand per day scenario mentioned in the previous point and the hours of activity. On top of that, computing the hours of activity per station gives an additional level of variability to the model. Instead of saying “all stations are active from 9 am to 7 pm” which is quite arbitrary, we will only count an hour as active for each particular station if the hour has a higher share of trips than the average amount of trips per hour that the station had. This leaves us with a balanced distribution, with a mean of around 11 hours of activity per day (portrayed in green in the plot below).

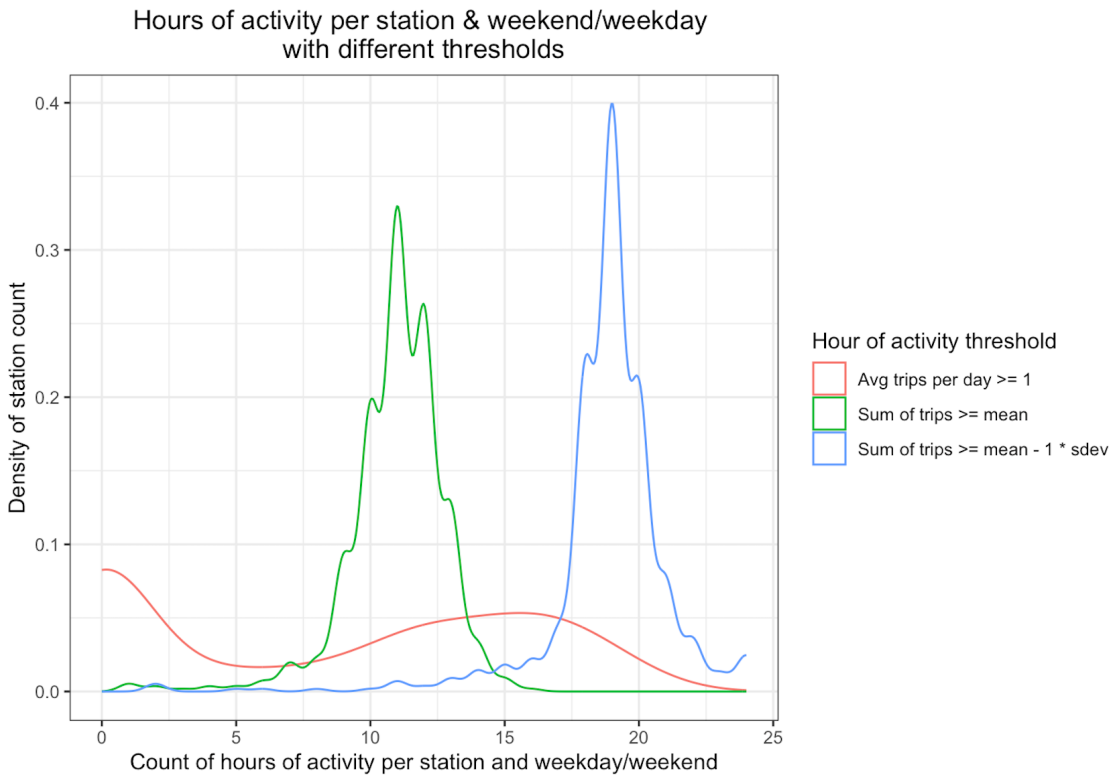


Figure 3.8: Looking at different thresholds to consider an hour where a station had extractions or deposits as active.

We also explore the distribution of hours of activity per station considering other thresholds and compare it with our preferred option. The count of hours of activity distribution when hours of activity have more than 1 trip per day is drawn in red. Although this rationale makes sense, this threshold seems to be challenging, given that there are many stations with few hours of activity, which does not seem to be the case in practice.

We further show the distribution of hours of activity when we reduce the threshold to the difference between the average quantity of trips per station and their standard deviation. This makes most of the stations have almost all of the available hours within a day as active, which we know is also not the case. Therefore, we have chosen the threshold of hours of activity to be hours with a higher number of trips than the average for that station, and we will divide demand that was estimated per day by that quantity, per station and per weekend or weekday.

Having the demand per active hour, as proposed by Liu et al. (2015) [6], ensures that we are computing demand for each center by the same unit of measurement. Stations with more activity per day have a higher demand per active hour, but their standard deviation to stations with a lower demand per day decreases, given that highly active stations also are active throughout more hours in the day, as reflected in Figure 3.6. Furthermore, this way of computing demand can also be tied back to the CFLP model mentioned at the end of Section 3.1. Having a demand per active hour guarantees that we can further optimize the system at a certain point in

time, especially considering that bike rides in our bike-sharing systems cannot last more than 90 minutes..

3.3 Calculating Variable Costs

Now that we know the demand each station faces, we need to determine where it came from. Demand location is a key factor that goes into the objective function that we would like to minimize, since we need to compute the distance between station locations and centers of demand. By doing this, we will quantify how much it costs for users to travel to a certain station.

3.3.1 Locating centers of demand

Replicating what Liu et al. (2015) [6] did in their paper, we will build demand centers as the centroids of the Voronoi regions that surround each station. A Voronoi diagram is a partition of a plane into regions, where each region is composed of all the points that are closest to the object we care about. In our case, we are using station locations as the objects we would like to divide our plane on. The rationale behind this is that every user looking for a bike ride will go to the station that is closest to them. Therefore, we are assuming that all users that are within a Voronoi region will go to the station that is in the area. Furthermore, each station will have a Voronoi region surrounding it, and a center of demand at the center of said Voronoi area. We observe what the Voronoi diagram looks like using our station location in Figure 3.9.

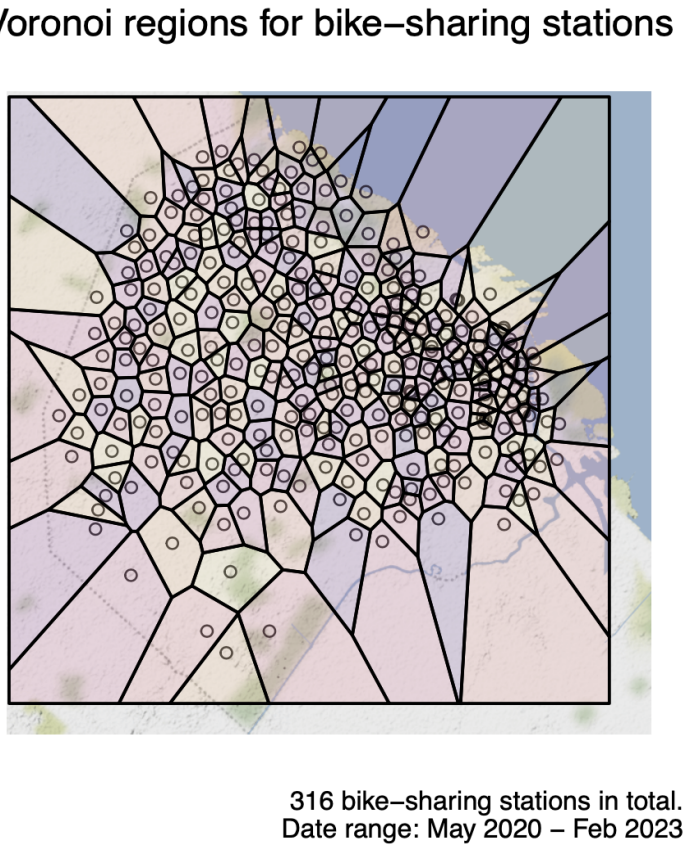


Figure 3.9: Voronoi Diagram based on station locations.

Being geometric figures, each Voronoi region around a bike-sharing station is surrounded by its vertices. We average all the vertices of a figure to decide where the physical demand will be located, and repeat this for all Voronoi zones. Hence, the center of each area around a station will become the position of the center of demand. The distribution of distances between stations and their centers of demand can be seen in Figure 3.10

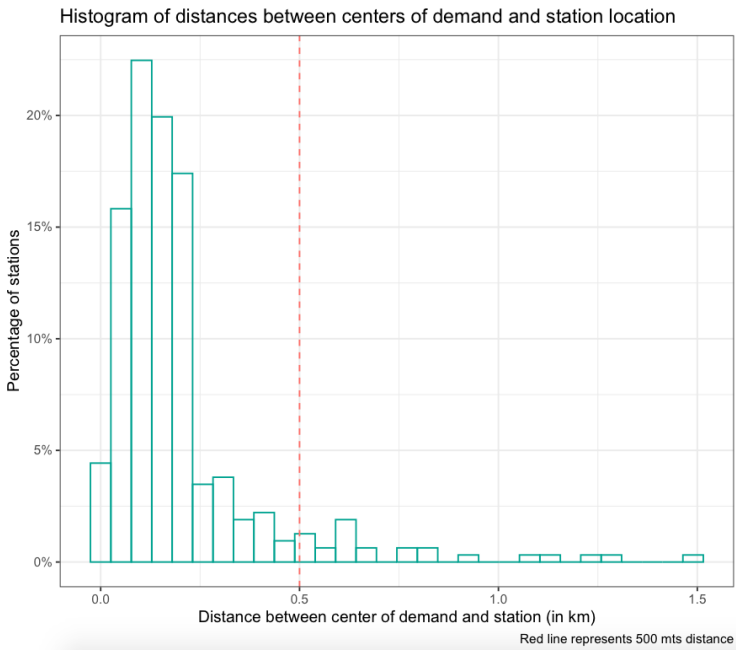


Figure 3.10: Distribution of distances between stations and their centers of demand.

The farther the stations are from each other, the larger the Voronoi area is. In the downtown area, where there are many bike-sharing stations, we observe tiny regions. When we move closer to the peripheral regions that have a lower density of stations, Voronoi regions are quite large, since more users have that only station at their disposal.

Averaging the vertices that surround a station is a technique that will compute the centroid of the figure around each station. However, the stations around the periphery of the city are only limited by an artificial restriction that is imposed by the Voronoi package. Ideally, we would like to define this limit as the city perimeter, to avoid having centers of demand that are very far away from their corresponding stations.

Out of the total 316 stations, only 12% of the stations had a distance to their center of demand larger than 500 mts, as observed in Figure 3.10. All of these stations are along the peripheral area of the city and have to do with the fact that the perimeter is ill-defined. Then, we built an artificial city perimeter, by connecting the cities that surround the city. The original centers of demand are shown in Figure 3.11. To solve this problem, we excluded all Voronoi vertices that fall outside of that perimeter. Figure 3.12 shows how demand centers' positions change when

Voronoi vertices that fall outside of the city are excluded. After the exclusion, all demand centers fall within the city perimeter.

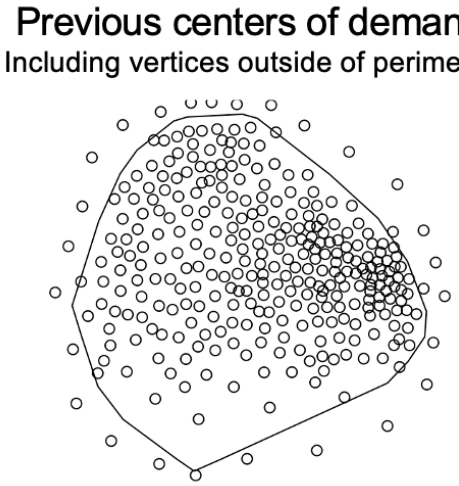


Figure 3.11: Centers of demand prior to the vertex exclusion. The perimeter based on peripheral stations is represented as the black line around the dots.

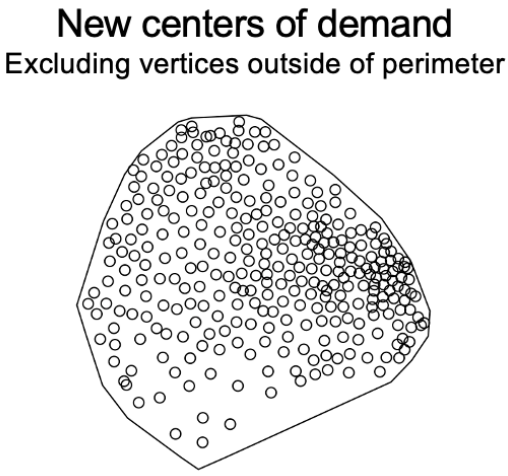


Figure 3.12: Centers of demand after the vertex exclusion. The perimeter based on peripheral stations is represented as the black line around the dots.

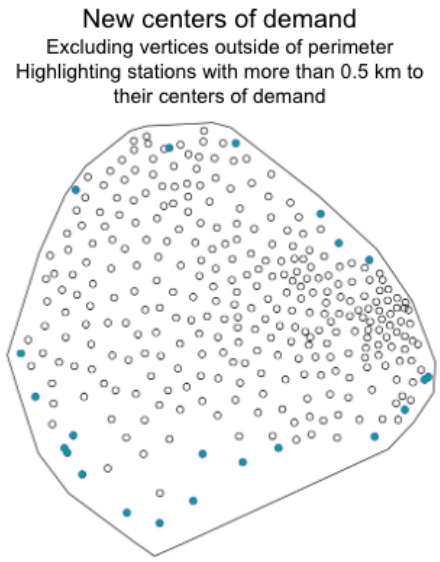


Figure 3.13: Centers of demand after the vertex exclusion, highlighting stations that still have more than 0.5 km to their center of demand, in green. The perimeter based on peripheral stations is represented as the black line around the dots.

This leaves around 7% of stations with a distance larger than 500 mts to their center of demand, highlighted in Figure 3.13 as green dots. Since centers of demand are a fabrication in order to be able to position the demand somewhere, we changed the distance to these centroids to be 100 mts away from the station they came from. This allows us to have feasible solutions when we reduce the "*T*" parameter from the walkable distance restriction, reported in Section 3.1.

3.3.2 Distances between stations & centers of demand

The final input that we are missing to run the model is the variable costs, represented by the distance between centers of demand and station locations. There are many ways to quantify the distance between two points. One could select the Manhattan distance, which is the sum of absolute differences between latitudes and longitudes of the two points, or the Euclidean distance, which is the root sum of squares of differences between latitudes and longitudes of the two points. A third option is to query the Distance Matrix API (i.e. Google Maps, from now on DMA) and ask it what is the distance for the route that should be taken between two points if a person was to walk between point A and point B. These three types of distances are exemplified in Figure 3.14.

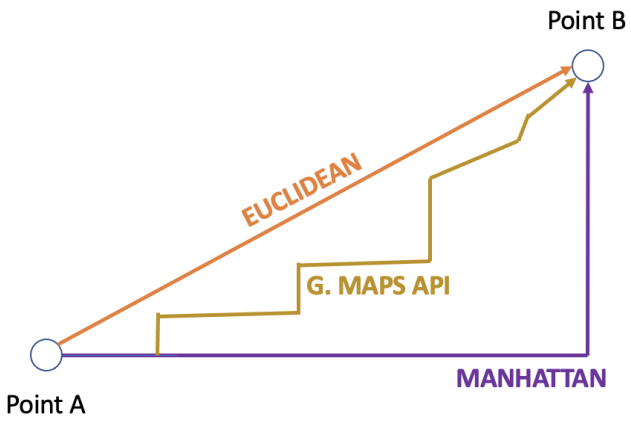


Figure 3.14: Illustrative example of types of distances that exist between two points.

Since people will be walking to their station of choice, our preferred distance would be the one coming from the DMA. However, because we have 316 stations and 316 centers of demand, and we have to calculate the distance between all of these points (close to 100k pairs of distances), querying the DMA would become expensive.

To validate whether the Euclidean or the Manhattan distance was a better choice compared to the DMA, we selected a subset of 10 different station locations, built the 45 pairs of locations and passed them through DMA. Then we calculated the Euclidean and Manhattan distance between each of the pairs of points. Finally, we analyzed the percentage gap that each distance had to the actual DMA value. We show this value for each pair of points in Figure 3.15, a blue line for Manhattan and a red line for Euclidean. The median gap for both distances are charted as dotted lines.

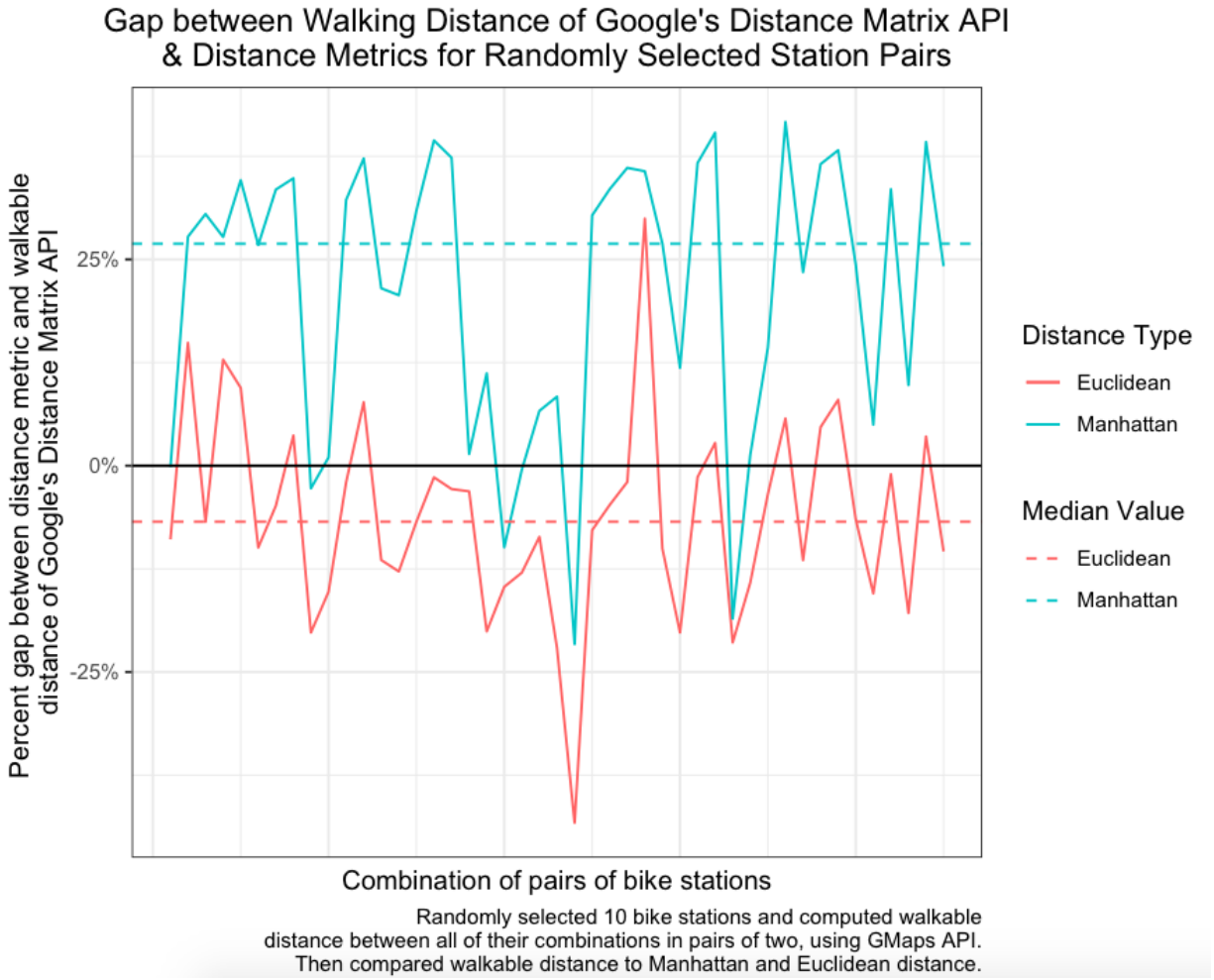


Figure 3.15: Percentage Gap in walkable distance between Distance type & Distance Matrix API (i.e. Google Maps).

While the gap between the suggested walkable path between two points coming from DMA and the Euclidean or Manhattan distance may be above 25% higher for some pairs, the Euclidean distance is always closer to the DMA distance than the Manhattan one.

To avoid outliers intervening in the analysis, we analyzed the median of the gap to the DMA calculation for the Euclidean distance, which was around -6% and around 21% for the Manhattan distance. This is the reason behind our choice to consider the Euclidean distance between stations and centers of demand as the distance metric for our model input. We have also chosen the kilometer scale given the size and scope of the problem and the city size.

3.4 Summary of execution pipeline

Before getting into the optimization results, we provide a brief executive summary of the process that was followed to build the dataset, assemble the inputs and construct the CFLP. The final output will give us an optimal solution showing how many stations we need to open to meet demand while minimizing the total system's cost. Figure 3.16 shows us the different stages of this thesis.

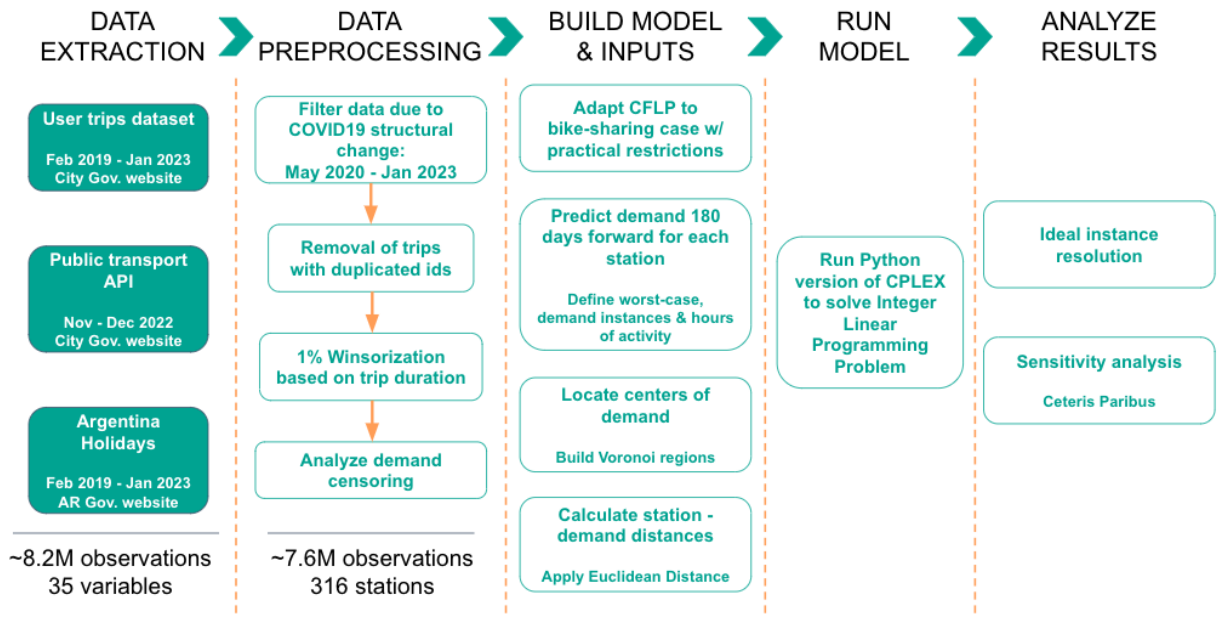


Figure 3.16: Summary of entire pipeline, from data extraction to results analysis. Solid boxes represent datasets, transparent boxes symbolize tasks being run on top of the datasets. Figure built by the author.

As shown throughout this thesis, we start extracting data from publicly available sources. We incorporate a trips dataset, that has the pickup and dropoff station that users ride to, along with the duration of their trip; the public transport API dataset that holds information about station capacity; the Argentina holidays dataset to analyze differences in trips per day due to public holidays.

Then we get into preprocessing the data: we only consider data after the COVID19 quarantine due to structural changes in the time series; we remove outliers based on duplicated trips and journeys that have a very long duration; we analyze demand to assess whether it's worth uncensoring.

Later, we adapt the CFLP problem to our bike-sharing case, adding some practical restrictions that will make our end result pragmatic. We then build the demand per station as the quantity of trips per day, and predict demand 180 days going forwards. We define the worst-case scenarios and keep demand for the mean and upper confidence interval of the prediction, for weekends

and weekdays. We locate demand centers using Voronoi regions around stations, allowing for each station to have their respective center of demand. Finally, we calculate stations to demand centroids distances, using Euclidean formulations.

In terms of modeling, we run the model we built via the Python interpreter of CPLEX, solving our Integer Linear Program and the inputs we built for it. In general, the model converges to a solution in under 20 seconds for each instance resolution, when using an Apple MacBook Pro from 2019, with 16 GB of RAM and 8-core Intel I9 processor.

Finally, we analyze results of the ideal instance that we will define below and run a sensitivity analysis to understand how much each parameter affects the end solution.

4. Results

4.1 Ideal Instance Resolution

All the work and research of the problem done in the previous section leads us to believe that the ideal instances to solve will be composed of the following parameters:

- $C_{i,j}$: variable costs measured in km, given the space and dimension of the city. Calculated using Euclidean distance.
- d_i: demand is predicted per active hour. We build four worst-case scenarios, combining the predictions for weekdays and weekends with the prediction's mean and upper confidence interval bound.
- k_j: capacity for each station extracted from API.
- T: walkable distance from station to center of demand restriction threshold set to 1 km. We do not want people to walk more than approximately 10 blocks for a bike ride. This is a decision we make for practical purposes, but it's an arbitrary model parameter that can be changed at will.
- F_j: fixed costs assumed to be 100. We will change this in the next section and run a sensitivity analysis on this parameter to see how it affects the final result.

This leaves us with a total of 4 instances (2 predictions of demand (mean, CI upper bound) * weekend or weekday) to solve. As a recap, our model will take the current scenario, where all bike stations are online and available for our users, and decide which of the existing stations should be turned off in order to minimize costs while satisfying the restrictions. All demand for bicycles should be met and can be shared across stations and users cannot walk more than 10 blocks from their center of demand to the station to grab a bike. As mentioned before, this is just a model parameter that we define to make our solutions become pragmatic, but it can be modified to whatever value the modeler choses. The solutions to this problem are reported in Table 4.1, along with the total stations that were opened and the improvement in cost they had.

Mean/CI Upper bound	Weekday or Weekend	Original Cost	Optimized Cost	Cost gain	Q Stations opened
Mean	Weekday	31,647	8,303	-73.76%	81 (25%)
Mean	Weekend	31,647	7,298	-76.94%	71 (22.5%)
CI Upper bound	Weekday	31,647	9,406	-70.28%	92 (29%)
CI Upper bound	Weekend	31,647	8,001	-74.72%	78 (24.6%)

Table 4.1: Results of ideal instances in terms of their minimized cost and the quantity of stations that were opened.

By using CPLEX as a mathematical programming solver to solve the CFLP model proposed in Section 3 for each of these instances, we obtain an improvement of above 70% in cost in all

cases, just by turning off facilities that could be repurposed while meeting demand. Furthermore, even in the worst case we could face, using the confidence interval upper bound and on a weekday, we can support demand per active hour with just 92 stations out of the total 316 opened today. This means that close to 71% of stations could be removed today without any harm to the level of trips.

Another way of looking at the same problem is through the lens of the load factor per station. Load factor can be defined as the demand that each station has to meet, divided by its total capacity. If our solution works, we should observe that the load factor per station increases at the optimal point, given that we are making the most out of the resources we have. Table 4.2 shows the increase in load factor for each of the demand instances we considered in our ideal instance resolution.

Mean/CI Upper bound	Weekday or Weekend	Original Avg. Load Factor	Optimized Avg. Load Factor	Load Factor Gain
Mean	Weekday	21.4%	70.7%	2.3x
Mean	Weekend	12.1%	52.3%	3.3x
CI Upper bound	Weekday	27%	78.5%	1.9x
CI Upper bound	Weekend	17.8%	64.9%	2.65

Table 4.2: Results of ideal instances in terms of their load factor gain.

The average load factor gain shows a clear increase, between 2x and 3.3x depending on the instance being analyzed. This is in line with our expectations, given that our system wants to minimize the total cost and will try to satisfy demand with the least amount of stations possible, while minimizing the user's travel cost to each station. However, the average can be a metric that is deceitful. Therefore, we also plotted the distribution of load factors per station for each of the instances, ensuring that our insights are robust.

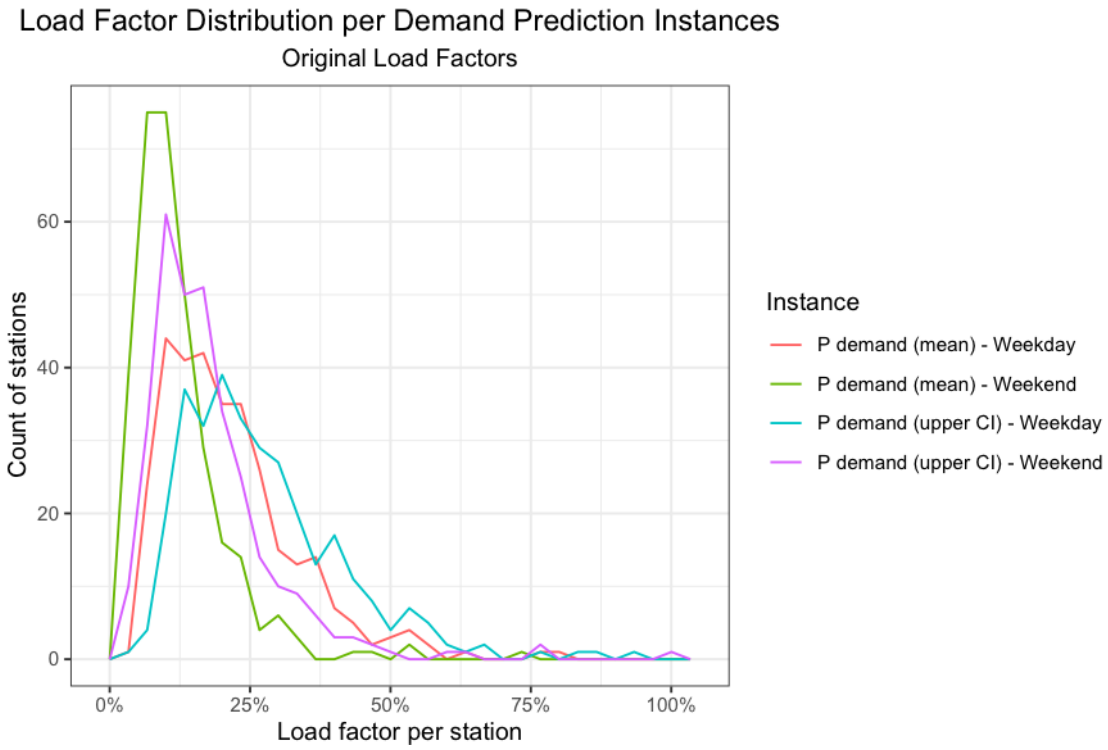


Figure 4.1: Load factor distribution per station for the original situation in our CFLP problem.

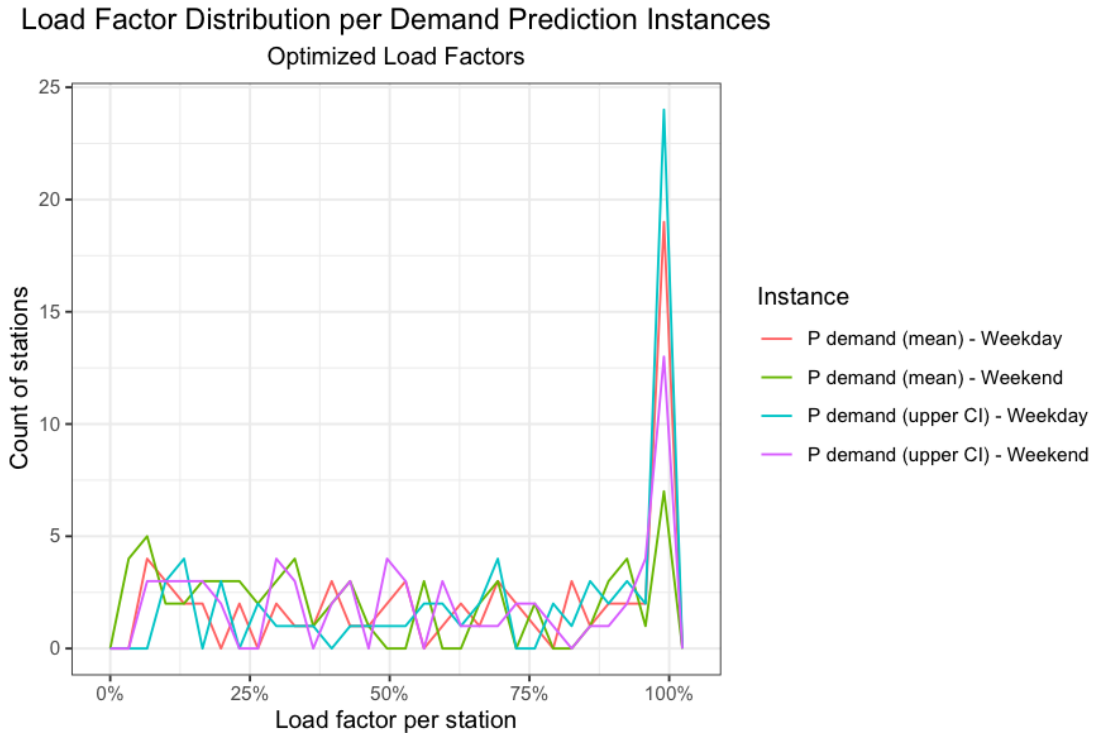


Figure 4.2: Load factor distribution per station for the optimized solution of our CFLP problem.

As observed in Figure 4.1 in the original state of nature, most of the stations are underutilized, even in the worst case scenario of the upper confidence interval demand prediction on weekdays. On the other hand, in our optimized network observed in Figure 4.2, most of the stations have a load factor of around 100% in all instances. There are also some load factors which we cannot maximize, which probably have to do with the 1 km radius walkable distance maximum restriction that we included in the model. Since there are no stations around certain demand points, those stations have to be turned on to guarantee that total demand will be satisfied.

Furthermore, our optimal solutions can be clearly visualized in Figure 4.3, with stations and centers of demand. We managed to locate them spatially based on their longitude and latitude, which makes it easier to see how the new network fits inside of the city. Blue nodes represent stations that the optimization decided to maintain from the original network. Their capacity is represented by the blue node size. Orange nodes represent centers of demand. The edges show which centers of demand are connected to each station.

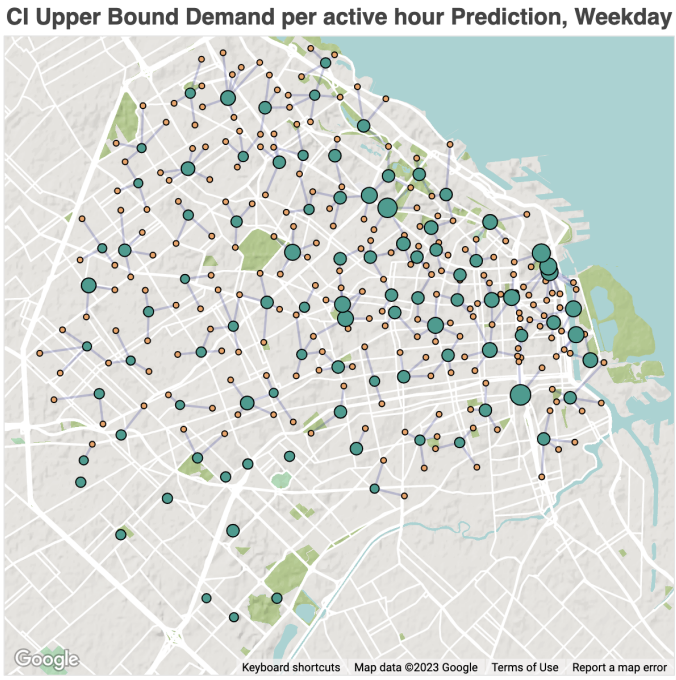


Figure 4.3: Optimal distribution of stations to address demand from centers. Stations are symbolized by blue nodes, orange nodes represent centers of demand. Using example for CI upper bound Demand prediction on weekdays. Stations that appear to have no edges are in fact connected to the center of demand that is closest to them.

Stations that are farther apart from each other, which is something typically observed along peripheral areas, especially towards the South of the city, will generally be active. This has to do with the 10 block walkable distance radius that we are imposing on the model. Beyond that obvious fact, it's interesting to see how the station density in the downtown area is reduced and becomes homogeneous across the rest of the city. This has to do with the fact that there was

not so much demand in that zone, so there is actually no need for so many bike-sharing stations. Furthermore, the stations in the downtown area have more capacity than other stations in general, shown by the node size. This makes it easier for them to support the existing demand.

We also compare the difference between weekdays and weekends for the same level of demand prediction. Our findings are located in Table 4.3, where we show the stations that are only present in the instance and the stations they have in common.

Mean/CI Upper bound	Weekday or Weekend	Q stations only present in day of week	Q stations shared between weekday & weekend	Q stations opened
Mean	Weekday	43	38	81
Mean	Weekend	33	38	71
CI upper bound	Weekday	33	59	92
CI upper bound	Weekend	19	59	78

Table 4.3: Comparing results between weekdays and weekends for the same level of demand.

The reduced amount of demand per active hour on weekends makes the system require less stations to satisfy the user requirement. However, that does not mean that the difference in total number of stations for weekends and weekdays are the only ones that will be positioned differently. In fact, approximately half of the stations are shared in the mean scenario, and more than 65% when considering the Confidence Interval upper bound scenario. This is evidence that the system is reoptimizing the stations that are opened in each of the cases to guarantee that demand is met as efficiently as possible. We visualize the location of all stations and the centers of demand they serve for weekdays and weekends in Figure 4.4. Then, we also indicate where the different stations are located in Figure 4.5.

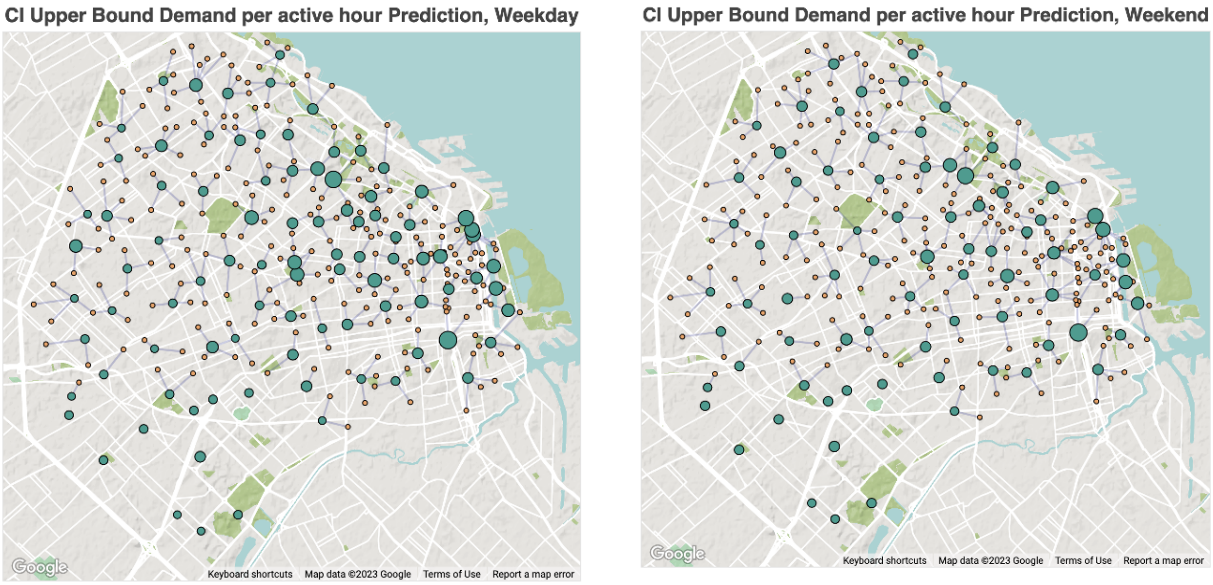


Figure 4.4: Comparing weekdays (left) and weekends (right) optimal station solution location of stations and how they tackle each center of demand. Stations are symbolized by blue nodes, orange nodes represent centers of demand. Using CI upper bound demand prediction on weekdays.

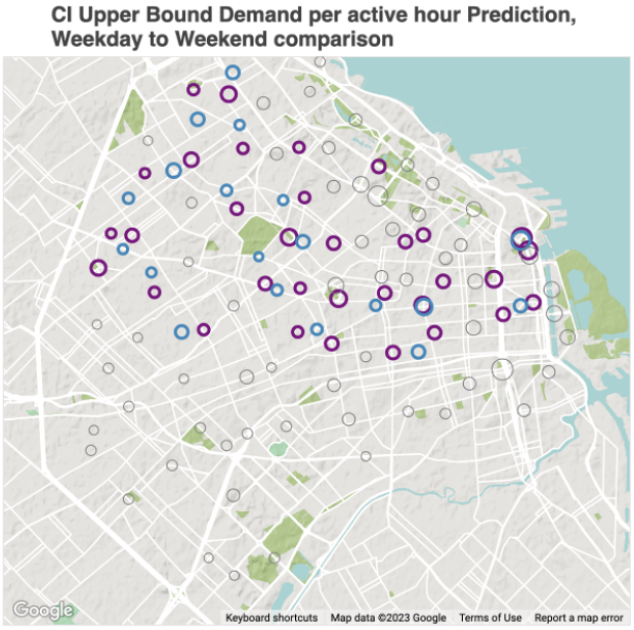


Figure 4.5: Analyzing differences between weekends and weekdays for CI upper bound demand predictions, to make them easier to spot. Purple nodes represent stations that are only present on weekdays, blue points are stations that are only present on weekends, and gray stations are stations in common. Using CI upper bound demand prediction on weekdays.

Figure 4.5 clearly shows that all the southern and northern borders of the city remain the same across weekends and weekdays. The big change is in the west border, and along the city center and downtown zone. The typical solution implies splitting demand into two for some cases

where demand is higher in that area on a weekday instead of just one bike-sharing station in the middle of both nodes, satisfying all constraints.

4.2 Sensitivity Analysis

After analyzing results for the ideal instance, we opted for running a sensitivity analysis over all of the data inputs that were required to build the instance. The concept behind this section of research is to determine how each input affects the final results in terms of the quantity of stations that are opened in the optimal solution. We complement this metric with the cost gain that comes from each optimal result, comparing it to the initial cost that the original state had, where all bike-sharing stations are open. Essentially, throughout each of these subsections, we are trying to answer the following question: “what happens to the optimal solution when we change this input, and everything else remains constant?”.
We will use the following input parameters for each of these sensitivity analysis, unless the input under scrutiny modifies it:

- No walkable distance constraint applied. This means that for all of the sensitivity analysis we will lift the restriction of the amount of blocks that a user has to walk to their station. This has to do with the fact that we would like to analyze the true elasticity of each parameter, without them being affected by practical matters.
- Variable costs: measured in km, using Euclidean distance.
- Fixed costs: set to $100. This is a relative value that has to be in proportion to the distance that users traveled, given that that depends on the share of demand that was allocated to each station, which is a continuous variable.
- Demand per active hour prediction, varying between mean & CI upper bound, and weekend or weekday.

4.2.1 Walkable distance restriction threshold

We started by assessing the impact that different thresholds had on the walkable distance. Recapping the restriction we included from section 3.1, what we are changing here is the parameter T, ranging from a radius of 500 mts to a 3 km radius.

(3) Walkable distance constraint: $x_{i,j} = 0 \;\; if \; C_{i,j} > T$

The changes in the number of opened stations and the gains in cost are reported in Figure 4.6. Cost gains are symbolized by straight lines and the quantity of stations that were opened optimally are represented by dashed lines. Each color represents a different demand prediction that we have discussed throughout the thesis.

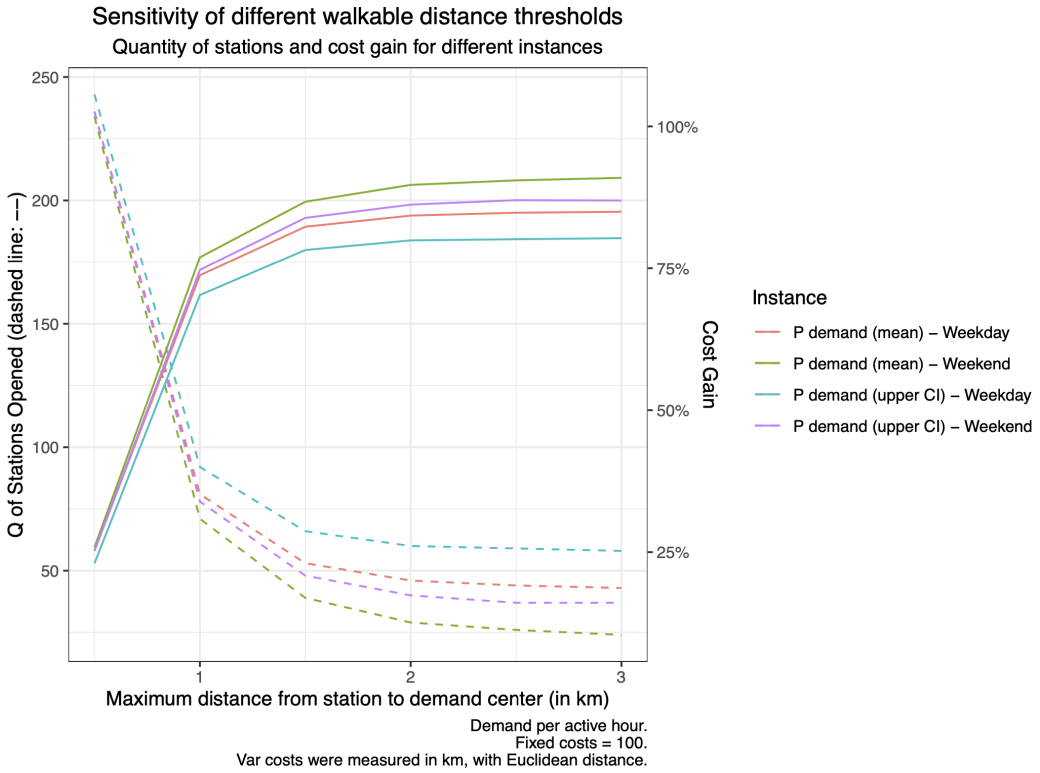


Figure 4.6: Sensitivity analysis for different walkable distance thresholds. Each color represents a demand prediction type. Cost Gain is represented by straight lines. Quantity of stations opened is represented in dashed lines.

As we relax the walkable distance constraint, the system finds solutions that have less stations active, given that the objective is to minimize the total cost while attending to all of the demand. Furthermore, there seems to be a steep gain in cost when the walkable distance is changed from 500 mts to 1 km. This is the main reason behind our choice of 1 km as a distance threshold when analyzing the ideal instance. In fact, the cost reductions show a diminishing return when larger thresholds are implemented.

On top of that, a threshold of 2 km means that there will be users that may have to walk up to 20 blocks for a bike ride from their nearest station. In practice, this is the same as saying that this user will not have access to a bike station and that not all demand will be met. The extreme case where there is no restriction on the distance that people have to walk from their center of demand to their nearest bike sharing station is shown in the right hand side of the next three graphs. We also show the different optimal solutions for the 500 mts and 1 km threshold that we mentioned previously.

Figure 4.7: Graphical representation of different walkable distance thresholds: 500 mts (upper-left), 1 km (upper-right), none (bottom). Stations that appear to have no edges are in fact connected to the center of demand that is closest to them. Using CI upper bound demand prediction on weekdays.

The network distribution for the 500 mts walkable distance threshold shows that close to 70% of total stations need to be opened. The reason is that most centers of demand are closest to their stations, and they cannot be satisfied by the next closest station, even though it may have idle capacity, as shown by the load factor analysis in Table 4.2. The problem with this is that demand per active hour is not as high as total capacity, so there will be many bikes left in the stations that people will never use. While five blocks is something manageable by people, this restriction also seems extreme, since people who want to go on a bike ride probably like to do physical activity and may be fine with walking a bit more.

On the contrary, when there is absolutely no restriction to the number of blocks that people have to walk to a station the optimization leverages the idle capacity as much as possible. It focuses on minimizing cost and generally opens big stations, because the fixed-cost of opening a station is assumed to be capacity-independent in our model. However, it opens enough stations to minimize the variable costs as well, balancing both objectives. Finally, as mentioned before, it connects centers of demand to stations that are very far away, which in practice is not realistic.

4.2.2 Fixed & variable costs

We explore the tradeoff between the two components composing our objective function. While this function is multi-objective and compares average distance traveled per trip (in km) and the budget needed to open stations (in dollars), we believe that this analysis is worth getting into. Eventually, if we had a metric of monetary cost per km, we could translate both amounts to dollars. Alternatively, we could also include an artificial coefficient to shift both terms to have the same magnitude.

Variable costs represent the average distance per trip between centers of demand and opened stations. Since our goal is to minimize costs, we would like to make the sum of average distances per trip to be as small as possible. The corollary of this is that variable costs will be minimized only when each station that is closest to every center of demand is opened. Our problem structure, where we have as many centers of demand as stations, implies that variable costs minimization will lead to all stations being opened.

On the other hand, we have our fixed costs, which are interpreted as the dollar amount that the owner of the bike-sharing system has to pay to open each station. Again, since our goal is to minimize costs, fixed costs will be minimal when we open as few stations as possible. In the edge case, where fixed costs have a higher weight than variable costs due to their magnitude, our problem would be solved with a single open station, assuming that the system needs to be operational.

Therefore, there is a clear tradeoff between both types of costs. Variable costs have a tendency to open more stations to be minimized, while fixed costs lead to fewer stations being open. We show the cost gain and quantity of stations opened for different scenarios and both types of costs in Figure 4.8 and Figure 4.9 respectively. We vary the scale in which variable costs are measured from kilometers * 100 to meters. We changed fixed costs from ten cents per station to $10,000. We replicated this analysis for each of the demand instances we have.

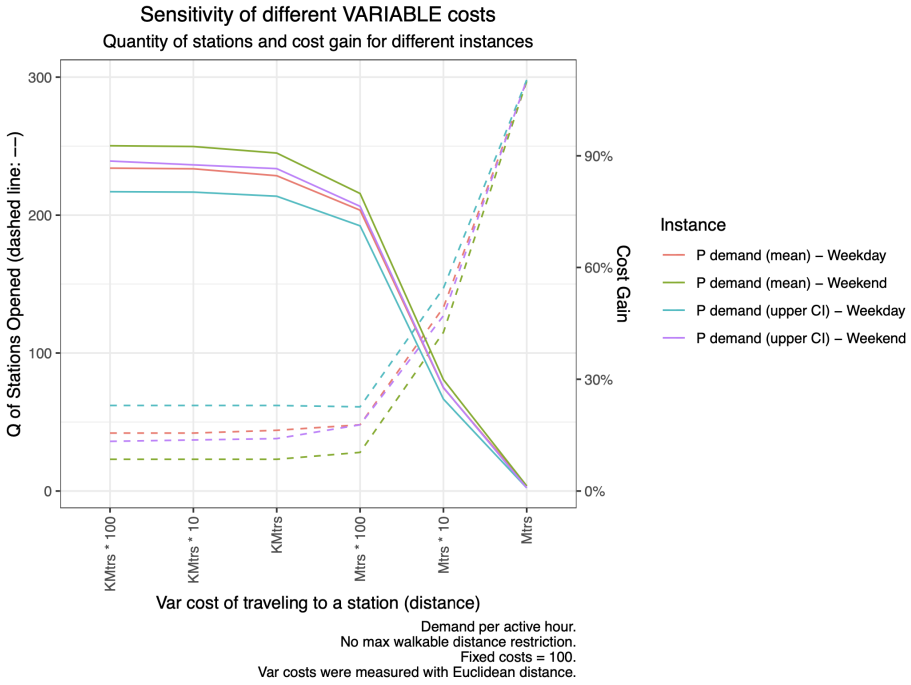


Figure 4.8: Sensitivity analysis for variable costs. Each color represents a demand prediction type. Cost Gain is represented by straight lines. Quantity of stations opened is represented in dashed lines.

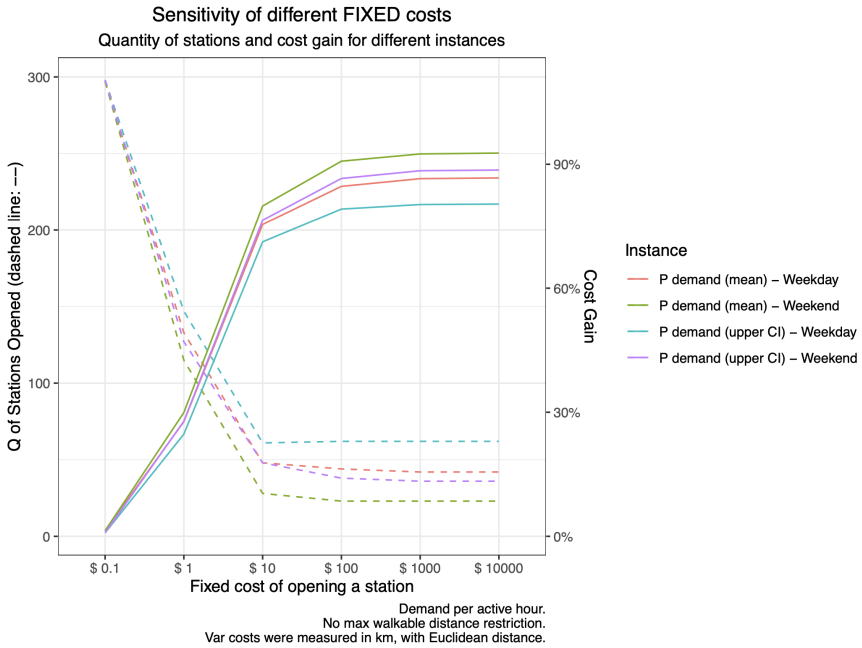


Figure 4.9: Sensitivity analysis for fixed costs. Each color represents a demand prediction type. Cost Gain is represented by straight lines. Quantity of stations opened is represented in dashed lines.

As expected, curves in Figure 4.8 and Figure 4.9 have opposite behaviors. The larger the variable costs, the more stations we opened and the lower the gains in costs. The curves for fixed costs show that the larger the fixed costs, the least amount of stations are opened in the optimal solution, minimizing costs.

These plots also show the main reason behind why we have chosen to represent variable costs in kilometers and have set fixed costs to $100. These scales seem to be the smallest scales by which the cost gains are stable. In other words, if we were to increase fixed costs from $100 to $1000, ceteris paribus, we would not see big improvements in cost gains comparing the two optimal solutions. The same goes for variable costs shrinking from being measured in kilometers to being measured in kilometers multiplied by ten.

4.2.3 Demand positive shocks

This final sensitivity check has to do with observing what happens when demand suddenly increases evenly amongst all centers. Conceptually, this can be tied back to the safety stock we should have in terms of active stations, in case people suddenly start going for more bike rides. Having safety stocks is a good idea given that opening a station is not instantaneous. In the case where there is a surge in demand, we would like to have bike-sharing stations in place to avoid demand exceeding the system capacity.

We replicated the steps in the aforementioned process. We increased demand in discrete intervals of 10%, homogeneously across all centers of demand and all demand instances, and mapped out the optimal solutions' gains in costs and number of stations opened. We observe sensitivity analysis results when no maximum walkable distance restriction is applied in Figure 4.10. We also implement the 1 km maximum walkable distance restriction and run the same sensitivity analysis in Figure 4.11.

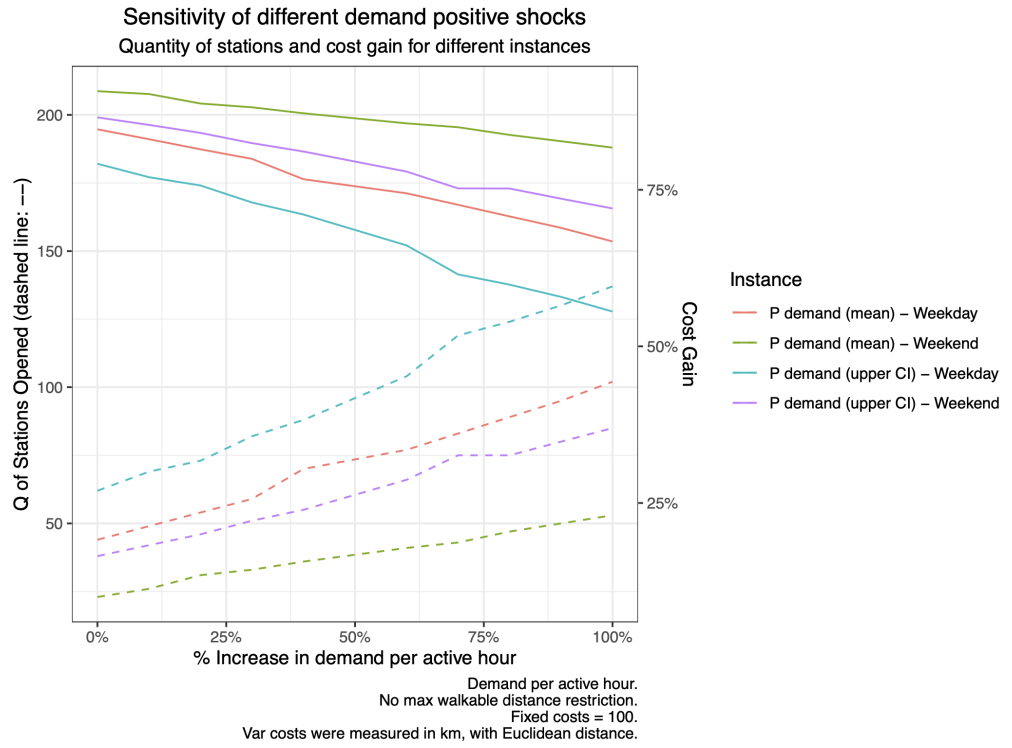


Figure 4.10: Sensitivity analysis for different demand positive shocks when no walkable distance restriction is applied. Each color represents a demand prediction type. Cost Gain is represented by straight lines. Quantity of stations opened is represented in dashed lines.

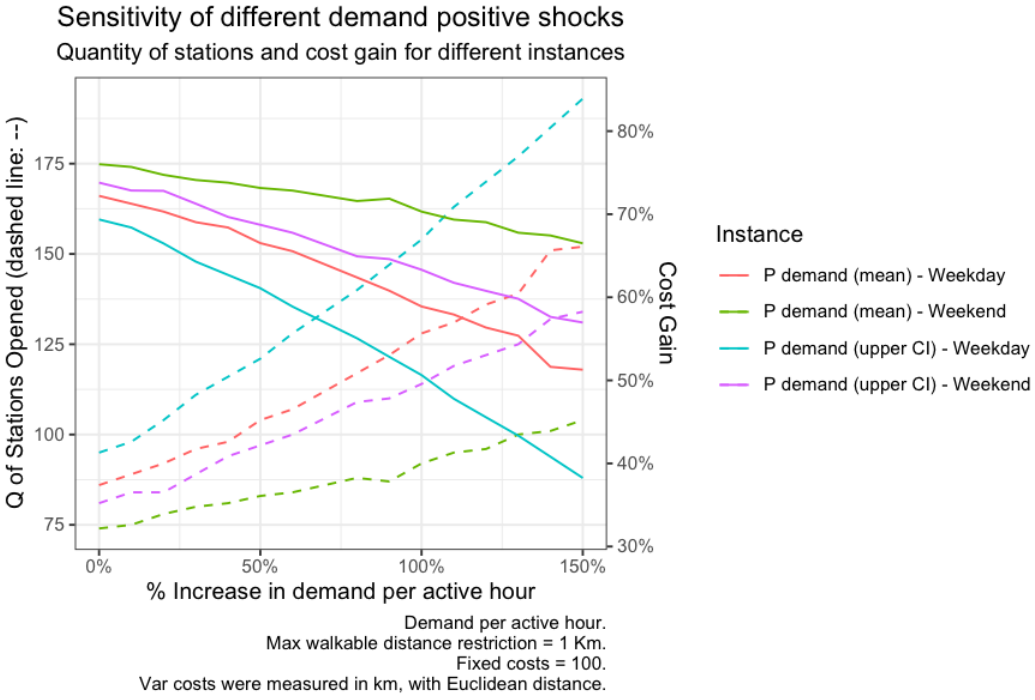


Figure 4.11: Sensitivity analysis for different demand positive shocks when 1 km max walkable distance restriction is in place. Each color represents a demand prediction type. Cost Gain is represented by straight lines. Quantity of

stations opened is represented in dashed lines.

Since the demand increases are evenly distributed across all centers of demand, we do not observe concave shaped costs gain curves as we did for previous experiments. As expected, the higher the demand, the more stations need to be opened to be able to meet it and the lower the cost gains will be. The spread across each of the demand instances has to do with the pessimistic nature of the CI upper bound on weekdays, compared to the average prediction on weekends. While they are both using the prediction with the maximum value for the next 180 days, the former has a demand per active hour that's generally much higher than the latter.

Furthermore, we compared how the optimal solution behaves when the 1km max walkable distance constraint is applied (Figure 4.11) and what happens when that is lifted (Figure 4.10). As expected, when the restriction is applied, cost gains are lower and the quantity of stations is higher. This has to do with the fact that we have to take into account how much users should move to get to their station as well as station idle capacity to decide which stations we should open. The more restrictions we include in the system, the less amount of costs it will be able to reduce. Since users cannot walk more than 1 km to a station from their center of demand, and stations are not evenly distributed throughout the city, there are some stations that need to be active to satisfy the entirety of the demand.

Moreover, it's interesting to analyze how many more stations are opened when the demand increases in a certain percentage. This seems quite linear for the case without any walkable distance restriction in Figure 4.8, in the upper plot, with a slope below 1. In layman's terms, this means that while demand can increase by 2x, opened stations will increase by less than 2x in this scenario. However, when the walkable distance restriction is applied, the quantity of station curves looks more exponential. Again, the reason behind this is that stations are not distributed homogeneously across the territory and the model needs to satisfy the entirety of the demand. Therefore, the slope is higher than the case with no walkable distance constraint, but is still not as high as the increase in demand, meaning that there are less stations opened than the increase in demand.

As a last step, we were keen on visualizing how the optimal network configuration spatially changed with demand increases, without the walkable distance restriction implemented. We plotted three scenarios in Figure 4.11: no changes to demand per active hour on the left; 2x increase in demand per active hour, on the right; 3x increase in demand per active hour, at the bottom.

Figure 4.12: Graphical representation of different demand positive shocks: no shock (upper-left), +2x positive shock to demand (upper-right), +3x positive shock to demand (below). Using CI upper bound demand prediction on weekdays.

Comparing the 2x case increase in the center with the regular demand per active hour, we observe that more facilities have been opened to satisfy the positive shock. Furthermore, there does not seem to be as many long connections in the plot on the right as there is in the plot on the left. Implicitly, the increase in demand made it so that the opened stations had to satisfy the demands that they had close by in order to minimize variable costs. This positive shock of doubling our demand estimation gives an optimal network setup that seems visually more practical than previous setups without any walkable distance restrictions. Naturally, variable costs will also increase when demand goes up, and probably more than fixed costs, given the slope of most cost gains and active stations curves comparing Figure 4.8 and Figure 4.9.

Since there is no restriction to the walkable distance in this ceteris paribus inspection, the plot on the top-right from Figure 4.12 also shows connections that would not be feasible in practice. Another thing worth noting is that since demand centers can be shared between stations, some stations that are close to a center do not have the capacity to fulfill the increase in demand, so the graph seems to show connections between stations. However, it still does not open all facilities since the optimizer calculates that costs can still be minimized by shutting down some of the existing stations and reallocating those centers of demand to previously opened ones.

In fact, when increasing demand per active hour, we noticed that the last feasible solution is achieved at an increase of ~3.5x. This means that the current active network, with all stations opened, is equipped to face up to a 4x surge in demand when no walkable distance restrictions are incorporated.

5. Conclusions

Cycling is an activity that nourishes the body and the mind. It is also a low-cost way of transportation when we can leverage bike-sharing systems that are implemented in the city. However, the facilities within our network are located in areas that do not need them, meaning that there is an opportunity to improve the current infrastructure.

Our solution ensures that people in peripheral areas have greater access to the system, guaranteeing that the current quantity of trips will not be reduced if stations are relocated. This is a win-win situation: it is good for people because they will have more transportation options and it is good for the bike-sharing business, because they will have an increased revenue from all the trips made by new users.

Our work shows sufficient evidence that there is an excess of capacity that is equivalent to a demand that could be up to four times higher, even thinking about a worst-case scenario. We have also shown that given the uneven distribution of facilities throughout the city of Buenos Aires, most of the stations in the downtown area can be turned off. Stations in peripheral areas should be kept open given that most of them are farther spread out and people use them.

In general, the model views stations that are turned off as a reduction in cost only if the demand from that station can be met by another station that is not more than 1 km away. Beyond the conservative estimation of 70% cost reduction, these spaces could be repurposed by the owner to make a higher profit from them. For example, bike-sharing stations that are closed by the model could be reinvented to be safe bicycle holders for all other cycling users that own their bicycle and need a place to park it.

Furthermore, the stations that are closed in downtown areas could be simply relocated to areas that have a much lower density of stations, especially towards Southern areas of the city of Buenos Aires. After placing stations in these new areas that currently have a high usage of the facilities they have, this model could be run again. It could even become an iterative process: when stations are opened in zones with no facilities in a radius of ten blocks, demand is predicted based on usage and the model is reoptimized once again.

Going beyond the macro result, other facility optimizations could be done at a weekly level. Emptying stations is another way of closing them temporarily. Albeit more tactical, there is also evidence for the owner to optimize where bikes should be deployed depending on the moment of the week. This could also allow the business to fix bicycles that require a trip to the mechanic, ensuring that demand is consistently met to the highest standards of service.

5.1 Future work

One of the assumptions behind this thesis is that stock is equal to capacity, meaning that all docks within a station are full with all the bicycles they can have. Therefore, the first avenue for future work could be to relax this assumption and incorporate bicycle stock as another variable in the model. Including this variable would also imply that there is a reason to avoid having less

stock than total capacity, which means that something would have to be done with the rationale behind using free station docks to deposit bicycles. This could be tied back to the demand function we built as well, actually predicting only extractions instead of the maximum between extractions and deposits for each day. One could even predict what is the trend for deposits and incorporate a new restriction within the model that makes it so that the demand for bicycle deposits is also met.

Another front that could be worth exploring is to expand this model to make fixed costs changeable, so that they depend on the capacity that each station has. This would be more realistic, given that building a bike station with 10 docks and bicycles should not be the same as installing a station with 100 docks and bicycles. Moreso, capacity could even become a new decision variable for the model to decide which stations it should open, and what their capacity should be. Even thresholds of walkable distances could become variable depending on where centers of demand are located. We could expand the walkable thresholds in zones with a lower density of stations to allow more flexibility in the model's result.

The last assumption from the model worth lifting is the facility locations. The model we used assumes that the station position is given and only decides whether the facility should be turned on or off. Potentially, a random positioning algorithm could be used to build different sets and decide which configuration is optimal given the demand. That way we could incorporate another level of stochastic decision making that could make the model more robust with respect to locating the actual demand position, which is endogenous to where stations are currently located.

Finally, further objective functions could be used to describe a different central planner. For the sake of this work, we focused on minimizing cost. However, the central planner could be more interested in maximizing their profit, which involves including a form of revenue model. The bike-sharing system in the city of Buenos Aires has only recently started including a segmented tariff. A fixed price per bike ride is charged only to tourists any day of the week and to locals during weekends. We could estimate different demands for each of them, trying to enrich the data via some other sources that were not available for this work, incorporating the income coming from each trip.

6. Appendix

6.1 Public Transport API query result

In order to allow for future work on this subject, we add the query results in the following table, showing the capacity of each station and their latitude and longitude. These results come from the last time the API was consumed, on Dec 25th, 2022.

Station id	Latitude	Longitude	Capacity
2	-34.59242413	-58.37470989	36
3	-34.611032	-58.3682604	20
4	-34.601822	-58.368781	20
5	-34.5805497	-58.4209542	42
6	-34.628526	-58.369758	20
7	-34.606498	-58.381098	16
8	-34.6094218	-58.3893364	24
9	-34.585443	-58.407741	24
12	-34.5927096	-58.388807	16
13	-34.61009	-58.406	30
14	-34.577424	-58.426387	30
17	-34.6064101	-58.4187306	20
21	-34.640111	-58.406432	24
22	-34.5938629	-58.3825498	20
23	-34.600139	-58.379836	12
24	-34.610583	-58.3808943	18
25	-34.5894269	-58.4161178	24
26	-34.600752	-58.3638723	30
27	-34.599068	-58.3900887	16
29	-34.6079414	-58.4335573	30
30	-34.5908211	-58.3973698	20
31	-34.6033431	-58.439521	16
32	-34.6072074	-58.3735984	16
33	-34.5970909	-58.3989807	20
35	-34.5964246	-58.371847	32
36	-34.6045481	-58.3767677	16
38	-34.5970497	-58.3828403	20
41	-34.6371232	-58.4058883	20
43	-34.584018	-58.389921	28

44	-34.5755148	-58.4138829	20
45	-34.6018635	-58.3866934	20
49	-34.6290527	-58.422611	16
50	-34.5837348	-58.4010798	20
51	-34.6014776	-58.3821261	12
54	-34.5982097	-58.4220694	16
56	-34.588567	-58.425999	16
57	-34.6126898	-58.37125	12
58	-34.5752773	-58.4346883	20
59	-34.617654	-58.380565	20
60	-34.6016509	-58.371079	20
61	-34.6189273	-58.5051769	12
63	-34.5986	-58.373062	24
64	-34.5936508	-58.3941087	20
65	-34.5873124	-58.4157873	20
66	-34.5945475	-58.4138713	20
68	-34.552148	-58.480464	16
69	-34.5961006	-58.4046092	16
70	-34.5926862	-58.4260597	20
71	-34.6026673	-58.3833559	30
73	-34.6306814	-58.3718235	16
74	-34.60439	-58.43454	30
75	-34.6122976	-58.3989871	20
76	-34.6074084	-58.3950548	20
77	-34.581135	-58.501487	16
79	-34.61189	-58.36393	30
80	-34.6245807	-58.4341232	12
82	-34.6078917	-58.4263947	20
83	-34.603269	-58.3893728	28
85	-34.5948057	-58.4091784	20
86	-34.6212681	-58.4016808	20
87	-34.619845	-58.4314942	16
89	-34.5825475	-58.4056671	20
91	-34.6174482	-58.397602	20
92	-34.6316444	-58.4053386	20
93	-34.620798	-58.3944635	20
94	-34.591511	-58.449652	30

95	-34.6021121	-58.3781678	16
96	-34.6027814	-58.4116586	20
98	-34.6081643	-58.3779002	16
99	-34.5960961	-58.435408	20
101	-34.5891857	-58.4424397	20
102	-34.5851209	-58.4492989	12
104	-34.587617	-58.455212	30
107	-34.63037718	-58.395844	16
111	-34.6054877	-58.3646858	30
112	-34.612075	-58.380384	20
114	-34.5949745	-58.3722554	32
116	-34.5921708	-58.4025894	12
117	-34.6201008	-58.3741759	16
118	-34.6170196	-58.4026531	20
120	-34.617509	-58.4092876	20
121	-34.6011732	-58.4285093	20
122	-34.5915614	-58.4198163	16
124	-34.580538	-58.411965	20
126	-34.6402672	-58.3692243	16
128	-34.60515159	-58.36882117	28
130	-34.5917376	-58.37436403	40
131	-34.5984043	-58.3990158	16
132	-34.6033685	-58.372763	12
134	-34.683188	-58.468952	12
135	-34.595125	-58.377535	20
137	-34.6155977	-58.3674923	24
138	-34.6353595	-58.3876724	12
144	-34.6018744	-58.4060944	20
146	-34.6221118	-58.4078419	20
149	-34.6153266	-58.3813642	16
150	-34.6187547	-58.3554654	36
151	-34.6118145	-58.361285	24
152	-34.6181645	-58.3596311	28
153	-34.6307765	-58.3620701	16
155	-34.6380383	-58.4114346	16
156	-34.5775895	-58.4074696	20
158	-34.592735	-58.4450697	20

161	-34.6020779	-58.4196761	20
162	-34.608985	-58.401924	20
163	-34.6095663	-58.4064308	30
164	-34.617301	-58.3698984	20
165	-34.597048	-58.407614	12
166	-34.58834666	-58.39414794	12
167	-34.606984	-58.44854	16
168	-34.6186217	-58.3812271	16
169	-34.6123459	-58.411856	12
171	-34.6032813	-58.3997553	20
172	-34.625426	-58.371082	16
174	-34.597225	-58.391768	20
175	-34.626851	-58.380707	48
176	-34.555254	-58.494845	16
177	-34.568165	-58.412121	12
179	-34.6384786	-58.3642885	16
181	-34.5926649	-58.4120072	20
182	-34.5780479	-58.4352466	20
183	-34.6156994	-58.3899728	28
184	-34.6306129	-58.3913419	20
186	-34.6136356	-58.4064415	20
187	-34.552571	-58.450897	16
188	-34.62393	-58.39125	24
189	-34.5886889	-58.3852113	20
190	-34.5850763	-58.4111136	20
191	-34.6079305	-58.3808358	16
193	-34.5908626	-58.4061652	20
194	-34.6060758	-58.4224635	16
196	-34.6275351	-58.3657211	20
197	-34.620998	-58.493044	20
199	-34.6222601	-58.4160137	20
200	-34.5890696	-58.4053617	16
202	-34.583749	-58.390602	30
203	-34.628757	-58.356259	24
204	-34.614948	-58.427818	24
205	-34.583323	-58.428016	16
206	-34.58495	-58.437339	16

207	-34.65237685	-58.48735936	16
208	-34.5807161	-58.438404	8
210	-34.572165	-58.411278	20
212	-34.600275	-58.434875	20
213	-34.599659	-58.442685	20
215	-34.585878	-58.424996	20
216	-34.589968	-58.411493	20
219	-34.6360274	-58.4156332	16
220	-34.635128	-58.427573	16
222	-34.572583	-58.420628	20
223	-34.6234	-58.424853	16
227	-34.61036075	-58.43276297	20
228	-34.6164879	-58.3656683	16
229	-34.581576	-58.45153	12
230	-34.5678255	-58.4645037	16
231	-34.60511787	-58.44599959	16
232	-34.5599779	-58.4790578	20
234	-34.5480059	-58.4469439	20
235	-34.573734	-58.48692409	24
236	-34.562161	-58.455166	16
237	-34.63670936	-58.50135526	16
239	-34.56533652	-58.42062076	20
241	-34.600874	-58.494123	12
242	-34.57716	-58.403214	20
245	-34.552594	-58.4429397	16
247	-34.5838957	-58.466494	18
248	-34.573522	-58.474635	20
251	-34.64485798	-58.40974866	12
252	-34.647121	-58.374336	16
253	-34.6163544	-58.4170737	16
254	-34.620717	-58.441607	16
255	-34.622092	-58.448547	16
257	-34.570825	-58.481236	24
258	-34.565521	-58.455334	16
259	-34.5591522	-58.4441762	16
260	-34.55307941	-58.43522349	12
261	-34.616151	-58.440584	16

262	-34.609761	-58.467476	16
263	-34.622003	-58.457555	12
265	-34.64196	-58.4505	16
267	-34.5976121	-58.4985424	16
268	-34.5503	-58.477	36
269	-34.577329	-58.457799	18
270	-34.640114	-58.43026	20
271	-34.630108	-58.473844	20
273	-34.616758	-58.446751	16
275	-34.562277	-58.459289	20
277	-34.563539	-58.436115	12
278	-34.564122	-58.469813	16
280	-34.633528	-58.449379	24
281	-34.613778	-58.458315	24
284	-34.631018	-58.435056	20
289	-34.559801	-58.448314	24
291	-34.617247	-58.381627	16
299	-34.631705	-58.466143	20
301	-34.66036166	-58.46763869	20
302	-34.6518464	-58.415771	16
304	-34.5893	-58.4848	16
307	-34.6499971	-58.424773	12
308	-34.567633	-58.436752	20
309	-34.62141207	-58.51978099	16
310	-34.67711802	-58.47562444	12
311	-34.5972098	-58.47421143	20
316	-34.605567	-58.453475	16
318	-34.603936	-58.457317	20
322	-34.551304	-58.454181	20
323	-34.565409	-58.459298	24
324	-34.578933	-58.4840556	16
327	-34.67713052	-58.45428605	16
329	-34.593141	-58.435187	20
330	-34.6008306	-58.4721271	12
333	-34.6008	-58.50335	20
335	-34.615945	-58.47098	20
336	-34.5997308	-58.5111458	28

340	-34.54754	-58.467844	20
342	-34.616813	-58.484297	20
348	-34.610482	-58.474369	12
349	-34.5703869	-58.4663122	16
353	-34.599036	-58.364695	28
355	-34.567483	-58.446381	16
358	-34.561486	-58.465586	12
359	-34.555602	-58.450479	20
361	-34.569187	-58.453608	20
362	-34.5658514	-58.479847	12
363	-34.6439	-58.463114	16
366	-34.6352784	-58.4825327	12
367	-34.61621214	-58.47720947	16
368	-34.59807	-58.482079	12
369	-34.592244	-58.491797	16
370	-34.590964	-58.500336	20
371	-34.6317	-58.45534	12
372	-34.636406	-58.470136	12
373	-34.58233692	-58.48108315	16
374	-34.6471	-58.4698	16
375	-34.60459736	-58.48481212	16
376	-34.626778	-58.487078	20
378	-34.623123	-58.468287	20
379	-34.5602	-58.4281	20
381	-34.6162466	-58.4658836	12
382	-34.60547	-58.47739	16
383	-34.5717986	-58.4895409	16
384	-34.57951881	-58.46183473	18
385	-34.60615991	-58.49314135	16
386	-34.5654	-58.4759	8
387	-34.602028	-58.465568	16
392	-34.554581	-58.485381	16
393	-34.59027327	-58.4669322	12
395	-34.59684929	-58.45327986	16
400	-34.57899	-58.46982	16
403	-34.64552326	-58.39666367	20
407	-34.6424248	-58.478266	16

408	-34.56912	-58.50025	16
413	-34.6614271	-58.5015315	16
416	-34.553262	-58.469133	28
417	-34.57634	-58.50248	20
418	-34.5809383	-58.4445804	16
420	-34.5446	-58.4396	16
422	-34.56454	-58.50289	16
423	-34.559255	-58.487772	12
424	-34.638584	-58.39965	16
425	-34.6440763	-58.422091	16
426	-34.5557	-58.4579	20
427	-34.648402	-58.5136448	16
428	-34.587458	-58.4739506	20
429	-34.63897407	-58.51010242	16
431	-34.542628	-58.436913	20
432	-34.619879	-58.435801	20
433	-34.637697	-58.373726	24
434	-34.54569	-58.46514	16
435	-34.544503	-58.459499	16
436	-34.57653	-58.44349	12
440	-34.5906475	-58.428899	16
441	-34.609801	-58.3748	20
444	-34.608936	-58.370716	24
448	-34.58226812	-58.37909621	24
449	-34.62883578	-58.46329738	24
453	-34.6294802	-58.4944854	16
454	-34.63403	-58.50694	12
455	-34.62652	-58.50805	16
457	-34.62805	-58.52174	16
458	-34.636275	-58.444041	20
459	-34.61072	-58.51914	16
460	-34.6075	-58.51193	16
461	-34.58088	-58.49363	16
464	-34.541	-58.4441	28
465	-34.5904625	-58.5071166	12
466	-34.608096	-58.4118397	16
467	-34.6055135	-58.3958925	12

468	-34.6042288	-58.3937375	8
469	-34.5934185	-58.513837	12
471	-34.5947293	-58.5033086	12
473	-34.6134032	-58.4933583	12
474	-34.6108915	-58.50302	12
475	-34.6165412	-58.5260181	12
476	-34.6182924	-58.4985253	12
477	-34.6350046	-58.3950539	12
478	-34.572056	-58.4475236	12
479	-34.6262877	-58.4553001	12
480	-34.6293335	-58.4836028	12
481	-34.6430378	-58.5127001	12
482	-34.550468	-58.430024	12
483	-34.5436888	-58.4771259	12
484	-34.5554843	-58.4768041	12
485	-34.553465	-58.42144	12
486	-34.5420905	-58.4705699	12
487	-34.558137	-58.4672593	12
488	-34.5657087	-58.4952184	12
489	-34.5476039	-58.4565662	12
490	-34.56477054	-58.402021	16
491	-34.6122963	-58.443295	8
492	-34.6097105	-58.4215633	16
493	-34.5967435	-58.4594031	8
494	-34.6147875	-58.5116722	12
496	-34.5831596	-58.5131664	12
497	-34.5744245	-58.4962318	12
498	-34.5865976	-58.4949585	12

Table 6.1: Public transport API query result on Dec 25th, 2022. Includes a total of 316 bike-sharing stations and their capacities.

7. References

1) Daskin, M., (2013). Network and discrete location: models, algorithms and applications. Second Edition, Wiley Online Books.
2) Dell'Amico, M., Hadjicostantinou, E., Iori, M., Novellani, S., (2013). The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. Omega 45 7–19.
3) Freund, D., Henderson, S., O'Mahony, E., Shmoys, D., (2019). Analytics and Bikes: Riding Tandem with Motivate to Improve Mobility. Informs journal of applied analytics, vol 49, no. 5.
4) García-Palomares, J.C., Gutiérrez, J., Latorre, M., (2012). Optimizing the location of stations in bike-sharing programs: A GIS approach. Applied Geography 35, 235-246.
5) Gonzalez, F., (2017). Using bike share and the subway to commute. Published in their personal blog.
6) Liu, J., Li, Q., Qu, M., Chen, W., Yang, J., Xiong, H., Zhong, H., Fu, Y., (2015). Station site optimization in bike sharing systems. IEEE International Conference on Data Mining, Atlantic City.
7) Martinez, L., Caetano, L., Eiro, T., Cruz, F., (2012). An optimization algorithm to establish the location of stations of a mixed fleet biking system: an application to the city of Lisbon. Procedia – Social and Behavioral Sciences 54, 513-524.
8) Nikiforiadis, A., Aifadopoulou, G., Salanova Grau, J.M., Boufidis, N., (2020). Determining the optimal locations for bike sharing stations: Methodological approach and application in the city of Thessaloniki, Greece. In Proceedings of the 23rd EURO Working Group on Transportation, Paphos, Cyprus.
9) Ohana, M., (2021). Predicción de demanda en el uso de bicicletas públicas de CABA. Published as a thesis for the Master in Management and Analytics, UTDT.
10) Snyder, L., Shen, Z. (2019). Fundamentals of Supply Chain Theory. Wiley, second edition, Hoboken, New Jersey. Chapter 8: Facility Location Models. Pages 267-355.
11) Taylor, S. J., Letham B., (2017). Forecasting to Scale. PeerJ 5 Preprints: e3190v2.
12) Wolsey, L. (2021). Integer Programming. Second Edition. Formulations, pp. 1-5.