THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

Ridership Recovery and the New York City Subway

Candidate Number: 50766

August 2023

Word count: 9861

GY485 - Dissertation - MSc Geographic Data Science

Contents

I. Abstract

The COVID-19 pandemic caused many public transit systems to face drastic ridership reductions and troubled finances due to the loss of fare revenue. Researchers have long discussed the frequency of transit service as a driver of public transit ridership, however, post-pandemic public transit ridership trends have altered the way people live and commute. Therefore, using newly released spatial data on New York City, this paper seeks to measure the relationship between train frequency and ridership in regard to hourly, weekly, month and yearly time variations across the NYC Subway. The results of this paper attempt to show that ridership growth is caused by an increase in Subway train frequency, particularly on the weekend where ridership return has been the strongest on the Subway. Furthermore, this ridership return growth correlates with NYC neighborhoods that have higher incomes, gainful employment, and population density. Therefore, the paper concludes transit planning policy initiatives should consider focusing on expanding service to residents in the urban core, by methods such as using underutilized urban commuter rail infrastructure or upgrading of subway signaling to allow for higher frequencies. Likewise, the paper concludes further research should consider expanding land use reform to near suburban areas to induce further long term ridership growth and looking at subway fare reform to attract more lower income riders as a part of the green transition and climate resilience.

II. Introduction

The COVID-19 pandemic had profound impacts on transportation in urban environments. Across the world, millions of people stopped commuting via public transit to their jobs in their city because they began working from home or shifted to alternative forms of transit such as driving (Young et al., 2021)(Zipper, 2021). This led to temporary and chronic financial strain on existing public transit systems across the world and especially amongst cities in the United States, where bailouts were needed to not jeopardize transit operations funding (Shepherdson, 2020). However, many American cities and some other cities across the world, have yet to recover the prior to the pandemic ridership on their public transit systems as of 2023 (Zipper, 2023). This can lead to more driving, and more congestion and emissions that can have the adverse effects of worse pollution and climate change outcomes in the near to long term (Popovich and Lu, 2019). This has also led to significant long term financial strain on public transit systems all over, as fare revenue remains lower due lower ridership (Lee, 2023). However, the likely continued work from home company policies and the rise of "hybrid" work means that on a per capita basis people commuting to work via public transit, and in general, may never return to pre-pandemic levels (Sahadi, 2023). This has led many transit agencies to restructure their approach to public transit. Traditionally, public transit service operated mostly at peak commuting hours in the morning and evening (Walker, 2023), this is perhaps highlighted by the synecdoche often used in the US to describe what other countries would call suburban or regional rail: 'Commuter Rail' (Walker, 2021). However, with the emergence of hybrid work, and significant ridership returns off-peak and on weekends, transit agencies have been rethinking their approach towards commuter oriented services towards all day frequent service to match off-peak and weekend ridership growth (WMATA, 2023) (Hicks, 2023). One such example is the New York City Subway.

New York City has by far the highest public transit usage in the US on an absolute and per capita basis (NTD, 2020). However, ridership on the Subway has remained below 2019 levels for the last 3 years. Nevertheless, ridership recovery has been steady, and strongest off peak, and especially on weekends (Duddridge, 2023). Using newly produced powerful ridership data from the Metropolitan Transportation Authority (MTA) that operates the New York City Subway, this paper seeks to analyze the relationship between ridership return and increases in train service by comparing equal months between 2022 and 2023. The structure for analyzing this relationship is as follows: First, it will discuss the relevant literature towards metro systems' ridership and their services, how they interact with the built environment, and their effect on the local economies of their respective cities. Second, it will discuss the background and layout of the New York City Subway, how its layout and services differ from other global metro systems, and how that will be important to interpreting the data. Third is the data section, which will discuss the summary statistics of the Subway's ridership and performance and the economic indicators of neighborhoods of the city it runs through. In the fourth part, the paper outlines the rationale for determining the Ordinary Least Squares (OLS) model specification, the flaws of that model, and why a Fixed Effect (FE) Model was selected as a better method for determining causal relationships between Subway ridership and the number of trains per hour, along with a robustness check for said relationship. Last are the Results and Conclusion Sections which will discuss the limitations of the data and both the OLS and FE Models, and what the results mean for policy prescriptions towards public transit going forward in a post-pandemic and green-transition economy.

III. Literature Review

Much in terms of transport literature has been written about the impact of railroads and metro systems. Since the 19th century, the initial introduction of railroads has been associated with reduced trade costs and rises in real incomes (Donaldson, 2018). Entering the 20th and 21st centuries, railroads, in the form of metros, have drastically shaped the urban environment (Wu and Hong, 2017), and have been leading infrastructure in carbon emissions from electrification of public transit and better land use (Jing et al., 2022).

In regard to passenger rail, train improvements, such as Wi-Fi, also appear to have a positive effect on ridership (Dong et al. 2013). Congruently, studies suggest that public transportation equity via commuting access improves job accessibility and economic flexibility (Kawabata and Shen, 2007). Likewise, higher ridership also is associated with special events such as concerts (Santanam et al., 2021), highlighting that trains as public transport also cater to not just commuting but also leisure ridership. Furthermore, on top of event studies, transport literature has focused on using machine learning to build ridership forecast models that account for population within a certain distance of stations as a way to predict ridership (Li et al., 2016).

Recently, public transit literature has been focusing on the effect of the pandemic on ridership, of which studies find that the pandemic left mostly vulnerable populations as riders of public transit (Liu and Miller, 2020), and how COVID was limited by changes in metro systems' service policies (Xiang et al., 2021). Given that the pandemic greatly affected ridership, this paper seeks to offer new insight into modeling public transit ridership return rather than the existing literature prior to the pandemic new ridership growth studies. Additionally building off of existing distance based spatial analysis (Nelson and Hibberd, 2023) and focusing on the variance of peal/weekday and off peak/weekend ridership, this paper attempts to analyze the growing change in rider preferences since 2022.

IV. Background

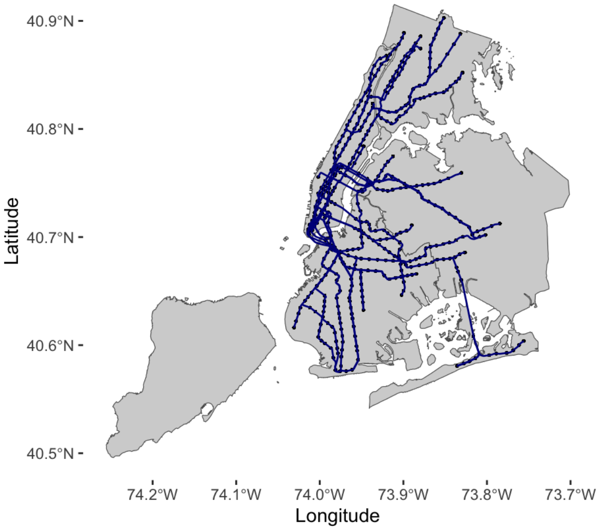
The New York City Subway is the largest and busiest metro system in not just the United States of America but also all of North and South America (Hutt, 2018). The system is also in the top 15 longest metro systems globally. With 424 stations, it is also the metro system with the most stations as of 2022 (TBS, 2022). Prior to the pandemic, the Subway also saw daily weekday ridership average well over 5 million riders, making it also one of the most used metro systems in the world (MTA, 2020).

In the city, the Subway operates in 4 out of the 5 boroughs that make up New York City: Manhattan, Brooklyn, Queens and the Bronx, while the 5th borough Staten Island has its own single line metro system called the Staten Island Railway (SIR). The Subway currently operates 28 services assigned a number or letter across 36 lines with all services, except the 'G' train service and 2 short shuttle services titled 'S', serving Manhattan (MTA, 2020). New York City also has another metro system beyond the Subway and SIR called the Port Authority Trans Hudson (PATH) that operates 13 stations split between New Jersey and Manhattan (PANYNJ, 2023). Furthermore, The Subway is operated as a subsidiary of the Metropolitan Transportation Authority (MTA) that also operates the city's buses, many roads and bridges, and the commuter railroads of Long Island Railroad (LIRR) and Metro North Railroad (MNR) (MTA, 2022).

The Subway also has many features that make it unique amongst the world's metro systems. It operates on flat fares not tied to a distance or zone, and it is one of the only metro systems that never closes, operating 24 hours a day, 7 days a week, 365 days a year. Moreover, while smaller cities like Copehagen due operate 24/7/365, whilst other large cities like Chicago operate a partial system 24/7/365, and other cities like Tokyo operate 24/2/365 on weekends, the Subway is the only major metro system in the world to operate almost fully without closing (Wright, 2013). It is able to do this because the system makes extensive use of quad tracking and track sharing. Quad tracking refers to 4 tracks being used for service, while this is often present on many mainline rail services globally, very few cities incorporate 4 tracks into their metro systems, and none do as much as New York City. Because of this quad tracking, maintenance is also easier to do without closing the Subway due to the other mentioned unique feature of the subway: track sharing (Levy, 2017).

In most metro systems, the simplest form of metro line is a single line between two termini, with a series of stations along the way. The second simplest form of metro line is a metro line between three or more termini in the form of branches. At its simplest, it includes 1 line with 1 branch and 3 termini that form something like a "Y" shape. However, this can be scaled up to any number of complicated service patterns with *n* number of branches and *n+2* number of termini, but nevertheless the most common around the world is single or double branched lines. However, the New York City Subway, forms a more complicated network that incorporates combined branching outside of a city center, so that 2 separate lines in Manhattan share tracks on branches outside of Manhattan; this can be illustrated in a "YY" shape, as shown in Figure 1 below:

Figure 1: Map of New York City's Boroughs, Subway Stations and Lines



Source: Generated for the purposes of this paper, based on data from NY State.

Because of this, alternating services at a select given station in the Bronx, Brooklyn and Queens can take passengers to different parts of Manhattan without transferring between lines (MTA Maps, 2023). As one can imagine, this also allows for easier maintenance schedules as services can be rerouted when a part of a line is closed so that the whole line/branch does not have to close. Finally this highlights the complexity of the New York City Subway and its service patterns that are essential to understanding the data in the next section.

V. Data

The data is collected from 3 different governmental levels within the United States. The first of which is from the federal government via the United States Census. Using the yearly American Community Survey (ACS) panel data, the US Census estimates via a representative sample descriptive statistics of economic data related to income, gender, age, race, education, employment, commuting, and households and more. Due to the nature of the pandemic greatly affecting domestic migration in and out of New York City, each variable is taken from aggregating the most recent 2019, 2020, and 2021 surveys to minimize the concern for pandemic related outliers. From this, the summary statistics of the relevant subset of variables is shown in the table below for every census tract within New York City:

Table 1: Neighborhood Characteristics of New York City

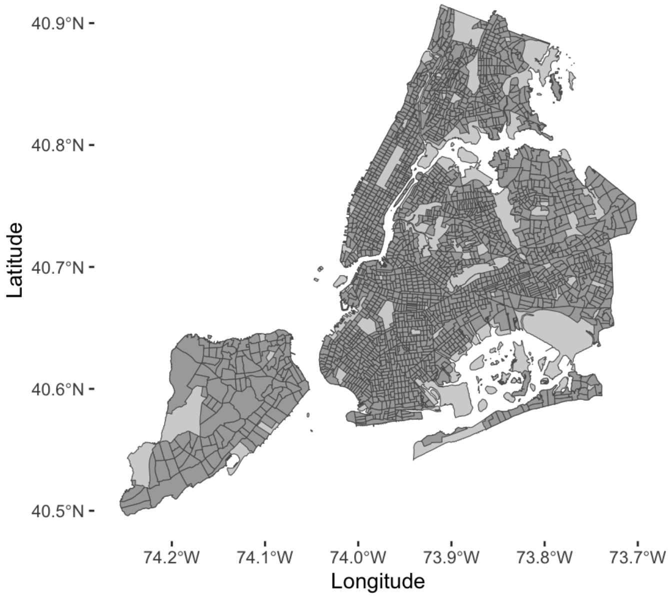
Variable	Min.	25%	50%	75%	Max.	Mean.
Population	64	2426	3547	4863	16629	3860
Population Density	217.8	26748.1	44417.9	70669.7	311040.2	53310.7
Adult Population	52	1955	2852	3934	15259	3140
Workforce	52	1212	1772	2500	10426	1996
% in Workforce	0.093	0.579	0.633	0.687	1.000	0.633
% Employed	0.484	0.910	0.940	0.959	1.000	0.930
% Unemployed	0.000	0.037	0.059	0.089	0.435	0.069
Commuting Population	38	1089	1603	2256	9822	1807
% Drive	0.000	0.143	0.268	0.449	0.888	0.303
% Public Transport	0.000	0.397	0.530	0.637	0.925	0.513
% Walk	0.000	0.030	0.063	0.108	0.721	0.088
% Work from Home	0.000	0.024	0.052	0.094	0.632	0.068
Mean Commute Time	13.9	38.6	43.2	47.2	67.9	42.4
Households	32.0	818.8	1269.0	1837.2	8177.0	1452.9
Median Household Income	9740	50277	69642	91081	244702	74396
Mean Household Income	17488	69516	90444	114872	550058	101042
Per Capita Income	7173	23940	31818	43433	320536	40247

Number of Observations: 6344

Census tract level data was chosen, as it is the smallest administrative unit of economic data that can be applied to New York City as whole as the city contains thousands of census tracts. Therefore, this census tract level data most accurately shows the great neighborhood discrepancies by variable. This is highlighted by such examples as the average per capita income by census tract having an ~45 fold increase between the minimum average and the maximum average. Likewise, some census tracts in New York City see near universal walking or taking of public transportation to work whereas some census tracts see a plurality using cars to get to work or just work from home in general. Hereafter, while parts of New York City, such as Manhattan, contain some of the densest populations on Earth, other areas of New York City are as sparsely populated as other US states on the whole. These discrepancies highlight the need to assess neighborhood differences to run a powerful regression on public transit ridership.

The second data source is the New York City provided census tract geospatial vector data, which maps all the census tracts from the 2020 Census within New York City, as seen in the figure below:

Figure 2: Census Tracts of New York City



Note: The missing area between tracts represent uninhabited waterways, parks, graveyards, airports, etc.

Source: Generated for the purposes of this paper, based on data from NY State.

The data provides information on the shape, borough, census tract name, and a unique identification code. This allows the vector data to be joined by the identification code to the ACS panel data by matching both ID codes to allow the ACS economic data to become spatial data.

The third and final source of data is from New York State, of which the MTA data is provided by the state's database. Using new MTA data published for the first time in May 2023, hourly individual station data is the main dataset used to run regression analyses against the socio-economic variables provided by the census. However, this paper also seeks to combine existing unlinked MTA datasets that provide other metrics, such as wait times, train frequency, and train performance metrics in conjunction with the ridership data, while also creating new data from published time tables. First of the datasets MTA is the *MTA Subway Wait Assessment*, which records "The total number of scheduled timepoints where trips pass the wait assessment standard, per month and subway line" (Metropolitan Transportation Authority, 2023). This wait assessment keeps track of the number of trains deployed each month relative to the number of scheduled trains per month per each Subway line in New York City. However, the dataset does not provide information on how the wait assessment is calculated for each line for each month. Nevertheless, the relevant descriptive statistics are displayed below:

Table 2.a: Subway Wait Assessment

Variable	Min.	25%.	50%	75%.	Max.	Mean.
Passed Timepoints	3	1976	4500	10512	556484	12603
Scheduled Timepoints	28	2692	6278	15238	763235	17776

Number of Observations: 940

Each variable in the table above is also assigned by line and month, and also broken down into 4 periods, weekday-peak, weekday-off peak, weekend-peak and weekend-off peak, the sum of which gives the total timepoints for a given line each month. Additionally, in this dataset, peak hours are defined as 7-10 am and 4-7 pm on weekdays and 10 am to 6pm on weekends, with all other hours being off peak, and the weekend being defined as Saturday and Sunday only. Furthermore, this dataset was first released in February 2023 but is updated monthly back from January 2020 to present.

The next dataset used in this paper is the *MTA Subway Service Delivered* dataset. Also first released in February 2023, but beginning the data in January 2020, this dataset records the number of scheduled and actual trains by line per month. Unlike the *Wait Assessment* dataset however, each month's number of trains scheduled and provided are only subdivided into weekday and weekend service, and not peak versus off peak service, the summary statistics of which are shown in Table 2.b below:

Table 2.b: Subway Service Delivered

Variable	Min.	25%.	50%	75%.	Max.	Mean.
Scheduled Trains	95	990	1536	2092	50039	2943
Actual Trains	72.0	952.5	1464.0	1978.5	47380.0	2768.9

Number of Observations: 479

As can be seen from the table above, both the numbered of scheduled trains and the number of actual trains per weekdays/weekend per month have a significantly diverging mean from a lower median, suggesting that most lines see significantly less trains, while some lines see frequent service, this will be relevant to the results section later.

The next dataset to be used, is the *MTA Subway Customer Journey-Focused Metrics* dataset which estimates the service delays due to train frequency (platform time) and train speed (train time) by line, peak vs off peak, month, and year as shown in Table 2.c:

Table 2.c: Subway Service Delays (Minutes)

Variable	Min.	25%.	50%	75%.	Max.	Mean.
Platform Time	0.1643	0.9788	1.2542	1.5756	3.1755	1.2696
Train Time	-1.1029	0.0992	0.4260	0.6209	1.3770	0.3480

Number of Observations: 479

The table shows that a combined average of ~1.62 minutes of delays is on each customer journey throughout the Subway.

Finally, the last MTA provided dataset is the one that provides the main dependent variable: Hourly Ridership. Provided by the *MTA Subway Hourly Ridership* dataset, the hourly ridership numbers per station (both total ridership and the subset of those who are transferring) are accompanied by the station's name, unique identification code (as several stations share the same name), the Subway line routes that operate at each station, the borough each station is in, and the latitude and longitude of each station. The numeric variables of which are shown in the descriptive statistics below:

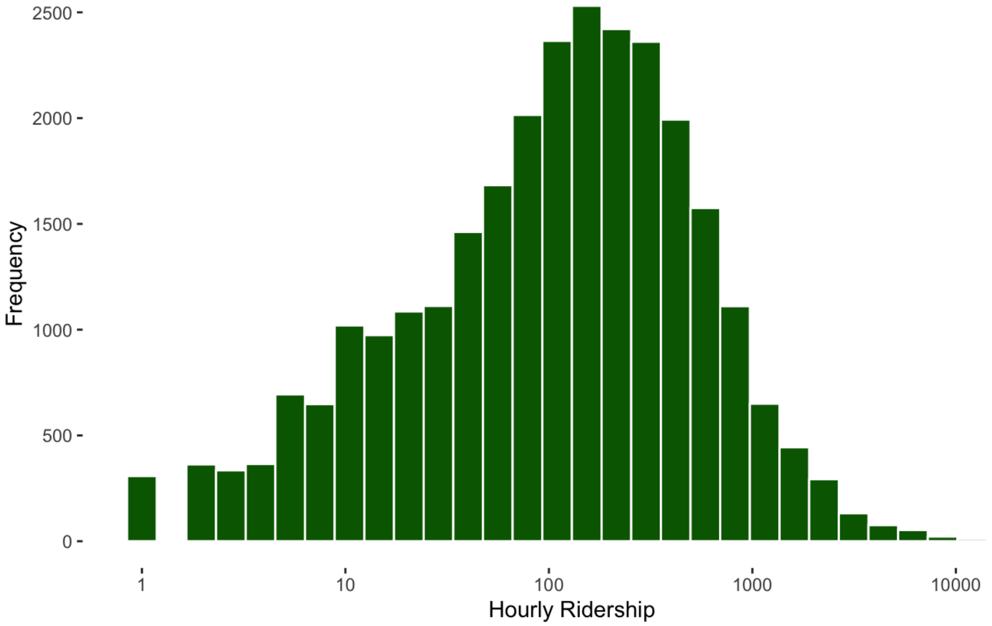
Table 2.d: Subway Hourly Ridership

Variable	Min.	25%.	50%	75%.	Max.	Mean.
Transfers	0.0	0.0	2.0	10.0	2374.0	12.7
Ridership	1.0	33.0	122.0	323.0	24845.0	299.6

Number of Observations: 2810659

As shown, the dataset for hourly ridership is quite large by accounting for each station, each hour of each day of the year. Additionally, as is expected, the mean is significantly higher than the median for both *Ridership* and *Transfers* as the median station is a local, non transfer station in an outer borough, meaning the the heavily used Manhattan station complexes with multiple skew the mean upwards due to the sheer volume of commuters congregating in Manhattan's Central Business District for work. This is shown in the histogram of the 1% sample of ridership which is log transformed to be shown normally distributed in Figure 3 as follows:

Figure 3: NYC Subway Hourly Ridership



Source: Generated for the purposes of this paper, based on data from NY State.

In addition to the *Ridership* data above, as a robustness check, new data is created using the time it takes for a subway service to make one round trip using MTA produced subway time tables per line (MTA Schedules, 2023). This involved the manual scraping of train departure times at one terminal to their arrival times at the other terminal per each of the four stated periods and then their times are summed for each station for all lines connecting at said station, providing the total route time. The summary statistics of which are created in Table 2.e:

Table 2.e: Subway Trip Time (Minutes)

Variable	Min.	25%.	50%	75%.	Max.	Mean.
Route Time	0.0	118.0	172.0	301.0	2150.5	229.5

Number of Observations: 2810659

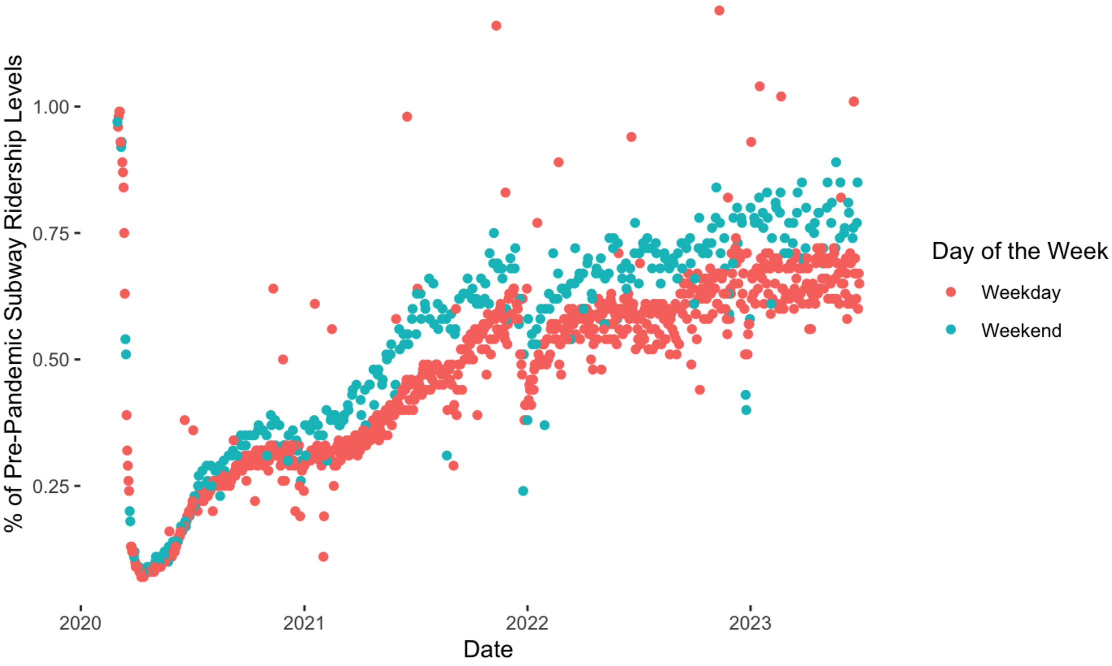
Here the minimum represents the excluded shuttle stations as they are outliers, and maximum represents the station with the most transfers: Times Square-42nd Street/Port Authority Bus Terminal. Due to the experimental and imprecise nature of Table 3's data, it will be discussed how it will be used in the next section as exclusively as part of a robustness check.

Finally, it is important to note that due to the large number of observations in Table 2.d & 2.e, Table 2 only include data from the months February-June 2022 and February-June 2023, discarding the rest of 2022's *Ridership* data for both its size in Table 2.d and relevancy. Furthermore, the empirical analysis and results incorporate only using a 1% sample of the *Ridership* dataset, but the full extent of all other datasets for spatial analysis. Additionally removing the *Wait Assessment*, *Service Delivered*, and *Service Delays* data prior to 2022 will be discussed further in the Empirical Strategy section next.

VI. Empirical Strategy

Given that immediately prior to the pandemic the past decade in the United States saw falling public transit ridership, even as systems expanded (Congressional Research Service, 2019), the pandemic offers new opportunities to explore the above relationship as ridership recovers in the hopes that further research will assess how to best effectively develop and utilize public transit infrastructure that will be vital to a greener and more sustainable future for the globe and the United States economy in particular. One time period where public transit ridership recovery in New York City has been strongest is on the weekend as shown in Figure 4:

Figure 4: Subway Ridership Return by Day of the Week



Source: Generated for the purposes of this paper, based on data from NY State.

VI.I. Ordinary Least Squares

Therefore, the primary purpose of this paper is to assess the relationship between ridership and train frequency by accounting for the 4 travel periods: weekday-peak, weekday-off peak, weekend-peak, and weekend-off peak. Henceforth, before accounting for periods, to start with the most basic model, the independent variable of interest is train frequency, and the dependent variable of interest is ridership. This is represented most fundamentally by the Ordinary Least Squares Equation below:

$$(1.1) \quad R_{i,t} = \beta_0 + \theta tph_{i,t} + \varepsilon_{i,t}$$

In which, $R_{i,t}$ is the natural logarithm of ridership at a given subway station *i,* which is on a given day at a specific hour *t*. Thereby, $tph_{i,t}$ is the natural logarithm of trains per hour at the same hour *t*, and the same subway station *i*, with θ being the coefficient of the regressor and β_0 and being the intercept and $\varepsilon_{i,t}$ being the error term. The natural logarithm is taken of both *R* and *tph* to approximate a normal distribution better.

However, as mentioned in the previous section the data is not provided in the *trains per hour* form. Therefore, using the *Wait Assessment* data to estimate the average trains per the four periods (weekday-peak, weekday-off peak, weekend-peak, and weekend-off peak), the period can be divided by the amount of hours contained in the period each month, $tph_{i,t}$ from (1.1) above can therefore also be represented as follows below:

$$(1.2) \quad tph_{i,t} = \sum_{l=1}^{L} \frac{Wp,m}{Wm \times Hp,m} tpm_{l,m}$$

Where $tph_{i,t}$ is the same variable as before, but $w_{p,m}$ is the wait assessment for passed timepoints of a given subway line in a given period p in a time m (months from 2/22 to 6/22 & 2/23 to 6/23) and w_m is the total wait assessment for passed timepoints in a given line over the whole month m. $H_{p,m}$ is the number of hours in a given period p for a given month m, as this varies per month based on the number of days in the month and the number of weekdays and weekends in the month. Finally, $tpm_{l,m}$ is the number of trains per month m, per subway line l, which is as provided in the raw data form. This is then all summed and grouped by the number and name of the lines L at subway station i to account for the total trains per hour at transfer stations. This provides an alternative way to write Equation 1.1 as follows:

$$(1.3) \quad R_{i,t} = \beta_0 + \theta \sum_{l=1}^{L} \frac{Wp,m}{Wm \times Hp,m} tpm_{l,m} + \varepsilon_{i,t}$$

However, one of the concerns of this basic linear model is that there are other potential exogenous factors that may better account for the variation in ridership than just trains per hour alone. Therefore, to start the stated five month period is selected from both 2022 and 2023 to compare only shared months so far in 2023 and to remove seasonal differences from the dataset that may change ridership in the fall and winter of 2022.

Likewise, starting with the right side of the equation above, the variation in ridership may be attributed to variables such as *relatively* time independent neighborhood differences between stations, such as whether the area around the station is more affluent, densely populated, has lower unemployment, etc. To account for this, a Ordinary Least Squares Model containing the relevant census variables is constructed to better account for a variation in ridership. This is provided as follows:

$$(2.1)\quad R_{i,t} = \beta_0 + \theta\sum_{l=1}^{L} \frac{Wp,m}{Wm \times Hp,m} tpm_{l,m} + \beta_1 income_{i,\ ct} + \beta_2 density_{i,\ ct} + \beta_3 lines_{i,\ ct} + \beta_4 employment_{i,\ ct}$$
$$+ \beta_5 public\_transit_{i,\ ct} + \beta_6 commute_{i,\ ct} + \varepsilon_{i,t}$$

Where, *income* is the natural logarithm of the mean income for the census tract (*ct*) containing the station *i*, and *density* is the mean population density (square miles), *lines* is the number of subway lines, *employment* is the mean employment rate (normalized from 0 to 1), *public_transit* is the mean of the percent of people who commute by public transit (also normalized from 0 to 1), and *commute* is the mean commute time for people living in the census tract, for station *i* in census tract *ct*, all averaged between 2019, 2020, and 2021 ACS respectively.

Additionally, adding in the time dependent variables to the right side of the equation, allows for the control of the difference in ridership levels based on the type and time of day, as ridership is not constant throughout a given day. Equation (2.2) adds them in below:

$$(2.2)\quad R_{i,t} = \beta_0 + \theta\sum_{l=1}^{L} \frac{Wp,m}{Wm \times Hp,m} tpm_{l,m} + \beta_1 income_{i,\ ct} + \beta_2 density_{i,\ ct} + \beta_3 lines_{i,\ ct} + \beta_4 employment_{i,\ ct}$$
$$+ \beta_5 public\_transit_{i,\ ct} + \beta_6 commute_{i,\ ct} + \beta_7 we_t + \beta_8 op_t + \beta_9 month + \beta_{10} year + \varepsilon_{i,t}$$

Here, equation (2.2) estimates the effect of trains per hour on ridership by accounting for whether it is a weekday = 0 or weekend = 1 in the binary variable *we*, and whether it is peak = 0 or off peak = 1 in the variable *op* at a given time *t* (hours) both respectively. The numeric variable *month* accounts for whether the data is February = 2 all the way to June = 6, and the variable *year* is whether it is 2022 or 2023 respectively.

As will be discussed in the following section, the Ordinary Least Squares equation (2.2) is significantly better at accounting for variation between ridership and trains per hour than equation (1.3) alone. However, there are multiple concerns that may make the coefficients on the regressors biased. First, not all subway riders that enter a station live or work in the neighborhood that station is in, as many riders transfer from other modes to that station via either an out-of-system transfer or most commonly: the bus. Therefore, subtracting out those who transfer to the station better accounts for the local effect of ridership on the neighborhood characteristics. This results in a better OLS estimation equation (2.3) below:

$$(2.3) \quad e^{Ni,t} = e^{Ri,t} - e^{Ti,t}$$

Where $T_{i,t}$ is the natural logarithm of the number of riders who transfer to station *i* at a given time *t*, and $N_{i,t}$ is the natural logarithm of the number of riders who do not transfer then taking at station *i* at a given time *t*. Because the time independent neighborhood characteristics better estimate $N_{i,t}$ instead of $R_{i,t}$, but the time dependent variables also may affect $R_{i,t}$ more, adding $T_{i,t}$ to both sides of the equation allows for accounting of transfers on ridership while still controlling for both the time dependent and independent variables. Rearranging the equation, the final Ordinary Least Squares Equation is as follows:

$$(2.4) \quad R_{i,t} = \beta_0 + \theta \sum_{l=1}^{L} \frac{Wp,m}{Wm \times Hp,m} tpm_{l,m} + \gamma T_{i,t} + \beta_1 income_{i,\,ct} + \beta_2 density_{i,\,ct} + \beta_3 lines_{i,\,ct} + \beta_4 employment_{i,\,ct} + \beta_5 public\_transit_{i,\,ct} + \beta_6 commute_{i,\,ct} + \beta_7 we_t + \beta_8 op_t + \beta_9 month + \beta_{10} year + \varepsilon_{i,t}$$

Here γ is the new coefficient of estimator transfers ($T_{i,t}$) on ridership ($R_{i,t}$). Although this maximizes the explanatory nature of the relevant variables by minimizing the error term, there are some other concerns with using just this Ordinary Least Squares approach.

VI.II. Fixed Effects

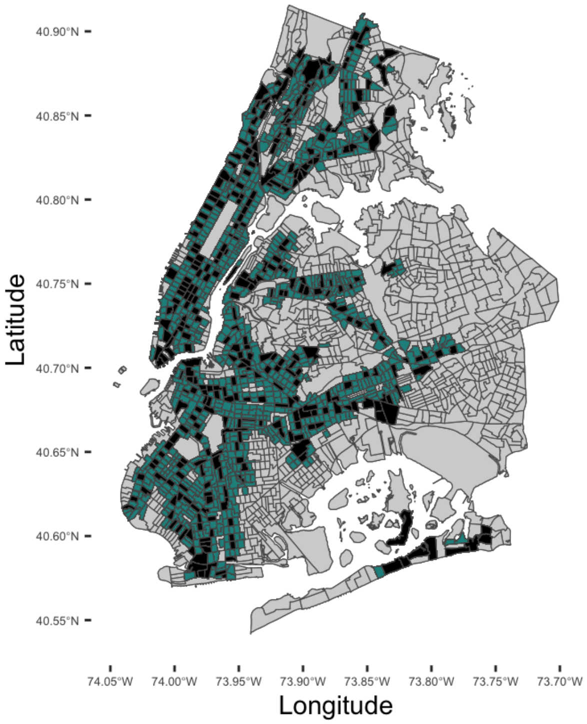
One of main concerns is that the statistical model will model for all of the variation between different time independent neighborhood variables of $income_{i,\,ct}$ through $commute_{i,\,ct}$ from equation (2.4) above via each census tract. To address this concern a Fixed Effect (FE) model will allow for the measurement of within group variation in ridership per census tract. This leads to equation (3) as shown:

$$(3) \quad R_{i,t} = \beta_0 + \theta \sum_{l=1}^{L} \frac{Wp,m}{Wm \times Hp,m} tpm_{l,m} + \gamma T_{i,t} + \beta_1 income_{i,\,ct} + \beta_2 density_{i,\,ct} + \beta_4 employment_{i,\,ct} + \beta_5 public\_transit_{i,\,ct} + \beta_6 commute_{i,\,ct} + \beta_{11} pwe_m + \beta_{12} pop_m + \delta_{lines} + \delta_{time} + \varepsilon_{i,t}$$

Where, as will be discussed in the next section, for better measurement of ridership's relative gains by the four different time periods, pwe_m replaces we_t as the percent of monthly ridership that is on the weekend for a given month m normalized from 0 to 1, and pop_m also replaces op_t as the percent of monthly ridership that is during an off peak period for a given month m normalized from 0 to 1. Likewise, δ_{lines} is the time independent number of subway lines fixed effect for each subway station. This means that all subway stations with 2 subway lines are grouped as the same, and all subway stations with 4 subway lines are grouped as the same. This is done over specific line names because the number of lines is more relevant than line names because major transfer stations will undoubtedly have higher ridership variation than local ones. Moreover, δ_{time} represents the time fixed effect as measuring the month as a numeric variable provides less insightful results than accounting for the existing differences in ridership due to the month of the year, δ_{time} also accounts whether the year is 2022 or 2023, and the four periods within each month. This allows for a better measurement of within group differences of a given census tract in regard to ridership and trains per hour over time by holding types of stations and travel times as constant.

Unfortunately, there are still some concerns of exogenous sourced error. One of the problems of using census tract level data for modeling subway stations is that census tracts are so tiny they can be easily crossed in less than 5 minutes by foot. Therefore, much of the ridership of a station within a given census tract may be from adjacent census tracts as well. Consequently, a spatial buffer is needed to make sure all census tracts within a reasonable walking distance of a station are included. Choosing a 1 kilometer radius buffer distance around each station using the assumption of a right-angle exclusive street grid and assumptions from the Pythagorean theorem, a max diagonal walking distance would be ~1.4 kilometers or roughly in line to a rough capture area of 90% of all walking trips to a public transit station (FHWA, 2013). While this is an imperfect method of estimation of the spatial catchment area of a subway station, the highly orderly NYC street grid and conservative choice of only 1 kilometer radius allows for the exclusion of census tracts that may see people walking to a neighboring census tract's station but whom's residents but have little effect on the total ridership of the station. Moreover, if the radius was too far then assumptions that people would make use of free bus transfers would start to introduce error into the exclusion of transfers from the dependent variable data. This constructed buffer can be seen in Figure 5:

Figure 5: Subway Stations and Neighboring Census Tracts



Note: Black represents census tracts containing stations, Blue represents census tracts within 1 km of a station

Source: Generated for the purposes of this paper, based on data from NY State.

From the figure above the constructed buffer can be used on equation (3) to calculate new coefficients as follows:

$$(4) \qquad R_{i,t} = \beta_0^{*} + \theta\sum_{l=1}^{L}\frac{Wp,m}{Wm \times Hp,m}tpm_{l,m} + \gamma T_{i,t} + \beta_1^{*}income_{i,\,ct} + \beta_2^{*}density_{i,\,ct} + \beta_4^{*}employment_{i,\,ct} + \beta_5^{*}public\_transit_{i,\,ct} + \beta_6^{*}commute_{i,\,ct} + \beta_{11}pwe_m + \beta_{12}pop_m + \delta_{lines} + \delta_{time} + \varepsilon_{i,t}$$

Where β_1^{*}, β_2^{*} , β_4^{*}, β_5^{*} and β_6^{*} are the new coefficient estimations of all time independent variables averaged over all census tracts *ct* contained solely within a 1 kilometer radius of a station *i*. This allows for better accounting for larger than census tract neighborhood effects on the regression as clusters of census tracts make up the ridership of a given station.

VI.III. Robustness Checks

In order to test the validity of the OLS and FE results in the following section, 2 OLS and 2 FE robustness check regression equations are constructed. First is the OLS equation (5), which is the same as equation (2.4) but with the same buffer of 1 kilometer from equation (4) to better estimate the linear combination of neighborhood effects on ridership. Equation (6) is the same as equation (5) but with the hourly late night ridership data excluded from the 1% sample. The MTA defines late night ridership as 12 am to 5 am and the purpose of its exclusion is to account for common partial closures and service disruptions/changes due to scheduled maintenance that happens during those hours (MTA Maps, 2023).

Additionally, a similar robustness check is applied to FE equation (4) in the exclusion of late night hours in regression equation (7) for the same closure and service change reason. Lastly equation 8 serves as a robustness check using the new data created in Table 2.e. Instead of using *tph* to estimate ridership, a newly constructed variable called *headway* is used, wherein:

$$(8.1) \quad R_{i,t} = \beta_0^* + \Delta headway_{i,t} + \gamma T_{i,t} + \beta_1^* income_{i,\,ct} + \beta_2^* density_{i,\,ct} + \beta_4^* employment_{i,\,ct} + \beta_5^* public\_transit_{i,\,ct} + \beta_6^* commute_{i,\,ct} + \beta_{11} pwe_m + \beta_{12} pop_m + \delta_{lines} + \delta_{time} + \varepsilon_{i,t}$$

Equation (8.1) serves as a robustness check on *tph*, where *headway* is the time between two trains on the same line at station *i* at the same time *t* as *tph* before. Moreover, *headway* can be represented as follows:

$$(8.2)\ headway = \sum_{l=1}^{L} (\frac{rt}{tph} + \ platform\_time \ + \ train\_time)$$

Wherein *rt* is the time it takes for a subway service to complete one full round trip to an end terminal, *tph* is the trains per hour variable from before, *platform_time* is average additional wait time on a platform, and *train_time* is the average additional wait time on a train. This hopes to estimate the average scheduled plus delayed wait times between two trains at each station averaged over the number of lines *l* at a given station *i.* However, due to the nature of this experimental newly created data the next section will discuss how it remains only as a robustness check on the FE models.

VII. Results

VII.I. OLS Results

The first regression output results that will be discussed are the simple Ordinary Least Squares Model of equation (1.1) and the complex OLS model of equation (2.4) and their robustness checks of equations (5) & (6). The coefficients are shown on the top row for each variable with the heteroskedasticity-robust standard errors in parentheses and the p-values as asterisks. They are shown in Panel 1 below:

Panel 1: Ordinary Least Squares Results

Variable	(1.1)	(2.4)	(5)	(6)
Log(Trains Per Hour)	0.6394***	-0.0224*	-0.0205*	-0.0084
	(0.0039)	(0.0091)	(0.0090)	(0.0077)
Log(Transfers)		0.6421***	0.6407***	0.4291***
		(0.0046)	(0.0046)	(0.0041)
Weekend		-0.2578***	-0.2627***	-0.4127***
		(0.0150)	(0.0150)	(0.0131)
Offpeak		-0.8519***	-0.8564***	-0.5030***
		(0.0161)	(0.0160)	(0.0143)
Month		0.0304***	0.0230***	0.0293***
		(0.0050)	(0.0050)	(0.0040)
Year		0.0693***	0.0698***	0.0839***
		(0.0135)	(0.0135)	(0.0109)
Log(Per Capita Income)		0.1318***	0.0571**	0.1278***
		(0.0139)	(0.0209)	(0.0167)
Log(Population Density)		0.1591***	0.3619***	0.4623***
		(0.0122)	(0.0209)	(0.0175)
Commute via		1.2032***	1.2786***	1.5032***
Public Transit (%)		(0.0623)	(0.0822)	(0.0707)
Employment Rate (%)		0.4183*	0.3026	1.5549***
		(0.1687)	(0.2812)	(0.2397)
Commute time		-0.0463***	-0.0525***	-0.0522***
		(0.0013)	(0.0018)	(0.0015)

Number of Subway Lines		0.1273***	0.1269***	0.1491***
		(0.0059)	(0.0057)	(0.0051)
Constant	3.4478***	-138.315***	-140.5685***	-171.8698***
	(0.0197)	(27.3102)	(27.2515)	(22.1506)
Observations	26314	26314	25996	19773
R^2	0.1284	0.5750	0.5792	0.65263

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$; Parentheses contain heteroskedasticity-robust standard error.

The simple OLS model of equation (1.1), shows a positive relationship between the natural logarithm of trains per hour and hourly ridership, wherein 1% increase in the trains per hour is associated with a ~0.64% increase in ridership. However, as expected the R^2 is relatively low at just 0.1284 so the explanatory nature of the predictor variable, trains per hour, is quite weak given the concerns of omitted variable bias. Therefore, adapting equation (1.1) to account for neighborhood differences throughout NYC in Equation 2.4 raises the R^2 to 0.5750 and decreases the coefficient θ to -0.0224 where a 1% increase in the trains per hour correlates with a ~0.02% reduction in ridership per hour, when holding all other independent variables constant. Moreover, the p-value increases to between 0.01 and 0.05 suggesting the confidence in this coefficient estimation has decreased. Nevertheless, the introduction of new variables gives predictable expectations on their effect on ridership. Transfers from the bus or commuter rail predictably positively correlate with higher ridership via the coefficient γ shown as 0.6421 in the data, as the data does not allow for more transfers than total riders. Additionally, higher incomes, population density, percent of people commuting by public transit, employment rates, and the number of subway lines in a given census tract positively correlate with higher ridership. This can be expected as a gainfully employed and relatively prosperous census tract within Manhattan and the inner outer boroughs of the Bronx, Brooklyn and Queens would most likely take the Subway frequently. Interestingly, longer commute times are associated with riding the subway. On first glance this may not seem intuitive as subway trains would appear to allow for less draining and longer commuting times than walking, hybrid work, or driving. However, this can be reasonably explained by those with longer commutes preferring not to ride the subway off-peak and on weekends for recreational reasons, ultimately dragging down ridership. Additionally, those who live further away from Manhattan, where a plurality of jobs are, tend to live in less dense and therefore lower ridership areas on local stations. This suggests most Subway riders have relatively shorter commutes living in the dense core while a minority of less dense census tracts have riders who commute longer on public transit. Ultimately however, the coefficients on the time independent neighborhood variables just mentioned appear logical and strongly predictive in equation (2.4) given their small p-values. Thereafter, in regard to the time dependent variables, the binary variables *weekend* and *off peak* are predictably negative as even in uneven pandemic ridership recovery, weekend and off peak times result in lower ridership levels compared to the combined rush hour weekday peak times. Congruently, as expected, the month of the year and the year itself are positively associated with increases in ridership. Ultimately, the results from (2.4) produce a stronger model to account for the variation in ridership, but also suffer from concerns of omitted variable bias, mainly through the form of neighborhood effects.

As mentioned in the empirical strategy, the relatively micro scale of census tracts means that multiple census tracts can produce riders at a single station. Using the buffer from the Fixed Effects Models, equation (5) shows the OLS results for accounting for the buffer in the analysis. Importantly, it suggests that trains per hour, transfers, the month of the year, the per capita income, the employment rate, and the number of subway lines have less strong of an effect on ridership, that is the absolute values of these variables in (5) are lower than the their respective counterparts for (2.4). Whereas, the variables of weekend, off peak, year, population density, percent commute by public transit, and the average commute time, have stronger effects on ridership as their absolute values are higher than in the results for equation (2.4). However, the lower confidence in the explanatory variable, trains per hour, remains lower, and the employment rate becomes statistically insignificant. Therefore this accounting for groups of census tracts most likely better predicts the coefficients of each time independent variable as the R^2 also increases slightly.

Lastly, equation (6) builds off of the higher R^2 of the results of equation (5) by excluding the late nights (Midnight to 5 am) from the subway data as service closures and rerouting are common on individual subway lines to conduct routine maintenance during this time. This results in a significantly higher R^2 in the results for (6) when compared to the robustness check of (5). Moreover, the coefficients remain the same sign when compared to (2.4) and (5), and the within variable difference remains below a whole percentage point except in regard to employment rate which becomes significantly more positively correlated with ridership, while also achieving a much lower p-value and therefore higher confidence in the coefficient. This perhaps suggests that maybe more economically deprived (higher unemployment) census tracts experience more late night ridership as people work evening shifts. Finally, the glaring problem with this OLS model despite the higher R^2 is the statistically insignificant coefficient on the main explanatory variable *tph*, highlighting the need for a different approach to estimating a causal relationship between trains per hour and ridership as will be outlined in the Fixed Effect Model section next.

VII.II. FE Results

One of the problems with Ordinary Least Squares approach of equations (1.1), (2.4) and their robustness checks of equations (5) and (6) are that they measure differences across groups. Therefore, measuring the presence of the weekend on ridership will predictably result in a negative coefficient as commuting to work remains the number one reason people take public transit. Attempting to measure the impact of growing weekend and off peak ridership, the normalization of ridership is used for the Fixed Effects Model in equations (3), (4), (7) and (8.1). Additionally, holding the period, month and year as time fixed effect and the number of subway lines as a state fixed effect allows the measurement of within group differences more accurately, and better estimates the coefficients and the confidence levels as outlined in Panel 2 below. Coefficients remain on the first line for each variable, with their corresponding p-values as asterisks and heteroskedasticity-robust standard errors in parentheses on the line directly below. Furthermore, Panel 2 also states that the state and time fixed effects were present in all four equations.

Panel 2: Fixed Effect Results

Variable	(3)	(4)	(7)	(8.1)
Log(Trains Per Hour)	0.0343**	0.0260*	0.0395***	
	(0.0112)	(0.0112)	(0.0093)	
Log(Headway)				-0.2710***
				(0.0259)
Log(Transfers)	0.6386***	0.6333***	0.4186***	0.3970***
	(0.0045)	(0.0045)	(0.0040)	(0.0051)

Weekend Ridership	2.3670***	2.3155***	2.3730***	2.7845***
(% of Total)	(0.1494)	(0.1497)	(0.1471)	(0.1854)
Offpeak Ridership	-0.7802***	-0.7807***	-1.5631***	-1.2406***
(% of Total)	(0.1577)	(0.1592)	(0.1471)	(0.2072)
Log(Per Capita Income)	0.1071***	0.1151***	0.1916***	0.2565***
	(0.0139)	(0.0164)	(0.0128)	(0.0170)
Log(Population Density)	0.1739***	0.4225***	0.5265***	0.5516***
	(0.0122)	(0.0199)	(0.0163)	(0.0206)
Commute via	0.9790***	0.7254***	0.8047***	0.5405***
Public Transit (%)	(0.0664)	(0.0671)	(0.0575)	(0.0759)
Employment Rate (%)	0.4627**	1.0735***	1.4223***	1.6093***
	(0.1735)	(0.1679)	(0.1401)	(0.1763)
Commute time	-0.0418***	-0.0336***	-0.0336***	-0.0217***
	(0.0013)	(0.0014)	(0.0012)	(0.0017)
Year, Month	Yes	Yes	Yes	Yes
& Period FE				
Subway Lines	Yes	Yes	Yes	Yes
Number FE				
Constant	0.6967*	-2.9328***	-4.3330***	-5.2228***
	(0.2866)	(0.3828)	(0.3212)	(0.4041)
Observations	26314	25996	19773	11735
R^2	0.5886	0.5904	0.6735	0.5982

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$; Parentheses contain heteroskedasticity-robust standard error.

The first equation of question is (3) as the results exclude the effects of neighboring census tracts only incorporating the local neighborhood effect of a given station's corresponding census tract. As can be seen, when accounting for the period fixed effects and line fixed effects, a stronger confidence level and positive coefficient appears on trains per hour, in contrast to the corresponding result for (2.4). Here θ is estimated as 0.0343 and therefore explains that 1% increase in the trains per hour at a given station is associated with an on average ~0.034% increase in ridership at a given station. As mentioned, this relationship is positive, suggesting that results of the OLS regressions of (2.4), (5) and (6) may have been biased downwards for not accounting for within group differences, and (1.1) was expectedly biased to far upwards as doubling the train frequency would result in an over 60% increase in ridership which would put ridership significantly above pre-pandemic levels, therefore most likely overstating the effect. The more modest ~0.034% most likely better estimates the steady growth in both ridership and number of trains per hour the MTA has deployed onto the system.. Likewise, the coefficient on T, γ, remains relatively constant compared to the OLS regressions of equations (2.4) & (5), and the time independent variables of income, population density, and employment rate fall within the range of values estimated by the equations (2.4), (5) and (6). Interestingly, commuting by public transit and the average commute time by all modes has slightly less of an effect on ridership when compared to the OLS regression results, but that can be reasonably explained by the fixed effects controlling for variations in non commuting time periods such as on weekends, off peak, and the intersection of the two.

Continuing on, the percent of ridership that is on a weekend during a given month is very strongly correlated with an increase in ridership. Therefore, as weekend ridership becomes a larger and larger portion of ridership for a given month, the total ridership increases as well. This suggests a strong shift towards growth in leisure and recreational travel post-pandemic. However, off peak ridership as a percentage of total ridership's coefficient is strongly negative. This suggests that the highest ridership return has been during peak periods on the weekend, which aligns with the summary statistics of ridership and Figure 4.

Additionally, the FE model of equation (4), which includes a 1 kilometer buffer of census tracts in its regression results in a mildly higher R^2 than the results of (3), but also better accounts for the neighborhood effects when calculating within group differences. Consequently, between the results of equations (3) and (4), the coefficients remain relatively constant (within an order of magnitude) on subway transfers, weekend and off peak ridership as individually a percent of total ridership, per capita income, the percentage of people who commute by public transport, and the mean commute time. However, the coefficients on the population density and the employment rate both see large increases, suggesting that the individual census tract that a station is in may not always accurately sample the neighboring census tracts' population density and employment rate as the their distribution may be leptokurtic distributed downwards around stations relative to other control variables. Another difference between the results of Equations (3) and (4) is, much like the OLS results, the decrease in confidence again of the coefficient on the regressor trains per hour. The results for (4) may be understating the effect (given the decrease in confidence) of trains per hour on ridership. Therefore, a robustness check is included for the fixed effect models in the results for equations (7) and (8.1).

Equation (7) results highlight a better estimation of the coefficient on trains per hour via a higher confidence level and a stronger R^2 overall. Much like equation (6) by excluding late nights from service, which often has low ridership, complicated service patterns and part closure, equation (7)'s building on (4) results' leads to the strongest estimation of all coefficients across all models. Every variable is at the highest confidence level, and accounting for within group variation allows for stronger certainly in previously weaker variables of employment rate and trains per hour. This suggests that equations (3) & (4) were underestimating off peak ridership, per capita income, population density, and the employment rate significantly, while the mean commute time, commuting by public transit, and weekend ridership were estimated relatively consistently. As for the two most important variables, trains per hour & transfers, the decrease in p-value reflects that the greater the positive value in trains per hour the greater the confidence level, showing the flaws in the OLS approach. Therefore, here it can be said with a high certainty that a 1% increase in trains per hour results in a ~0.04% increase in ridership. Additionally, the relatively strong decline in the value of the coefficient on transfers suggest that transfers from outside the subway system are most common and explanatory off peak and late nights in regard to ridership, this is to be expected as night buses often replace partial subway closures and disruptions, and are significantly faster than daytime buses given the minimal traffic. It is however important to note that results for equation (7), while more perhaps and causal in nature, are secondary and only serve as a robustness check because the exclusion of inconvenient data gives less credence to these results serving as a primary explanation. Nevertheless, they do suggest the general causal and positive and negative relationship of the independent variables in relation to ridership.

Finally, the final robustness check excludes the use of trains per hour directly as the independent variable and instead uses headway which normalizes trains per hour over the length of a given line's subway track. The results of equation (8.1) suggest the same positive and negative relationships of the other 3 fixed effect model results for all independent variables. Additionally, the new variable *headway* corresponds to the predicted negative relationship between longer headways and ridership. Here, a 1% increase in the headway time (minutes) between trains results in a ~0.27% reduction in ridership for a given station. Thereafter, it is important to note while this result does involve the use of creating new data, it is included only as a robustness check because of the imperfect nature of not being able to account for whether trains run the full length of their route or terminate early at terminals, or turn around tracks due to schedules, delays and disruptions. Nevertheless, the relatively consistent coefficients on the other regressors and the predictable negative relationship with ridership suggests the other 3 fixed effect models are strong and explanatory in their nature.

IIX. Conclusion

This paper attempts to use newly released data to show the diverging relationship between public transit ridership and traditional commuting in a post-pandemic world. New York City is just one example of many cities who have seen stronger off-peak, and particularly weekend, ridership returns and slower peak and weekday ridership return growth (Smith, 2023)(TfL, 2022). This highlights the change in modal split towards work from home, and an increasing interest in public transit focusing on all day frequent service. Many cities, including New York City, have taken steps to smooth out the kurtosis of train frequency towards frequent, but less rush-hour focused, all day train service. This includes recent announcements by the MTA to increase their weekend service on the Subway to better match ridership preferences and demand (Hicks, 2023). This is in line with what this paper has shown: Increasing train frequency induces higher ridership, which in turn requires more trains, creating a virtuous cycle over polluting and dangerous private motor vehicles. Further research on this subject should consider examining the change in bus route ridership, and perhaps most strikingly, commuter railroads across the US which significantly underperform other countries' ridership levels relative to size. Particularly in New York City's case, the commuter railroads of MNR and LIRR have higher fares and less frequent service than adjacent subway stations which may hinder urban ridership greatly despite capital expansion (MTA, 2023)(Gordon, 2023).

However, overall this paper has attempted to show using FE models that on top of increasing train service on the Subway, higher ridership is strongly associated with higher incomes and gainful employment, population density, and those who are already taking public transit to work, even if most of this ridership return it is outside of traditional of commuting hours. Therefore, further policies beyond increasing train service may want to examine ways to attract suburban and more economically depressed areas towards ridership, whether that be re-examining the peak-only oriented trains of suburban sections of commuter rail and the high fares of urban commuter rail respectively. All of this in conjunction will hopefully in the long run convince more Americans to choose public transit as a way of preferred mobility and significantly help combat transport emissions and vehicular collisions which both remain relatively high in the United States when compared to other high income countries (Chobrak, 2021)(Zipper, 2022).

IX. References

Census Bureau Data. (2019-2021). American Community Survey. *United States Census Bureau.*

Chrobak, Ura. (2021). Combustion-engine cars occupy a special place in American culture, but to reach net zero by 2050 the US needs to rethink its relationship with the automobile. *BBC.*

Congressional Research Service. (2019). Public Transit Ridership Continues to Decline. *Congressional Research Service.*

Donaldson, Dave. (2018). Railroads of the Raj: Estimating the Impact of Transportation Infrastructure. *American Economic Review.*

Dong, Zhi, Patricia L. Mokhtarian, Giovanni Circella, James R. Allison. (2014). The estimation of changes in rail ridership through an onboard survey: did free Wi-Fi make a difference to Amtrak's Capitol Corridor service? *Transportation.*

Duddridge, Natalie. (2023). MTA announces weekend service enhancements for 1 & 6 lines. *CBS New York.*

FHWA. (2013). Pedestrian Safety Guide for Transit Agencies. *United States Department of Transportation.*

Gordon, Aaron. (2023). New York May Have Actually Lost Transit Riders by Building An $11 Billion Train Station. *Vice.*

Hicks, Nolan. (2023). MTA plan to boost weekend subway service under Hochul's budget deal. *New York Post.*

Hutt, Rosamond. (2018). These are the 10 busiest metros in the world. *World Economic Forum.*

Jing, Q.-L., H.-Z. Liu, W.-Q. Yu, X. He. (2022). The Impact of Public Transportation on Carbon Emissions—From the Perspective of Energy Consumption. *Sustainability.*

Kawabata, Mizuki, Qing Shen. (2007). Commuting Inequality between Cars and Public Transit: The Case of the San Francisco Bay Area, 1990–2000. *Urban Studies.*

Lee, Johohn. (2023). Here's why public transit keeps running out of money. *CNBC.*

Levy, Alon. (2017). Why are the NYC subway's operating costs so high? *Curbed New York.*

Li, Junfang, Yao Minfeng, Qian Fu. (2016). Forecasting Method for Urban Rail Transit Ridership at Station Level Using Back Propagation Neural Network. *Hindawi.*

Liu, Luyu, Harvey J. Miller. (2020). The impacts of COVID-19 pandemic on public transit demand in the United States. *Plos One.*

Metropolitan Transportation Authority. (2023). Open Data. *State of New York.*

MTA. (2020). Subway and bus facts 2019. *Metropolitan Transportation Authority.*

MTA. (2022). About the MTA. *Metropolitan Transportation Authority.*

MTA. (2023). CityTicket for travel within NYC on Metro-North and LIRR. *Metropolitan Transportation Authority.*

MTA Maps. (2023). Maps. *Metropolitan Transportation Authority.*

MTA Schedules. (2023). Schedules. *Metropolitan Transportation Authority.*

Nelson, Arthur C., Robert Hibberd. (2023). Influence of Transit Station Proximity on Demographic Change Including Displacement and Gentrification with Implications for Transit and Land Use Planning After the COVID-19 Pandemic. *Transp Res Rec.*

NTD. (2020). Transit Profiles: 2020 Top 50 Reporters. *Federal Transit Administration U.S. Department of Transportation.*

NYC Open Data. (2023). Data. *City of New York.*

Popovich, Nadja, Denise Lu. (2019). The Most Detailed Map of Auto Emissions in America. *The New York Times.*

PANYNJ. (2023). Traffic and Volume. *Port Authority of New York and New Jersey.*

Sahadi, Jeanne. (2023). The latest on hybrid work: Who is WFH and who isn't. *CNN Business.*

Santanam, Tejas, Anthony Trasatti, Pascal Van Hentenryck, Hanyu Zhang. (2021). Public Transit for Special Events: Ridership Prediction and Train Optimization. *Georgia Institute of Technology.*

Shepherdson, David. (2020). U.S. public transit systems seek up to $36 billion in new bailout. *Reuters.*

Smith, Tracy. (2023). L.A. Metro Continues to Increase Ridership with Weekend Riders Driving Growth. *Los Angeles Metro.*

TBS Report. (2022). 10 largest metro rails in the world. *The Business Standard.*

TfL. (2022). Latest TfL figures show continued growth in ridership following lifting of working from home restrictions. *Transport for London.*

Walker, Alissa. (2023). The Off-Peak Rider Is the Future of the Subway. *Curbed.*

Walker, Jarrett. (2021). US Commuter Rail: What it Is and What It Could Be. *Humanist Transit.*

WMATA. (2023). Metro's Board Approves $4.8B Budget, Simplifies Fares and Increases Frequency of Service; Redesign of Better Bus Network. *Washington Metropolitan Area Transit Authority*

Wright, Robert. (2013). 24-hour transit: cities that try to travel all the time. *Financial Times.*

Wu, W., J. Hong. (2017). Does public transit improvement affect commuting behavior in Beijing, China? : A spatial multilevel approach. *Transportation Research Part D: Transport and the Environment.*

Xiang, W, Li Chen, Bin Wang, Qingwan Xue, Wei Hao & Xuemei Liu. (2021). Policies, population and impacts in metro ridership response to COVID-19 in Changsha. *Journal of Transportation Safety & Security.*

Young, Elise, Danielle Moran, Michelle Kaske. (2021). Working From Home for Some Threatens Mass Transit for All. *Bloomberg City Lab.*

Zipper, David. (2021). What if Working at Home Makes Us Drive More, Not Less? *Slate.*

Zipper, David. (2022). US Traffic Safety Is Getting Worse, While Other Countries Improve. *Bloomberg City Lab.*

Zipper, David. (2023). How to save America's public transit systems from a doom spiral. *Vox.*

IX. Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
#installs packages if required
if(!require("tidygraph")) install.packages("tidygraph")
if(!require("igraph")) install.packages("igraph")
if(!require("ggraph")) install.packages("ggraph")
#devtools::install_github("luukvdmeer/sfnetworks", force=TRUE)
if(!require("data.table")) install.packages("data.table")
if(!require("classInt")) install.packages("classInt")
if(!require("stplanr")) install.packages("stplanr")
if(!require("gstat")) install.packages("gstat")
if(!require("raster")) install.packages("raster")
if(!require("sf")) install.packages("sf")
if(!require("dplyr")) install.packages("dplyr")
if(!require("spdep")) install.packages("spdep")
if(!require("tmap")) install.packages("tmap")
if(!require("spatstat")) install.packages("spatstat")
if(!require("mapview")) install.packages("mapview")
if(!require("osmdata")) install.packages("osmdata")
if(!require("leaflet")) install.packages("leaflet")
if(!require("ggplot2")) install.packages("ggplot2")
if(!require("ggmap")) install.packages("ggmap")
if(!require("RColorBrewer")) install.packages("RColorBrewer")
if(!require("spData")) install.packages("spData")
if(!require("R.utils")) install.packages("R.utils")
#if(!require("GADMTools")) install.packages("GADMTools")
if(!require("terra")) install.packages("terra")
if(!require("exactextractr")) install.packages("exactextractr")
if(!require("viridis")) install.packages("viridis")
if(!require("spDataLarge")) install.packages("spDataLarge", repos =
"https://geocompr.r-universe.dev")
#if(!require("extract")) install.packages("extract")
if(!require("lubridate")) install.packages("lubridate")
if(!require("maps")) install.packages("maps")
if(!require("tidyverse")) install.packages("tidyverse")
if(!require("RCurl")) install.packages("RCurl")
if(!require("tidygeocoder")) install.packages("tidygeocoder")
if(!require("ggpubr")) install.packages("ggpubr")
if(!require("rvest")) install.packages("rvest")
#if(!require("stingr")) install.packages("stingr")
if(!require("purrr")) install.packages("purrr")
if(!require("tidycensus")) install.packages("Tidycensus")
if(!require("jsonlite")) install.packages("jsonlite")
if(!require("httr")) install.packages("httr")
if(!require("biscale")) install.packages("biscale")
if(!require("cowplot")) install.packages("cowplot")
if(!require("leaflet")) install.packages("leaflet")
if(!require("osrm")) install.packages("osrm")
if(!require("plotly")) install.packages("plotly")
if(!require("ggspatial")) install.packages("ggspatial")
if(!require("Hmisc")) install.packages("Hmisc")
if(!require("scales")) install.packages("scales")
if(!require("ggpmisc")) install.packages("ggpmisc")
if(!require("dbscan")) install.packages("dbscan")
if(!require("TSP")) install.packages("TSP")
if(!require("units")) install.packages("units")
if(!require("knitr")) install.packages("knitr")
if(!require("foreign")) install.packages("foreign")
if(!require("stringr")) install.packages("stringr")
if(!require("fixest")) install.packages("fixest")
if(!require("rgdal")) install.packages("rgdal")
#if(!require("regos")) install.packages("rgeos")
if(!require("lubridate")) install.packages("lubridate")
if(!require("zoo")) install.packages("zoo")
if(!require("plm")) install.packages("plm")
if(!require("lfe")) install.packages("lfe")

census_tracts <- read_sf("DIS/data/2020 Census Tracts -
Tabular/geo_export_7de52329-4f9c-42e1-8bc0-654930922144.shp")
dfsubway <-
read_csv("DIS/data/MTA_Subway_Hourly_Ridership__Beginning_February_2022.csv")
daily_ridership <-
read_csv("DIS/data/MTA_Daily_Ridership_Data__Beginning_2020.csv")
boroughs <- read_sf("DIS/data/Borough
Boundaries/geo_export_8177a95b-23fb-4ecb-87dc-a9e21086e0c4.shp")
#dfsubway <- read_csv("DIS/data/dfsubway.csv") #finished csv
lines <- read_sf("DIS/data/Subway
Lines/geo_export_cc3502b6-5b63-4db6-8ba5-2248eba6b8a1.shp")

df19p03 <- read_csv("DIS/data/ACS/ACSDP5Y2019.DP03-Data.csv")
df19p04 <- read_csv("DIS/data/ACS/ACSDP5Y2019.DP04-Data.csv")
df19p05 <- read_csv("DIS/data/ACS/ACSDP5Y2019.DP05-Data.csv")

df20p03 <- read_csv("DIS/data/ACS/ACSDP5Y2020.DP03-Data.csv")
df20p04 <- read_csv("DIS/data/ACS/ACSDP5Y2020.DP04-Data.csv")
df20p05 <- read_csv("DIS/data/ACS/ACSDP5Y2020.DP05-Data.csv")
```

```r
df21p03 <- read_csv("DIS/data/ACS/ACSDP5Y2021.DP03-Data.csv")
df21p04 <- read_csv("DIS/data/ACS/ACSDP5Y2021.DP04-Data.csv")
df21p05 <- read_csv("DIS/data/ACS/ACSDP5Y2021.DP05-Data.csv")

dfwait <- read_csv("DIS/data/MTA_Subway_Wait_Assessment__Beginning_2020.csv")
dfservice <-
read_csv("DIS/data/MTA_Subway_Service_Delivered__Beginning_2020.csv")
#dfdelayed <-
read_csv("DIS/data/MTA_Subway_Trains_Delayed__Beginning_2020.csv")
dfmetrics <-
read_csv("DIS/data/MTA_Subway_Customer_Journey-Focused_Metrics__Beginning_202
0.csv")

dfwait["year"] <- as.numeric(as.character(substr(dfwait$month, 0, 4)))
dfwait["month"] <- as.numeric(as.character(substr(dfwait$month, 6, 8)))
dfwait <- dfwait[dfwait$year >= 2022,]
dfwaitwd <- dfwait[dfwait$day_type == "1",]
dfwaitwe <- dfwait[dfwait$day_type == "2",]

dfservice["year"] <- as.numeric(as.character(substr(dfservice$month, 0, 4)))
dfservice["month"] <- as.numeric(as.character(substr(dfservice$month, 6, 8)))
dfservice <- dfservice[dfservice$year >= 2022,]
dfservicewd <- dfservice[dfservice$day_type == "1",]
dfservicewe <- dfservice[dfservice$day_type == "2",]

dfmetrics["year"] <- as.numeric(as.character(substr(dfmetrics$month, 0, 4)))
dfmetrics["month"] <- as.numeric(as.character(substr(dfmetrics$month, 6, 8)))
dfmetrics <- dfmetrics[dfmetrics$year >= 2022,]

dfwaitwd["id"] <- paste(as.character(as.numeric(dfwaitwd$year)),
                        as.character(as.numeric(dfwaitwd$month)),
                        dfwaitwd$line,
                        sep = "")
dfservicewd["id"] <- paste(as.character(as.numeric(dfservicewd$year)),
                        as.character(as.numeric(dfservicewd$month)),
                        dfservicewd$line,
                        sep = "")
dftrainswd <- merge(dfwaitwd, dfservicewd, by = "id")

dfwaitwe["id"] <- paste(as.character(as.numeric(dfwaitwe$year)),
                        as.character(as.numeric(dfwaitwe$month)),
                        dfwaitwe$line,
                        sep = "")
```

```r
dfservicewe["id"] <- paste(as.character(as.numeric(dfservicewe$year)),
                        as.character(as.numeric(dfservicewe$month)),
                        dfservicewe$line,
                        sep = "")
dftrainswe <- merge(dfwaitwe, dfservicewe, by = "id")

dftrains <- rbind(dftrainswd, dftrainswe)
rm(dfwaitwd, dfwaitwe, dfservicewd, dfservicewe, dftrainswd, dftrainswe)
rm(dfwait, dfservice)

dftrains$year.y <- NULL
dftrains$month.x <- NULL
dftrains$day_type.y <- NULL
dftrains$line.y <- NULL
dftrains$division.y <- NULL
dftrains$id <- NULL

dftrains <- dftrains %>% rename_at('division.x', ~'division')
dftrains <- dftrains %>% rename_at('line.x', ~'line')
dftrains <- dftrains %>% rename_at('day_type.x', ~'weekday')
dftrains <- dftrains %>% rename_at('period', ~'peak')
dftrains <- dftrains %>% rename_at('year.x', ~'year')
dftrains <- dftrains %>% rename_at('month.y', ~'month')

dftrains$weekday <- str_replace(dftrains$weekday, "2", "0")
dftrains$peak <- str_replace(dftrains$peak, "offpeak", "0")
dftrains$peak <- str_replace(dftrains$peak, "peak", "1")
dftrains["period"] <- paste(dftrains$weekday, dftrains$peak, sep = "")
dftrains$weekday <- as.numeric(as.character(dftrains$weekday))
dftrains$peak <- as.numeric(as.character(dftrains$peak))
dftrains$period <- as.numeric(as.character(dftrains$period))
dftrains <- dftrains %>% drop_na()

dftrains["id"] <- paste(as.character(as.numeric(dftrains$year)),
                        as.character(as.numeric(dftrains$month)),
                        as.character(as.numeric(dftrains$period)),
                        dftrains$line,
                        sep = "")
dftrains["id2"] <- paste(as.character(as.numeric(dftrains$year)),
                        as.character(as.numeric(dftrains$month)),
                        as.character(as.numeric(dftrains$weekday)),
                        dftrains$line,
                        sep = "")
```

```r
dftrains2 <- aggregate(dftrains$num_timepoints_passing_wait_assessment,
by=list(id2=dftrains$id2), FUN=sum)
dftrains2 <- dftrains2 %>% rename_at('x', ~'weight')
dftrains <- merge(dftrains, dftrains2, by="id2")
rm(dftrains2)
dftrains["percent_weight"] <- dftrains$num_timepoints_passing_wait_assessment
/ dftrains$weight
dftrains["trains_per_month"] <- dftrains$num_actual_trains *
dftrains$percent_weight

dfmetrics["peak"] <- dfmetrics["period"]
dfmetrics["peak"] <- str_replace(dfmetrics$peak, "offpeak", "0")
dfmetrics["peak"] <- str_replace(dfmetrics$peak, "peak", "1")

dfmetrics["id3"] <- paste(as.character(as.numeric(dfmetrics$year)),
                        as.character(as.numeric(dfmetrics$month)),
                        as.character(as.numeric(dfmetrics$peak)),
                        dfmetrics$line,
                        sep = "")

dftrains["id3"] <- paste(as.character(as.numeric(dftrains$year)),
                        as.character(as.numeric(dftrains$month)),
                        as.character(as.numeric(dftrains$peak)),
                        dftrains$line,
                        sep = "")
dftrains <- merge(dftrains, dfmetrics, by = "id3")
rm(dfmetrics)

dftrains$year.y <- NULL
dftrains$month.y <- NULL
dftrains$peak.y <- NULL
dftrains$line.y <- NULL
dftrains$division.y <- NULL
dftrains$period.y <- NULL
dftrains$num_passengers <- NULL
dftrains$over_five_mins <- NULL
dftrains$over_five_mins_perc <- NULL
dftrains$`customer journey time performance` <- NULL
dftrains$total_apt <- NULL
dftrains$total_att <- NULL
dftrains$id2 <- NULL
dftrains$id3 <- NULL
```

```r
dftrains <- dftrains %>% rename_at('division.x', ~'division')
dftrains <- dftrains %>% rename_at('line.x', ~'line')
dftrains <- dftrains %>% rename_at('peak.x', ~'peak')
dftrains <- dftrains %>% rename_at('year.x', ~'year')
dftrains <- dftrains %>% rename_at('month.x', ~'month')
dftrains <- dftrains %>% rename_at('period.x', ~'period')

df

dftrains["additional_time"] <- dftrains$`additional platform time` +
dftrains$`additional train time`
dftrains$`additional platform time` <- NULL
dftrains$`additional train time` <- NULL

dfsubway$payment_method <- NULL

dfsubway["year"] <- substr(dfsubway$transit_timestamp, 7, 10)

dfsubway["month"] <- substr(dfsubway$transit_timestamp, 0, 2)

dfsubway["day"] <- substr(dfsubway$transit_timestamp, 4, 5)

dfsubway["hour"] <- substr(dfsubway$transit_timestamp, 12, 22)
dfsubway$hour <- str_replace(dfsubway$hour, "12:00:00 AM", "00")
dfsubway$hour <- str_replace(dfsubway$hour, "01:00:00 AM", "01")
dfsubway$hour <- str_replace(dfsubway$hour, "02:00:00 AM", "02")
dfsubway$hour <- str_replace(dfsubway$hour, "03:00:00 AM", "03")
dfsubway$hour <- str_replace(dfsubway$hour, "04:00:00 AM", "04")
dfsubway$hour <- str_replace(dfsubway$hour, "05:00:00 AM", "05")
dfsubway$hour <- str_replace(dfsubway$hour, "06:00:00 AM", "06")
dfsubway$hour <- str_replace(dfsubway$hour, "07:00:00 AM", "07")
dfsubway$hour <- str_replace(dfsubway$hour, "08:00:00 AM", "08")
dfsubway$hour <- str_replace(dfsubway$hour, "09:00:00 AM", "09")
dfsubway$hour <- str_replace(dfsubway$hour, "10:00:00 AM", "10")
dfsubway$hour <- str_replace(dfsubway$hour, "11:00:00 AM", "11")
dfsubway$hour <- str_replace(dfsubway$hour, "12:00:00 PM", "12")
dfsubway$hour <- str_replace(dfsubway$hour, "01:00:00 PM", "13")
dfsubway$hour <- str_replace(dfsubway$hour, "02:00:00 PM", "14")
dfsubway$hour <- str_replace(dfsubway$hour, "03:00:00 PM", "15")
dfsubway$hour <- str_replace(dfsubway$hour, "04:00:00 PM", "16")
dfsubway$hour <- str_replace(dfsubway$hour, "05:00:00 PM", "17")
dfsubway$hour <- str_replace(dfsubway$hour, "06:00:00 PM", "18")
dfsubway$hour <- str_replace(dfsubway$hour, "07:00:00 PM", "19")
```

```
dfsubway$hour <- str_replace(dfsubway$hour, "08:00:00 PM", "20")
dfsubway$hour <- str_replace(dfsubway$hour, "09:00:00 PM", "21")
dfsubway$hour <- str_replace(dfsubway$hour, "10:00:00 PM", "22")
dfsubway$hour <- str_replace(dfsubway$hour, "11:00:00 PM", "23")

dfsubway["time"] <- paste(dfsubway$year, dfsubway$month, dfsubway$day,
dfsubway$hour, sep = "")

dfsubway$year <- as.numeric(as.character(dfsubway$year))
dfsubway$month <- as.numeric(as.character(dfsubway$month))
dfsubway$day <- as.numeric(as.character(dfsubway$day))
dfsubway$hour <- as.numeric(as.character(dfsubway$hour))
dfsubway$time <- as.numeric(as.character(dfsubway$time))

dfsubway <- dfsubway[dfsubway$month >= 2,]
dfsubway <- dfsubway[dfsubway$month <= 6,]

dfsubway["transfer_rate"] <- dfsubway$transfers / dfsubway$ridership * 100
dfsubway["no_transfer"] <- dfsubway$ridership - dfsubway$transfers

dfsubway["dat"] <- substr(dfsubway$transit_timestamp, 0, 10)
dfsubway$wday <- mdy(dfsubway$dat)
dfsubway["dat"] <- NULL
dfsubway["dow"] <- wday(dfsubway$wday, label=TRUE)
dfsubway["wday"] <- NULL

dfsubway["weekday"] <- dfsubway["dow"]
dfsubway$weekday <- str_replace(dfsubway$weekday, "Mon", "1")
dfsubway$weekday <- str_replace(dfsubway$weekday, "Tue", "1")
dfsubway$weekday <- str_replace(dfsubway$weekday, "Wed", "1")
dfsubway$weekday <- str_replace(dfsubway$weekday, "Thu", "1")
dfsubway$weekday <- str_replace(dfsubway$weekday, "Fri", "1")
dfsubway$weekday <- str_replace(dfsubway$weekday, "Sat", "0")
dfsubway$weekday <- str_replace(dfsubway$weekday, "Sun", "0")

dfsubway$id <- 1:nrow(dfsubway)
dfsubway2 <- dfsubway[c("id", "hour", "weekday")]
dfsubway2["peak"] <- dfsubway2$hour
dfsubway3 <- dfsubway2[dfsubway2$weekday == "1",]
dfsubway2 <- dfsubway2[dfsubway2$weekday == "0",]

dfsubway2['peak'][dfsubway2['peak'] >= 18] <- -1
dfsubway2['peak'][dfsubway2['peak'] >= 10] <- -2
```

```r
dfsubway2['peak'][dfsubway2['peak'] >= 0] <- -1
dfsubway2['peak'][dfsubway2['peak'] == -1] <- 0
dfsubway2['peak'][dfsubway2['peak'] == -2] <- 1

dfsubway3['peak'][dfsubway3['peak'] >= 19] <- -1
dfsubway3['peak'][dfsubway3['peak'] >= 16] <- -2
dfsubway3['peak'][dfsubway3['peak'] >= 10] <- -1
dfsubway3['peak'][dfsubway3['peak'] >= 7] <- -2
dfsubway3['peak'][dfsubway3['peak'] >= 0] <- -1
dfsubway3['peak'][dfsubway3['peak'] == -1] <- 0
dfsubway3['peak'][dfsubway3['peak'] == -2] <- 1

dfsubway2 <- rbind(dfsubway2, dfsubway3)
rm(dfsubway3)
dfsubway2$hour <- NULL
dfsubway2$weekday <- NULL

dfsubway <- merge(dfsubway, dfsubway2, by = "id")
rm(dfsubway2)
dfsubway$id <- NULL

dfsubway["period"] <- paste(dfsubway$weekday, dfsubway$peak, sep = "")
dfsubway$weekday <- as.numeric(as.character(dfsubway$weekday))
dfsubway$peak <- as.numeric(as.character(dfsubway$peak))
dfsubway$period <- as.numeric(as.character(dfsubway$period))

dfsubway["weekend"] <- dfsubway["weekday"]
dfsubway['weekend'][dfsubway['weekend'] == 1] <- -1
dfsubway['weekend'][dfsubway['weekend'] == 0] <- 1
dfsubway['weekend'][dfsubway['weekend'] == -1] <- 0

dfsubway['offpeak'] <- dfsubway['peak']
dfsubway['offpeak'][dfsubway['offpeak'] == 1] <- -1
dfsubway['offpeak'][dfsubway['offpeak'] == 0] <- 1
dfsubway['offpeak'][dfsubway['offpeak'] == -1] <- 0

dfsubway$id <- NULL

dfsubway["wd_per_month"] <- paste(dfsubway$year, dfsubway$month, sep = "")
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 202210] <- 21
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 202211] <- 22
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 202212] <- 22
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20222] <- 20
```

```
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20223] <- 23
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20224] <- 21
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20225] <- 22
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20226] <- 22
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20227] <- 21
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20228] <- 23
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20229] <- 22
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20231] <- 22
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20232] <- 20
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20233] <- 23
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20234] <- 20
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20235] <- 23
dfsubway['wd_per_month'][dfsubway['wd_per_month'] == 20236] <- 22
dfsubway$wd_per_month <- as.numeric(as.character(dfsubway$wd_per_month))

dfsubway["days_per_month"] <- dfsubway$month
dfsubway['days_per_month'][dfsubway['days_per_month'] == 1] <- 31
dfsubway['days_per_month'][dfsubway['days_per_month'] == 2] <- 28
dfsubway['days_per_month'][dfsubway['days_per_month'] == 3] <- 31
dfsubway['days_per_month'][dfsubway['days_per_month'] == 4] <- 30
dfsubway['days_per_month'][dfsubway['days_per_month'] == 5] <- 31
dfsubway['days_per_month'][dfsubway['days_per_month'] == 6] <- 30
dfsubway['days_per_month'][dfsubway['days_per_month'] == 7] <- 31
dfsubway['days_per_month'][dfsubway['days_per_month'] == 8] <- 31
dfsubway['days_per_month'][dfsubway['days_per_month'] == 9] <- 30
dfsubway['days_per_month'][dfsubway['days_per_month'] == 10] <- 31
dfsubway['days_per_month'][dfsubway['days_per_month'] == 11] <- 30
dfsubway['days_per_month'][dfsubway['days_per_month'] == 12] <- 31

dfsubway["we_per_month"] <- dfsubway["days_per_month"] -
dfsubway["wd_per_month"]

dfsubway["line"] <- str_replace_all(dfsubway$routes, ",", "")

dfsubway["id"] <- paste(as.character(as.numeric(dfsubway$year)),
                        as.character(as.numeric(dfsubway$month)),
                        as.character(as.numeric(dfsubway$period)),
                        dfsubway$line,
                        sep = "")

dfsubway <- merge(dfsubway, dftrains, by = "id", all.x = TRUE)

dfsubway["period.y"] <- NULL
```

```r
dfsubway["month.y"] <- NULL
dfsubway["year.y"] <- NULL
dfsubway["peak.y"] <- NULL
dfsubway["weekday.y"] <- NULL
dfsubway["line.y"] <- NULL
dfsubway["id"] <- NULL
dfsubway["id2"] <- NULL

dfsubway <- dfsubway %>% rename_at('year.x', ~'year')
dfsubway <- dfsubway %>% rename_at('month.x', ~'month')
dfsubway <- dfsubway %>% rename_at('weekday.x', ~'weekday')
dfsubway <- dfsubway %>% rename_at('peak.x', ~'peak')
dfsubway <- dfsubway %>% rename_at('period.x', ~'period')
dfsubway <- dfsubway %>% rename_at('line.x', ~'line')

dfsubway['trains_per_day'] <- dfsubway$trains_per_month / (dfsubway$weekday *
dfsubway$wd_per_month + dfsubway$weekend * dfsubway$we_per_month)

dfsubway['trains_per_hour'] <- (dfsubway$trains_per_day / ((dfsubway$weekday
* dfsubway$peak * 6) + (dfsubway$weekday * dfsubway$offpeak * 18) +
(dfsubway$weekend * dfsubway$peak * 8) + (dfsubway$weekend * dfsubway$offpeak
* 16)))

dfsubway["trip_time"] <- paste(dfsubway$routes, ",")
dfsubway["trip_time"] <- str_replace_all(dfsubway$trip_time, " ", "")
dfsubway["trip_time"] <- str_replace_all(dfsubway$trip_time, ",,", ",")

dfsubway0 <- dfsubway[dfsubway$period == 0,]
dfsubway1 <- dfsubway[dfsubway$period == 1,]
dfsubway10 <- dfsubway[dfsubway$period == 10,]
dfsubway11 <- dfsubway[dfsubway$period == 11,]

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "1,", "116 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "1,", "116 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "1,", "114
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "1,", "113
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "2,", "170 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "2,", "187 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "2,", "185
,")
```

```
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "2,", "197
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "3,", "123 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "3,", "128 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "3,", "129
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "3,", "138
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "4,", "132 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "4,", "126 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "4,", "141
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "4,", "139
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "5,", "108 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "5,", "114 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "5,", "162
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "5,", "170
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "6,", "115 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "6,", "118 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "6,", "129
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "6,", "116
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "7,", "72 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "7,", "74 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "7,", "71
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "7,", "70.5
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "A,", "176.5
,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "A,", "177.5
,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "A,", "172
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "A,", "174
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "B,", "0 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "B,", "0 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "B,", "162
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "B,", "168
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "C,", "136 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "C,", "137 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "C,", "130
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "C,", "138
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "D,", "159 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "D,", "183 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "D,", "179
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "D,", "186
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "E,", "104 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "E,", "106 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "E,", "101
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "E,", "100
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "F,", "198 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "F,", "192 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "F,", "200
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "F,", "193
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "G,", "65 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "G,", "71 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "G,", "63
,")
```

```
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "G,", "72
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "J,", "104 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "J,", "111 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "J,", "107
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "J,", "105
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "L,", "81 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "L,", "74 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "L,", "81
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "L,", "71
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "M,", "56 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "M,", "55 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "M,", "133
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "M,", "140
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "N,", "148 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "N,", "150 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "N,", "140
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "N,", "145
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "Q,", "120 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "Q,", "125 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "Q,", "123
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "Q,", "130
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "R,", "175 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "R,", "176 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "R,", "168
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "R,", "176
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "W,", "0 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "W,", "0 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "W,", "82
,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "W,", "82
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "Z,", "0 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "Z,", "0 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "Z,", "0 ,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "Z,", "49
,")

dfsubway0['trip_time'] <- str_replace_all(dfsubway0$trip_time, "S,", "0 ,")
dfsubway1['trip_time'] <- str_replace_all(dfsubway1$trip_time, "S,", "0 ,")
dfsubway10['trip_time'] <- str_replace_all(dfsubway10$trip_time, "S,", "0 ,")
dfsubway11['trip_time'] <- str_replace_all(dfsubway11$trip_time, "S,", "0 ,")

dfsubway <- rbind(dfsubway0, dfsubway1, dfsubway10, dfsubway11)
rm(dfsubway0, dfsubway1, dfsubway10, dfsubway11)
dfsubway["trip_time"] <- str_replace_all(dfsubway$trip_time, " ", "")
dfsubway["trip_time"] <- substr(dfsubway$trip_time, 1,
nchar(dfsubway$trip_time)-1)
dfsubway2 <- str_split_fixed(dfsubway$trip_time, ",", 16)
dfsubway3 <- as.data.frame(dfsubway2)
dfsubway3$V2 <- sub("^$", "0", dfsubway3$V2)
dfsubway3$V3 <- sub("^$", "0", dfsubway3$V3)
dfsubway3$V4 <- sub("^$", "0", dfsubway3$V4)
dfsubway3$V5 <- sub("^$", "0", dfsubway3$V5)
dfsubway3$V6 <- sub("^$", "0", dfsubway3$V6)
dfsubway3$V7 <- sub("^$", "0", dfsubway3$V7)
dfsubway3$V8 <- sub("^$", "0", dfsubway3$V8)
dfsubway3$V9 <- sub("^$", "0", dfsubway3$V9)
dfsubway3$V10 <- sub("^$", "0", dfsubway3$V10)
dfsubway3$V11 <- sub("^$", "0", dfsubway3$V11)
dfsubway3$V12 <- sub("^$", "0", dfsubway3$V12)
dfsubway3$V13 <- sub("^$", "0", dfsubway3$V13)
dfsubway3$V14 <- sub("^$", "0", dfsubway3$V14)
dfsubway3$V15 <- sub("^$", "0", dfsubway3$V15)
dfsubway3$V16 <- sub("^$", "0", dfsubway3$V16)
```

```
dfsubway3$V1 <- as.numeric(dfsubway3$V1)
dfsubway3$V2 <- as.numeric(dfsubway3$V2)
dfsubway3$V3 <- as.numeric(dfsubway3$V3)
dfsubway3$V4 <- as.numeric(dfsubway3$V4)
dfsubway3$V5 <- as.numeric(dfsubway3$V5)
dfsubway3$V6 <- as.numeric(dfsubway3$V6)
dfsubway3$V7 <- as.numeric(dfsubway3$V7)
dfsubway3$V8 <- as.numeric(dfsubway3$V8)
dfsubway3$V9 <- as.numeric(dfsubway3$V9)
dfsubway3$V10 <- as.numeric(dfsubway3$V10)
dfsubway3$V11 <- as.numeric(dfsubway3$V11)
dfsubway3$V12 <- as.numeric(dfsubway3$V12)
dfsubway3$V13 <- as.numeric(dfsubway3$V13)
dfsubway3$V14 <- as.numeric(dfsubway3$V14)
dfsubway3$V15 <- as.numeric(dfsubway3$V15)
dfsubway3$V16 <- as.numeric(dfsubway3$V16)

dfsubway3$route_time <- rowSums(dfsubway3)

dfsubway3$V1 <- NULL
dfsubway3$V2 <- NULL
dfsubway3$V3 <- NULL
dfsubway3$V4 <- NULL
dfsubway3$V5 <- NULL
dfsubway3$V6 <- NULL
dfsubway3$V7 <- NULL
dfsubway3$V8 <- NULL
dfsubway3$V9 <- NULL
dfsubway3$V10 <- NULL
dfsubway3$V11 <- NULL
dfsubway3$V12 <- NULL
dfsubway3$V13 <- NULL
dfsubway3$V14 <- NULL
dfsubway3$V15 <- NULL
dfsubway3$V16 <- NULL

dfsubway <- cbind(dfsubway, dfsubway3)

rm(dfsubway2, dfsubway3)

dfsubway["headway"] <- (dfsubway$route_time / dfsubway$trains_per_hour) / 2
dfsubway["real_headway"] <- dfsubway$headway + dfsubway$additional_time
#unique(dfsubway$headway)
```

```r
dfsubway$id <- paste(as.character(as.numeric(dfsubway$year)),
                        as.character(as.numeric(dfsubway$month)),
                        as.character(as.numeric(dfsubway$period)),
                        dfsubway$line,
                        sep = "")
dfsubwayic <- dfsubway[nchar(dfsubway$line) >= 2,]
dfsubwayic$line1 <- substr(dfsubwayic$line, 0, 1)
dfsubwayic$line2 <- substr(dfsubwayic$line, 2, 2)
dfsubwayic$line3 <- substr(dfsubwayic$line, 3, 3)
dfsubwayic$line4 <- substr(dfsubwayic$line, 4, 4)
dfsubwayic$line5 <- substr(dfsubwayic$line, 5, 5)
dfsubwayic$line6 <- substr(dfsubwayic$line, 6, 6)
dfsubwayic$line7 <- substr(dfsubwayic$line, 7, 7)
dfsubwayic$line8 <- substr(dfsubwayic$line, 8, 8)
dfsubwayic$line9 <- substr(dfsubwayic$line, 9, 9)
dfsubwayic$line10 <- substr(dfsubwayic$line, 10, 10)
dfsubwayic$line11 <- substr(dfsubwayic$line, 11, 11)
dfsubwayic$line12 <- substr(dfsubwayic$line, 12, 12)
dfsubwayic$line13 <- substr(dfsubwayic$line, 13, 13)
dfsubwayic$line14 <- substr(dfsubwayic$line, 14, 14)
dfsubwayic$line15 <- substr(dfsubwayic$line, 15, 15)
dfsubwayic$line16 <- substr(dfsubwayic$line, 16, 16)
dfsubwayic$id <- paste(as.character(as.numeric(dfsubwayic$year)),
                        as.character(as.numeric(dfsubwayic$month)),
                        as.character(as.numeric(dfsubwayic$period)),
                        sep = "")
dfsubwayic$id1 <- paste(dfsubwayic$id, dfsubwayic$line1, sep="")
dfsubwayic$id2 <- paste(dfsubwayic$id, dfsubwayic$line2, sep="")
dfsubwayic$id3 <- paste(dfsubwayic$id, dfsubwayic$line3, sep="")
dfsubwayic$id4 <- paste(dfsubwayic$id, dfsubwayic$line4, sep="")
dfsubwayic$id5 <- paste(dfsubwayic$id, dfsubwayic$line5, sep="")
dfsubwayic$id6 <- paste(dfsubwayic$id, dfsubwayic$line6, sep="")
dfsubwayic$id7 <- paste(dfsubwayic$id, dfsubwayic$line7, sep="")
dfsubwayic$id8 <- paste(dfsubwayic$id, dfsubwayic$line8, sep="")
dfsubwayic$id9 <- paste(dfsubwayic$id, dfsubwayic$line9, sep="")
dfsubwayic$id10 <- paste(dfsubwayic$id, dfsubwayic$line10, sep="")
dfsubwayic$id11 <- paste(dfsubwayic$id, dfsubwayic$line11, sep="")
dfsubwayic$id12 <- paste(dfsubwayic$id, dfsubwayic$line12, sep="")
dfsubwayic$id13 <- paste(dfsubwayic$id, dfsubwayic$line13, sep="")
dfsubwayic$id14 <- paste(dfsubwayic$id, dfsubwayic$line14, sep="")
dfsubwayic$id15 <- paste(dfsubwayic$id, dfsubwayic$line15, sep="")
dfsubwayic$id16 <- paste(dfsubwayic$id, dfsubwayic$line16, sep="")

dfsubwayic$line1 <- NULL
```

```
dfsubwayic$line2 <- NULL
dfsubwayic$line3 <- NULL
dfsubwayic$line4 <- NULL
dfsubwayic$line5 <- NULL
dfsubwayic$line6 <- NULL
dfsubwayic$line7 <- NULL
dfsubwayic$line8 <- NULL
dfsubwayic$line9 <- NULL
dfsubwayic$line10 <- NULL
dfsubwayic$line11 <- NULL
dfsubwayic$line12 <- NULL
dfsubwayic$line13 <- NULL
dfsubwayic$line14 <- NULL
dfsubwayic$line15 <- NULL
dfsubwayic$line16 <- NULL


unique(dfsubwayic$trains_per_hour)
dfsubwayic$tph1 <- dfsubway$trains_per_hour[match(dfsubwayic$id1,
dfsubway$id)]
dfsubwayic$tph2 <- dfsubway$trains_per_hour[match(dfsubwayic$id2,
dfsubway$id)]
dfsubwayic$tph3 <- dfsubway$trains_per_hour[match(dfsubwayic$id3,
dfsubway$id)]
dfsubwayic$tph4 <- dfsubway$trains_per_hour[match(dfsubwayic$id4,
dfsubway$id)]
dfsubwayic$tph5 <- dfsubway$trains_per_hour[match(dfsubwayic$id5,
dfsubway$id)]
dfsubwayic$tph6 <- dfsubway$trains_per_hour[match(dfsubwayic$id6,
dfsubway$id)]
dfsubwayic$tph7 <- dfsubway$trains_per_hour[match(dfsubwayic$id7,
dfsubway$id)]
dfsubwayic$tph8 <- dfsubway$trains_per_hour[match(dfsubwayic$id8,
dfsubway$id)]
dfsubwayic$tph9 <- dfsubway$trains_per_hour[match(dfsubwayic$id9,
dfsubway$id)]
dfsubwayic$tph10 <- dfsubway$trains_per_hour[match(dfsubwayic$id10,
dfsubway$id)]
dfsubwayic$tph11 <- dfsubway$trains_per_hour[match(dfsubwayic$id11,
dfsubway$id)]
dfsubwayic$tph12 <- dfsubway$trains_per_hour[match(dfsubwayic$id12,
dfsubway$id)]
dfsubwayic$tph13 <- dfsubway$trains_per_hour[match(dfsubwayic$id13,
dfsubway$id)]
dfsubwayic$tph14 <- dfsubway$trains_per_hour[match(dfsubwayic$id14,
dfsubway$id)]
dfsubwayic$tph15 <- dfsubway$trains_per_hour[match(dfsubwayic$id15,
dfsubway$id)]
dfsubwayic$tph16 <- dfsubway$trains_per_hour[match(dfsubwayic$id16,
dfsubway$id)]

dfsubwayic <- dfsubwayic %>% mutate(tph1 = ifelse(is.na(tph1), 0, tph1))
dfsubwayic <- dfsubwayic %>% mutate(tph2 = ifelse(is.na(tph2), 0, tph2))
dfsubwayic <- dfsubwayic %>% mutate(tph3 = ifelse(is.na(tph3), 0, tph3))
dfsubwayic <- dfsubwayic %>% mutate(tph4 = ifelse(is.na(tph4), 0, tph4))
dfsubwayic <- dfsubwayic %>% mutate(tph5 = ifelse(is.na(tph5), 0, tph5))
dfsubwayic <- dfsubwayic %>% mutate(tph6 = ifelse(is.na(tph6), 0, tph6))
dfsubwayic <- dfsubwayic %>% mutate(tph7 = ifelse(is.na(tph7), 0, tph7))
dfsubwayic <- dfsubwayic %>% mutate(tph8 = ifelse(is.na(tph8), 0, tph8))
dfsubwayic <- dfsubwayic %>% mutate(tph9 = ifelse(is.na(tph9), 0, tph9))
dfsubwayic <- dfsubwayic %>% mutate(tph10 = ifelse(is.na(tph10), 0, tph10))
dfsubwayic <- dfsubwayic %>% mutate(tph11 = ifelse(is.na(tph11), 0, tph11))
dfsubwayic <- dfsubwayic %>% mutate(tph12 = ifelse(is.na(tph12), 0, tph12))
dfsubwayic <- dfsubwayic %>% mutate(tph13 = ifelse(is.na(tph13), 0, tph13))
dfsubwayic <- dfsubwayic %>% mutate(tph14 = ifelse(is.na(tph14), 0, tph14))
dfsubwayic <- dfsubwayic %>% mutate(tph15 = ifelse(is.na(tph15), 0, tph15))
dfsubwayic <- dfsubwayic %>% mutate(tph16 = ifelse(is.na(tph16), 0, tph16))

dfsubwayic$trains_per_hour <- dfsubwayic$tph1 + dfsubwayic$tph2 +
dfsubwayic$tph3 + dfsubwayic$tph4 + dfsubwayic$tph5 + dfsubwayic$tph6 +
dfsubwayic$tph7 + dfsubwayic$tph8 + dfsubwayic$tph9 + dfsubwayic$tph10 +
dfsubwayic$tph11 + dfsubwayic$tph12 + dfsubwayic$tph13 + dfsubwayic$tph14 +
dfsubwayic$tph15 + dfsubwayic$tph16

dfsubwayic$tph1 <- NULL
dfsubwayic$tph2 <- NULL
dfsubwayic$tph3 <- NULL
dfsubwayic$tph4 <- NULL
dfsubwayic$tph5 <- NULL
dfsubwayic$tph6 <- NULL
dfsubwayic$tph7 <- NULL
dfsubwayic$tph8 <- NULL
dfsubwayic$tph9 <- NULL
dfsubwayic$tph10 <- NULL
dfsubwayic$tph11 <- NULL
dfsubwayic$tph12 <- NULL
```

```
dfsubwayic$tph13 <- NULL
dfsubwayic$tph14 <- NULL
dfsubwayic$tph15 <- NULL
dfsubwayic$tph16 <- NULL

dfsubwayic$id1 <- NULL
dfsubwayic$id2 <- NULL
dfsubwayic$id3 <- NULL
dfsubwayic$id4 <- NULL
dfsubwayic$id5 <- NULL
dfsubwayic$id6 <- NULL
dfsubwayic$id7 <- NULL
dfsubwayic$id8 <- NULL
dfsubwayic$id9 <- NULL
dfsubwayic$id10 <- NULL
dfsubwayic$id11 <- NULL
dfsubwayic$id12 <- NULL
dfsubwayic$id13 <- NULL
dfsubwayic$id14 <- NULL
dfsubwayic$id15 <- NULL
dfsubwayic$id16 <- NULL

dfsubwaync <- dfsubway[nchar(dfsubway$line) < 2,]
dfsubway <- rbind(dfsubwaync, dfsubwayic)
rm(dfsubwayic, dfsubwaync)

dfsubway["average_headway"] <- (dfsubway$route_time /
dfsubway$trains_per_hour) / 2

dfsubway["dailyid"] <- paste(as.character(as.numeric(dfsubway$year)),
                          as.character(as.numeric(dfsubway$month)),
                          as.character(as.numeric(dfsubway$day)),
                          dfsubway$station_complex_id,
                          sep = "")
dfsubway2 <- aggregate(dfsubway$ridership, list(dfsubway$dailyid), FUN=sum)
dfsubway2 <- dfsubway2 %>% rename_at('Group.1', ~'dailyid')
dfsubway2 <- dfsubway2 %>% rename_at('x', ~'daily_ridership')
dfsubway <- merge(dfsubway, dfsubway2, by="dailyid")

rm(dfsubway2)

dfsubway3 <- dfsubway[dfsubway$peak == 0,]
dfsubway2 <- aggregate(dfsubway3$ridership, list(dfsubway3$dailyid), FUN=sum)
dfsubway2 <- dfsubway2 %>% rename_at('Group.1', ~'dailyid')
```

```
dfsubway2 <- dfsubway2 %>% rename_at('x', ~'offpeak_ridership')
dfsubway <- merge(dfsubway, dfsubway2, by="dailyid")
dfsubway$dailyid <- NULL
rm(dfsubway2, dfsubway3)

dfsubway["date"] <- paste(as.character(as.numeric(dfsubway$year)),
                          as.character(as.numeric(dfsubway$month)),
                          as.character(as.numeric(dfsubway$day)),
                          sep = "-")
dfsubway["week"] <- strftime(dfsubway$date, format="%V")
dfsubway$date <- NULL
dfsubway["dateid"] <- paste(as.character(as.numeric(dfsubway$year)),
                          as.character(as.numeric(dfsubway$week)),
                          dfsubway$station_complex_id,
                          sep = "")
dfsubway2 <- aggregate(dfsubway$ridership, list(dfsubway$dateid), FUN=sum)
dfsubway2 <- dfsubway2 %>% rename_at('Group.1', ~'dateid')
dfsubway2 <- dfsubway2 %>% rename_at('x', ~'weekly_ridership')
dfsubway <- merge(dfsubway, dfsubway2, by="dateid")
rm(dfsubway2)

dfsubway3 <- dfsubway[dfsubway$weekday == 0,]
dfsubway2 <- aggregate(dfsubway3$ridership, list(dfsubway3$dateid), FUN=sum)
dfsubway2 <- dfsubway2 %>% rename_at('Group.1', ~'dateid')
dfsubway2 <- dfsubway2 %>% rename_at('x', ~'weekend_ridership')
dfsubway <- merge(dfsubway, dfsubway2, by="dateid")
dfsubway$dateid <- NULL
rm(dfsubway2, dfsubway3)

dfsubway["peak_ridership"] <- dfsubway$daily_ridership -
dfsubway$offpeak_ridership
dfsubway["percent_offpeak"] <- dfsubway$offpeak_ridership /
dfsubway$daily_ridership
dfsubway["percent_peak"] <- 1 - dfsubway$percent_offpeak

dfsubway["weekday_ridership"] <- dfsubway$weekly_ridership -
dfsubway$weekend_ridership
dfsubway["percent_weekend"] <- dfsubway$weekend_ridership /
dfsubway$weekly_ridership
dfsubway["percent_weekday"] <- 1 - dfsubway$percent_weekend

dfsubway["percent_hour"] <- dfsubway$ridership / dfsubway$daily_ridership
dfsubway["percent_week"] <- dfsubway$daily_ridership /
dfsubway$weekly_ridership
```

```
df19p03["year"] <- 2019
df20p03["year"] <- 2020
df21p03["year"] <- 2021
dfp03 <- rbind(df19p03, df20p03, df21p03)
rm(df19p03, df20p03, df21p03)
dfp03 <- dfp03 %>% select(-ends_with('A'))
dfp03 <- dfp03 %>% select(-ends_with('M'))

df19p04["year"] <- 2019
df20p04["year"] <- 2020
df21p04["year"] <- 2021
dfp04 <- rbind(df19p04, df20p04, df21p04)
rm(df19p04, df20p04, df21p04)
dfp04 <- dfp04 %>% select(-ends_with('A'))
dfp04 <- dfp04 %>% select(-ends_with('M'))

df19p05["year"] <- 2019
df20p05["year"] <- 2020
df21p05["year"] <- 2021
dfp05 <- rbind(df19p05, df20p05, df21p05)
rm(df19p05, df20p05, df21p05)
dfp05 <- dfp05 %>% select(-ends_with('A'))
dfp05 <- dfp05 %>% select(-ends_with('M'))

dfcensus <- cbind(dfp03, dfp04, dfp05)
rm(dfp03, dfp04, dfp05)

dup_df <- duplicated(colnames(dfcensus))
dfcensus <- dfcensus[!dup_df]
rm(dup_df)

dfcensus["geoid"] <- substr(dfcensus$GEO_ID, 10, 20)
dfcensus[1, 1] = "geography"
dfcensus[1, 2] = "ct_name"
dfcensus[1, 278] = "year"
dfcensus[1, 745] = "-1"
dfcensus$...1099 <- NULL
dfcensus$...1147 <- NULL
dfcensus$...715 <- NULL

dfcensus <- dfcensus %>% select(-ends_with('PE'))

census_tracts$cdeligibil <- NULL
census_tracts2 <- census_tracts[1,]
```

```
census_tracts2[1, 10] = "-1"
census_tracts <- rbind(census_tracts, census_tracts2)
rm(census_tracts2)

write.csv(dfsubway, "DIS/data/dfsubway.csv")
write.csv(dfcensus, "DIS/data/dfcensus.csv")

gdfcensus <- merge(dfcensus, census_tracts, by = "geoid")

gdfcensus["population"] <- as.numeric(as.character(gdfcensus$DP05_0001E))
gdfcensus["population_density"] <- gdfcensus$population / (
as.numeric(as.character(gdfcensus$shape_area)) * 0.0000000359)
gdfcensus["percent_male"] <- as.numeric(as.character(gdfcensus$DP05_0002E)) /
gdfcensus$population
gdfcensus["percent_female"] <- as.numeric(as.character(gdfcensus$DP05_0003E))
/ gdfcensus$population

gdfcensus["adult_pop"] <- as.numeric(as.character(gdfcensus$DP03_0001E))
gdfcensus["workforce"] <- as.numeric(as.character(gdfcensus$DP03_0002E))
gdfcensus["workforce_rate"] <- gdfcensus$workforce / gdfcensus$adult_pop
gdfcensus["female_workforce_rate"] <-
as.numeric(as.character(gdfcensus$DP03_0010E)) / gdfcensus$workforce
gdfcensus["employment_rate"] <-
as.numeric(as.character(gdfcensus$DP03_0004E)) / gdfcensus$workforce
gdfcensus["unemployment_rate"] <-
as.numeric(as.character(gdfcensus$DP03_0005E)) / gdfcensus$workforce

gdfcensus["commute_pop"] <- as.numeric(as.character(gdfcensus$DP03_0018E))
#gdfcensus["percent_commute"] <- gdfcensus$commuting_population /
as.numeric(as.character(gdfcensus$DP03_0004E))
gdfcensus["percent_drive"] <- (as.numeric(as.character(gdfcensus$DP03_0019E))
+ as.numeric(as.character(gdfcensus$DP03_0020E))) / gdfcensus$commute_pop
gdfcensus["percent_pt"] <- as.numeric(as.character(gdfcensus$DP03_0021E)) /
gdfcensus$commute_pop
gdfcensus["percent_walk"] <- as.numeric(as.character(gdfcensus$DP03_0022E)) /
gdfcensus$commute_pop
gdfcensus["percent_wfh"] <- as.numeric(as.character(gdfcensus$DP03_0024E)) /
gdfcensus$commute_pop
gdfcensus["mean_commute_time"] <-
as.numeric(as.character(gdfcensus$DP03_0025E))

#households
gdfcensus["households"] <- as.numeric(as.character(gdfcensus$DP03_0051E))
gdfcensus["med_house_inc"] <- as.numeric(as.character(gdfcensus$DP03_0062E))
```

```r
gdfcensus["mean_house_inc"] <- as.numeric(as.character(gdfcensus$DP03_0063E))
gdfcensus["per_capita_inc"] <- as.numeric(as.character(gdfcensus$DP03_0088E))
#add male, female, family, if want

gdfcensus <- gdfcensus %>% select(-starts_with('DP0'))

gdfcensus <- gdfcensus[-c(1), ]

gdfcensus$GEO_ID <- NULL
gdfcensus$ctlabel <- NULL
gdfcensus <- gdfcensus %>% rename_at('NAME', ~'name')
gdfcensus <- gdfcensus %>% drop_na()

gdf2 <- aggregate(list(gdfcensus[,15:34]), list(gdfcensus$geoid), FUN=mean)
gdf2 <- gdf2 %>% rename_at("Group.1", ~"geoid")
gdfcensus[,15:34] <- NULL
gdfcensus["year"] <- NULL
gdfcensus <- gdfcensus[!duplicated(gdfcensus), ]
gdfcensus <- merge(gdf2, gdfcensus, by="geoid")
rm(gdf2)

gdfcensus <- st_as_sf(gdfcensus)
gdfcensus <- st_set_crs(gdfcensus, "EPSG:4326")
gdfcensus <- st_transform(gdfcensus, "EPSG:2263")

gborough <- st_as_sf(boroughs)
gborough <- st_set_crs(gborough, "EPSG:4326")
gborough <- st_transform(gborough, "EPSG:2263")

glines <- st_as_sf(lines)
glines <- st_set_crs(glines, "EPSG:4326")
glines <- st_transform(glines, "EPSG:2263")

dfs <- sample_n(dfsubway, nrow(dfsubway)*0.01) # standard is 0.01
#rm(dfsubway)

gdfsubway <- dfs
gdfsubway$Georeference <- NULL
gdfsubway["long"] <- gdfsubway$longitude
gdfsubway["lat"] <- gdfsubway$latitude
gdfsubway <- st_as_sf(gdfsubway, coords = c("longitude", "latitude"))
gdfsubway <- st_set_crs(gdfsubway, "EPSG:4326")
gdfsubway <- st_transform(gdfsubway, "EPSG:2263")
```

```r
buffer <- st_buffer(gdfsubway, dist=3280.84) # 1 km 'dist' in feet
gdf <- st_join(gdfcensus, buffer, st_covered_by) #for OLS/FE robustness
gdf2 <- st_join(gdfcensus, gdfsubway) #for OLS/FE standard
gdf3 <- st_join(gdfcensus, buffer, st_covered_by) #just for mapping
rm(buffer)

#rm(gdfsubway)
gdf["ct_id"] <- paste(as.character(as.numeric(gdf$year)),
                      as.character(as.numeric(gdf$month)),
                      as.character(as.numeric(gdf$day)),
                      as.character(as.numeric(gdf$day)),
                      gdf$station_complex_id,
                      sep = "")
total_pop <- aggregate(gdf$population, by = list(gdf$ct_id), FUN=sum)
total_pop <- total_pop %>% rename_at('Group.1', ~'ct_id')
total_pop <- total_pop %>% rename_at('x', ~'total_population')
gdf <- merge(gdf, total_pop, by="ct_id")
gdf["agg_inc"] <- gdf$population * gdf$per_capita_inc / gdf$total_population
gdf["agg_pop_dens"] <- gdf$population * gdf$population_density /
gdf$total_population
gdf["agg_employment_rate"] <- gdf$population * gdf$employment_rate /
gdf$total_population
gdf["agg_percent_pt"] <- gdf$population * gdf$percent_pt /
gdf$total_population
gdf["agg_mean_commute_time"] <- gdf$population * gdf$mean_commute_time /
gdf$total_population
avg1 <- aggregate(gdf$agg_inc, by=list(gdf$ct_id), FUN=sum)
avg2 <- aggregate(gdf$agg_pop_dens, by=list(gdf$ct_id), FUN=sum)
avg3 <- aggregate(gdf$agg_employment_rate, by=list(gdf$ct_id), FUN=sum)
avg4 <- aggregate(gdf$agg_percent_pt, by=list(gdf$ct_id), FUN=sum)
avg5 <- aggregate(gdf$agg_mean_commute_time, by=list(gdf$ct_id), FUN=sum)
avg1 <- avg1 %>% rename_at('x', ~'n_income')
avg2 <- avg2 %>% rename_at('x', ~'n_pop_dens')
avg3 <- avg3 %>% rename_at('x', ~'n_employment_rate')
avg4 <- avg4 %>% rename_at('x', ~'n_percent_pt')
avg5 <- avg5 %>% rename_at('x', ~'n_mean_commute_time')
avg_tot <- merge(avg1, avg2, by="Group.1")
avg_tot <- merge(avg_tot, avg3, by="Group.1")
avg_tot <- merge(avg_tot, avg4, by="Group.1")
avg_tot <- merge(avg_tot, avg5, by="Group.1")
rm(avg1, avg2, avg3, avg4, avg5)
avg_tot <- avg_tot %>% rename_at('Group.1', ~'ct_id')
gdf <- gdf2
gdf["ct_id"] <- paste(as.character(as.numeric(gdf$year)),
                          as.character(as.numeric(gdf$month)),
                          as.character(as.numeric(gdf$day)),
                          as.character(as.numeric(gdf$day)),
                          gdf$station_complex_id,
                          sep = "")
gdf <- merge(gdf, avg_tot, by="ct_id")
rm(avg_tot)

gdf["line_count"] <- str_length(gdf$line)
gdf["log_rs"] <- log(gdf$ridership)
gdf["log_inc"] <- log(gdf$n_income)
gdf["log_pop_dens"] <- log(gdf$n_pop_dens)
gdf["log_no_trans"] <- log(gdf$no_transfer)
gdfz <- gdf[gdf$transfers == 0,]
gdfz["log_trans"] <- gdfz$transfers
gdf <- gdf[gdf$transfers != 0,]
gdf["log_trans"] <- log(gdf$transfers)
gdf <- rbind(gdf, gdfz)
gdfz <- gdf[gdf$trains_per_hour == 0,]
gdfz["log_tph"] <- gdfz$trains_per_hour
gdf <- gdf[gdf$trains_per_hour != 0,]
gdf["log_tph"] <- log(gdf$trains_per_hour)
gdf <- rbind(gdf, gdfz)
rm(gdfz)

#rm(gdfsubway)
gdf2["line_count"] <- str_length(gdf2$line)
gdf2["log_rs"] <- log(gdf2$ridership)
gdf2["log_inc"] <- log(gdf2$per_capita_inc)
gdf2["log_pop_dens"] <- log(gdf2$population_density)
gdf2["log_no_trans"] <- log(gdf2$no_transfer)
gdfz <- gdf2[gdf2$transfers == 0,]
gdfz["log_trans"] <- gdfz$transfers
gdf2 <- gdf2[gdf2$transfers != 0,]
gdf2["log_trans"] <- log(gdf2$transfers)
gdf2 <- rbind(gdf2, gdfz)
gdfz <- gdf2[gdf2$trains_per_hour == 0,]
gdfz["log_tph"] <- gdfz$trains_per_hour
gdf2 <- gdf2[gdf2$trains_per_hour != 0,]
gdf2["log_tph"] <- log(gdf2$trains_per_hour)
gdf2 <- rbind(gdf2, gdfz)
rm(gdfz)
```

```r
gdf["n_month"] <- paste(as.character(as.numeric(gdf$year)),
                        as.character(as.numeric(gdf$month)),
                        gdf$period,
                        sep = "-")
gdf2["n_month"] <- paste(as.character(as.numeric(gdf2$year)),
                        as.character(as.numeric(gdf2$month)),
                        gdf2$period,
                        sep = "-")

gdf1 <- gdf[gdf$hour > 5,]
gdf1 <- gdf1[gdf1$hour <= 23,]
gdf21 <- gdf2[gdf2$hour > 5,]
gdf21 <- gdf21[gdf21$hour <= 23,]

gdf1["log_rhw"] <- log(gdf1$real_headway)

ols1.1 <- feols(gdf2, log_rs ~ log_tph,vcov = "hetero")
ols2.4 <- feols(gdf2, log_rs ~ log_tph + log_trans + weekend + offpeak +
month + year + log_inc + log_pop_dens + employment_rate + percent_pt +
mean_commute_time + line_count, vcov = "hetero") #w/o buffer
ols5 <- feols(gdf, log_rs ~ log_tph + log_trans + weekend + offpeak + month +
year + log_inc + log_pop_dens + n_employment_rate + n_percent_pt +
n_mean_commute_time + line_count, vcov = "hetero") #w/ buffer
ols6 <- feols(gdf1, log_rs ~ log_tph + log_trans + weekend + offpeak + month
+ year + log_inc + log_pop_dens + n_employment_rate + n_percent_pt +
n_mean_commute_time + line_count, vcov = "hetero") #w/ buffer

fe3.1 <- feols(gdf2, log_rs ~ log_tph + log_trans + percent_weekend +
percent_offpeak + log_inc + log_pop_dens + employment_rate + percent_pt +
mean_commute_time + factor(line_count) + factor(n_month), vcov = "hetero")
#w/o buffer

fe4.1 <- feols(gdf, log_rs ~ log_tph + log_trans + log_inc + log_pop_dens +
employment_rate + percent_pt + mean_commute_time + + percent_weekend +
percent_offpeak + factor(line_count) + factor(n_month), vcov = "hetero") #w/
buffer

fe5.1 <- feols(gdf1, log_rs ~ log_tph + log_trans + log_inc + log_pop_dens +
employment_rate + percent_pt + mean_commute_time + + percent_weekend +
percent_offpeak + factor(line_count) + factor(n_month), vcov = "hetero") #w/
buffer

fe6.1 <- feols(gdf1, log_rs ~ log_rhw + log_trans + log_inc + log_pop_dens +
employment_rate + percent_pt + mean_commute_time + + percent_weekend +
percent_offpeak + factor(line_count) + factor(n_month), vcov = "hetero") #w/
buffer

daily_ridership <-
read_csv("DIS/data/MTA_Daily_Ridership_Data__Beginning_2020.csv")
daily_ridership["year"] <- substr(daily_ridership$Date, 7, 10)

daily_ridership["month"] <- substr(daily_ridership$Date, 0, 2)

daily_ridership["ym"] <- paste(daily_ridership$year, daily_ridership$month,
sep = "")
daily_ridership$ridership <- daily_ridership$`Subways: % of Comparable
Pre-Pandemic Day`
dr2 <- aggregate(daily_ridership$ridership, by = list(daily_ridership$ym),
FUN=sum)
dr2 <- dr2 %>% rename_at('Group.1', ~'ym')
dr2 <- dr2 %>% rename_at('x', ~'month_ridership')
daily_ridership <- merge(daily_ridership, dr2, by="ym")
rm(dr2)
daily_ridership["time"] <- as.Date(daily_ridership$Date, "%m/%d/%Y")

daily_ridership["dow"] <- wday(daily_ridership$time, label=TRUE)
daily_ridership$dow <- str_replace(daily_ridership$dow, "Mon", "Weekday")
daily_ridership$dow <- str_replace(daily_ridership$dow, "Tue", "Weekday")
daily_ridership$dow <- str_replace(daily_ridership$dow, "Wed", "Weekday")
daily_ridership$dow <- str_replace(daily_ridership$dow, "Thu", "Weekday")
daily_ridership$dow <- str_replace(daily_ridership$dow, "Fri", "Weekday")
daily_ridership$dow <- str_replace(daily_ridership$dow, "Sat", "Weekend")
daily_ridership$dow <- str_replace(daily_ridership$dow, "Sun", "Weekend")

plot1 <- ggplot(data = daily_ridership, aes(x=time, y=ridership, color=dow))
+
  geom_point() +
  theme(rect = element_blank()) +
  labs(x="Date", y="% of Pre-Pandemic Subway Ridership Levels", color="Day of
the Week")
plot1

plot2 <- ggplot(dfs, aes(x=ridership))+
geom_histogram(color="white", fill ="darkgreen")+
  theme(rect = element_blank()) +
  labs(x="Hourly Ridership", y="Frequency") +
  scale_x_log10()
plot2

plot3 <- ggplot(gdf, aes(x=trains_per_hour))+
geom_histogram(color="white", fill ="purple")+
  theme(rect = element_blank()) +
  labs(x="Trains Per Hour", y="Frequency") #+
  #scale_x_log10()
plot3

map1 <- ggplot()+
  geom_sf(data = gborough, fill = "lightgrey") +
  geom_sf(data = gdfsubway, size = 0.2, color = "black") +
  geom_sf(data = glines, size = 0.1, color = "darkblue") +
  theme(rect = element_blank()) +
  labs(x="Longitude", y="Latitude")
map1

map2 <- ggplot()+
  geom_sf(data = gborough, fill = "lightgrey") +
  geom_sf(data = gdfcensus, fill = "darkgrey") +
  theme(rect = element_blank(), legend.position="none") +
  labs(x="Longitude", y="Latitude")
map2

gdf3$stationarea <- gdf3$ridership / gdf3$ridership
gdf3 <-drop_na(gdf3, c("stationarea"))
gborough2 <- gborough[gborough$boro_name != "Staten Island",]
census_tracts2 <- census_tracts[census_tracts$boroname != "Staten Island",]

map3 <-ggplot()+
  geom_sf(data = gborough2, fill = "lightgrey") +
  geom_sf(data = census_tracts2, fill = "lightgrey") +
  geom_sf(data = gdf3, aes(fill = stationarea)) +
  scale_fill_viridis(direction = -1, option = "D") +
  geom_sf(data = gdf2, fill = "black") +
  theme(rect = element_blank(),
        axis.text.x = element_text(size=5),
        axis.text.y = element_text(size=5),
        legend.position="none") +
  labs(x="Longitude", y="Latitude")
rm(census_tracts2, gborough2)
map3
```