Developing Massachusetts-Specific Trip Generation Models for Land Use Projects

Principal Investigator (s)
Dr. Yuanchang Xie
Dr. Benyuan Liu
Dr. Danjue Chen

University of Massachusetts Lowell

Research and Technology Transfer Section
MassDOT Office of Transportation Planning

Trip Generation Manual
10th Edition • Volume 1: Desk Reference

U.S. Department of Transportation
Federal Highway Administration

Technical Report Document Page

1. Report No. 23-046	2. Government Accession No. n/a	3. Recipient's Catalog No. n/a		
4. Title and Subtitle Developing Massachusetts-Specific Trip Generation Models for Land Use Projects		5. Report Date September 2023		
		6. Performing Organization Code 23-046		
7. Author(s) Yuanchang Xie, Brandon Loesch, Zubin Bhuyan, Francesco Logozzo, Danjue Chen, and Benyuan (Ben) Liu		8. Performing Organization Report No.		
9. Performing Organization Name and Address University of Massachusetts Lowell 1 University Avenue Lowell, MA 01854		10. Work Unit No. (TRAIS) n/a		
		11. Contract or Grant No. 113773		
12. Sponsoring Agency Name and Address Massachusetts Department of Transportation Office of Transportation Planning 10 Park Plaza, Room 4150 Boston, MA 02116-3969		13. Type of Report and Period Covered Final Report (September 2023) (March 2021- September 2023)		
		14. Sponsoring Agency Code n/a		

15. Supplementary Notes
Project Champion - Lionel Lucien - MassDOT

16. Abstract

Vehicle trip generation is used to identify potential transportation impacts associated with new development projects and to provide a substantive basis for determining appropriate impact mitigation strategies. In the United States, trip rate estimation typically relies on vehicle trip rates for specific land use types as provided in the Trip Generation Manual published and updated by the Institute of Transportation Engineers (ITE). The ITE manual uses trip generation data derived primarily from suburban project sites that are highly automobile-dependent. ITE trip generation rates have been found to be sometimes overestimated for urban sites that typically benefit from their proximity to public transportation. This research aims to develop Massachusetts-specific trip generation models for high-priority land uses using location-based services (LBS) data. The accuracy of the LBS data is then validated using ground truth data derived from traffic videos captured at some selected development sites. The recorded videos are processed using a deep learning model to derive vehicle and person counts. The proposed approach based on LBS data can significantly reduce the efforts and time needed to collect trip generation data for model fitting, making it possible to frequently update trip generation models and develop more sophisticated models to capture temporal changes in trip generations. The computer vision and deep learning model provides a useful tool for MassDOT to collect ground truth trip generation data in the future for model development and validation purposes. A comparison of the trip generation results from different methods suggests that the LBS estimates are reasonably accurate and can be used as a promising data source for developing Massachusetts trip generation models.

17. Key Word Institute of Transportation Engineers(ITE), vehicle trip generation, land use development, location-based services		18. Distribution Statement		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified		21. No. of Pages 109	22. Price n/a

Developing Massachusetts-Specific Trip Generation Models for Land Use Projects

Final Report

Prepared By:

Yuanchang Xie, PhD, PE, Professor
Brandon Loesch, Graduate Research Assistant
Francesco Logozzo, Undergraduate Research Assistant
Danjue Chen, PhD, Associate Professor
Department of Civil and Environmental Engineering
University of Massachusetts Lowell

Benyuan Liu, PhD, Professor
Zubin Bhuyan, Graduate Research Assistant
Department of Computer Science
University of Massachusetts Lowell

Prepared For:

Massachusetts Department of Transportation
Office of Transportation Planning
Ten Park Plaza, Suite 4150
Boston, MA 02116

September 2023

Acknowledgements

Disclaimer

Executive Summary

Vehicle trip generation is used to identify potential transportation impacts associated with new development projects and to provide a substantive basis for determining appropriate impact mitigation strategies and informing transportation infrastructure management, planning, and public involvement. In the United States, trip rate estimation typically relies on vehicle trip rates for specific land use types as provided in the Trip Generation Manual published and updated by the Institute of Transportation Engineers (ITE). The ITE manual uses trip generation data derived primarily from suburban project sites that are highly auto-dependent. ITE trip generation rates have been found to be sometimes overestimated for urban sites that typically benefit from their proximity to public transportation. For some land use categories, the ITE data collection sample sizes are relatively small, with data collected many years ago. Many states have conducted research to validate and modify the ITE trip generation information to better reflect the current trip generation trends observed locally. However, collecting and updating trip rate data is both time-consuming and expensive because data is typically collected by manual counting. Each year, different divisions of MassDOT collect a large amount of trip data, which consumes a lot of resources. This limits the amount and frequency of trip generation data (e.g., sample size) that can be collected.

This research utilizes location-based service (LBS) data to develop Massachusetts-specific trip generation models. Location-based services gather and anonymize mobile device user data, which can easily generate trip rates within specific areas and time windows. Collecting LBS data is much easier and provides many advantages over traditional data collection methods. Using LBS data, models are developed in this study for several high-priority land uses. While the results for some land use categories are not particularly strong, there is a strong correlation between the LBS trip generations and explanatory variables for mid-rise multifamily housing (land use code 221). The corresponding model fitting R^2 based on 2018 LBS data is 0.89.

In addition, this project provides detailed guidance on the process of model development for the Massachusetts Department of Transportation (MassDOT). This guidance could be used for updating the models developed in this study (when new data is available) or for modeling other land use categories.

The research team has also developed a procedure for video data collection and automatic processing to evaluate the accuracy of LBS data and ITE trip generation models. Trail cameras are used to collect traffic videos, and then AI algorithms are used to process the video data and automatically generate trip counts. Applying this procedure (particularly to the mid-rise multifamily housing land use), the team has evaluated the accuracy of LBS and ITE trip estimates against ground truth data collected through traffic videos. The results showed that the absolute percentage error of the LBS trip data was lower than that of the ITE estimates. Given that it is much easier to collect LBS data, this study demonstrates the feasibility and great potential of using LBS data for future trip generation modeling.

Table of Contents

List of Tables

List of Figures

List of Acronyms

Acronym	Expansion
AADT	Annual Average Daily Traffic
AI	Artificial Intelligence
COVID	Coronavirus Disease
CTPP	Census Transportation Planning Program
DOT	Department of Transportation
EPA	Environmental Protection Agency
FPN	Feature Pyramid Network
FDOT	Florida Department of Transportation
GIS	Geographic Information System
GPU	Graphics Processing Units
GPS	Global Positioning System
HBO	Home-Based Other
HBW	Home-Based Work
HTS	Household Travel Survey
IR	Infrared
ITC	Internal Trip Capture
ITE	Institute of Transportation Engineers
LBS	Location-Based Service
LUC	Land Use Code
MassDOT	Massachusetts Department of Transportation
MDS	Multi-Dimensional Similarity
MNL	Multinomial Logistic
MPO	Metropolitan Planning Organization
MTC	Metropolitan Transportation Commission
MXD	Mixed-Use Development
NCHRP	National Cooperative Highway Research Program
NHB	Non-Home-Based
OD	Origin and Destination
SPP	Spatial Pyramid Pooling
TAP	Trip Analysis Polygon
TAZ	Traffic Analysis Zone

Acronym	Expansion
TDM	Travel Demand Management
TPP	Tax Parcel Polygon
TxDOT	Texas Department of Transportation
UMass	University of Massachusetts
VMT	Vehicle Miles Traveled
WGS	World Geodetic System
YOLO	You Only Look Once

1.0 Introduction

Accurate trip generation estimation is critical for quantifying the potential transportation impacts associated with new development projects and for providing a substantive basis to determine appropriate impact mitigations and inform transportation infrastructure management, planning, and public involvement. In the United States, trip generation estimation typically relies on vehicle trip rates for specific land use types as provided in the *Trip Generation Manual* published and updated by the Institute of Transportation Engineers (ITE). The ITE manual uses trip generation data derived mainly from project sites that are highly automobile-dependent. ITE trip generation rates have been found to sometimes overestimate trips for sites near public transportation and with easy access to walking and bicycling, such as many places in Massachusetts. Also, the ITE data collection sample sizes for some land-use categories are relatively small, with data collected many years ago. This small sample size may compromise the accuracy of the trip generation estimates. Additionally, old data does not reflect the impacts of cultural and societal changes such as working from home and online shopping on trip generation, and can lead to inaccurate trip generation estimates as well.

Over- and underestimation of trip generation may result in either significant waste and unfair financial burden on developers or inadequate infrastructure to support an area's economic activities. Improved trip generation estimates will help planning agencies accurately identify the impacts of new developments and propose appropriate action plans and policies. Several states have conducted research to validate and modify the ITE trip generation information to better reflect the current trip generation trends observed locally. However, collecting trip generation data manually is both time-consuming and expensive. This limits the amount and frequency of trip generation data that can be collected. Instead of relying on manual collection, this study investigates the feasibility of using nontraditional location-based service (LBS) data sources for estimating trip generation rates. Specifically, StreetLight data is used in this research, but there are other similar data products that can also be used. If this approach is proven to be accurate, it can provide a very cost-effective tool for trip generation data collection and significantly benefit MassDOT and other state Departments of Transportation (DOTs).

One major task of this research is to assist MassDOT in developing trip generation models for selected high-priority land uses in Massachusetts based on the StreetLight data. As an important part of this effort, a computer vision and deep learning tool has been developed to facilitate the collection of multimodal trip generation ground truth data. The ground truth data is further compared to the trip generation results derived from the StreetLight data to evaluate its accuracy. Detailed steps to derive trip generation data from StreetLight are clearly documented so that MassDOT can follow such steps to develop trip generation models for other types of land use or update the models developed in this study. MassDOT can also utilize the developed computer vision and deep learning tool to collect additional trip generation ground truth data for future model development and validation efforts.

This report is organized as follows: Chapter 2 reviews existing studies on trip generation adjustments and data collection methods; Chapter 3 describes how data for the trip generation models are collected in this project and the results of the developed models; Chapter 4 describes the method for developing and updating the models for future use by MassDOT; Chapter 5 describes the process of ground truth data collection and evaluates the accuracy of Streetlight data; and Chapter 6 summarizes the study, describes limitations, and provides recommendations for future work.

2.0 Review of Literature and Best Practices

Accurate trip generation estimation is critical to assessing the transportation impacts of development projects, and has attracted much attention over the years. A comprehensive review of these efforts is conducted in this research with a focus on trip generation estimation and data collection methods. The literature review is organized into the following categories:

- Adjusting ITE trip generation rates for dense and urban areas,
- Establishing local trip generation adjustment factors and model development,
- Novel data collection methods and variable exploration, and
- Comparison of methodologies used in previous studies.

2.1 Adjusting ITE Trip Generation Rates for Dense and Urban Areas

Several studies have been conducted detailing unique methodologies to determine adjustment factors for ITE trip generation rates. In these studies, some variables were identified that can be used to improve trip generation rates estimation accuracy. However, the feasibility of utilizing these methodologies and variables depends on data availability or data collection efforts.

Clifton et al. (*1*) collected person and vehicle trip rates data using manual counts and intercept surveys of 78 locations (including restaurants, convenience stores, and drinking places) to address the issue of inaccurate trip generation estimates for new developments in urban areas. Data was specifically collected between 5 and 7 p.m. on weekdays without rain. They found that the ITE handbook significantly overestimated vehicle trips for convenience stores and drinking places but produced good estimates for restaurants. They also surveyed 1,884 customers at these locations and collected information about travel mode, vehicle occupancy, home location, etc. Square footages of the businesses were obtained by contacting the managers, retrieved from regional building information, or estimated using Google Earth. The number of people entering and exiting buildings was manually counted. Similarly, vehicle trips were counted if the business had dedicated parking spaces, but if not, were estimated using the equation below:

$$T_{veh} = \frac{(P_{in}+P_{out})_{Obs}(\%AUTO)_{Survey}}{Veh\_Occ_{Survey}} \times \frac{1}{1000\ sqft} \tag{1}$$

where

$T_{veh} = estimated\ vehicle\ trips$;
$P_{in} = observed\ entering\ person\ count$;
$P_{out} = observed\ exiting\ person\ count$;
$\%AUTO = automobile\ mode\ share\ (from\ intercept\ surveys)$; and
$Veh_{occ} = average\ vehicle\ occupancy\ (from\ intercept\ surveys)$.

Nine trip generation rate models were developed considering different built environment characteristics within 0.5 miles of the development. Because strong correlations were found between these factors, each model only included one built environment factor as shown in Equation (2) below. Given the small sample size, they did not develop separate models for each land use. Instead, data from three land use types were combined and two dummy variables were included for convenience stores and restaurants. These models estimated the difference between observed vehicle trips and ITE vehicle trips. The results suggest that all nine models were able to generate reasonably accurate estimates with adjusted R^2 between 0.76 and 0.77.

$$Adj = \beta_0 + \beta_1 Dummy_{Conv} + \beta_2 Dummy_{Res} + \beta_3 Built\_Env \tag{2}$$

where
Adj = Observed trip rate minus ITE trip rate;
$Dummy_{Conv} = 1$ if the land-use type is convenience store, and 0 otherwise;
$Dummy_{Res} = 1$ if the land-use type is restaurant, and 0 otherwise; and
$Built\_Env$ = one of the following built environment variables: activity density, number of transit corridors, number of high-frequency bus lines, employment density, lot coverage, length of bicycle facilities, presence of rail transit, retail and service employment index, and intersection density.

The models were further evaluated at 34 additional establishments with varying built environments. The models performed clearly better than the ITE method except for restaurants.

Currans and Clifton (*2*) collected household travel survey data from Oregon (2011 Oregon Household Activity Survey), Washington (2006 Puget Sound Regional Travel Survey), and Maryland (2001 National Household Travel Survey Add-On Program), and built environment data from the 2000 Census Summary File 3, the 2000 Census Transportation Planning Package, and more. The built environments were measured for a 0.5-mile buffer area surrounding each site. They proposed three different ITE trip generation rate adjustments. Adjustment A is a table of mode share for eight land use types considering activity density. For Adjustments B and C, binary logistic regression models were used to estimate the probabilities for automobile and non-automobile trips. B and C chose a built environment variable with sensitivity to land use policies, which is different from A. In addition, linear regression models were developed to estimate vehicle occupancy rates.

On average, all three adjustments performed better than the ITE handbook. This demonstrates that urban context adjustments should be considered when using the ITE handbook trip generation rates for infill developments in densely populated areas with strong built environment support for non-automobile modes. Of the three adjustments, the simplest one was A, which performed similarly to the other, more complex methods.

In 2019, Clifton and Currans (*3*) also estimated the vehicle and person trip generation rates associated with some land use categories in the ITE Trip Generation Handbook to answer the following questions:

1. Does the built environment vary across various ITE Land Use Codes for multifamily housing?
2. How do vehicle trip rates and newly established person trip rates vary across urban locations?
3. How well does ITE's recommended practice of converting their vehicle trip rate data to person trip rates perform?

The trip generation data for multifamily housing was extracted from four previous studies, including Trip Generation and Data Analysis Study, Trip Generation Rates for Transportation Impact Analyses of Smart Growth Land Use Projects, SF TDM Framework for Growth, and Western District ITE. This data was augmented with the built environment data such as population density, average household size, vehicle ownership, median household income, employment accessibility, and employment entropy.

The authors considered LUC 220 (apartment), 221 (mid-rise multifamily housing), 222 (high-rise apartment), 223 (mid-rise apartment), 230 (residential condominium), 231 (low-rise residential condominium), and 232 (high-rise residential condominium). For Question #1 above, they found significant differences in the built environment (e.g., employment density) between some LUCs. For Question #2, they did not find it necessary to include a land use category beyond built environment variables for residential land uses. They did not find it necessary to include built environment variables beyond development-specific variables (e.g., number of dwelling units or square footage of retail space) for mixed-use sites. For Question #3, the ITE-recommended practice to convert vehicle trip rates into person trips significantly underestimated person trips for retail and service land uses. In other words, there were many non-automobile trips for those land uses.

Gulden et al. (*4*) proposed a mixed-use development (MXD) trip generation model. The MXD model can utilize the trip generations estimated by the ITE Manual and capture internal trips. The data used in their study was from various local, regional, and national sources, such as Wasatch Front Regional Council, the US census, and Salt Lake County assessor data. The independent variables included MXD area, employment in the region, household income, transit stop proximity, etc. The trip estimation process is outlined below:

- Calculate vehicle trips (for each land use) generated by the development using ITE Trip Generation.
- Allocate the estimated vehicle trips by mode (i.e., HBW, HBO, and NHB).
- Predict the number of vehicles per household using the 7D household vehicle ownership model.
- Split the estimated vehicle trips into internal and external vehicle trips using the MXD equations.
- Split the internal and external vehicle trips into trips by mode.
- Split the estimated internal and external vehicle trips into VMT.

The results show a reduction of between 6.3% and 7.9% in external vehicle trips compared with a typical traffic study that relies on summarizing the vehicle trips generated by the individual land use amounts contained in the proposed development.

2.2 Establishing Local Trip Generation Adjustment Factors and Model Development

Several studies were conducted to determine adjustment factors for local areas. These are important because the local areas have unique characteristics that may not be highlighted in the ITE handbook. The methods used in these studies could be important when developing adjustments for Massachusetts trip generation models.

Givechi and Sarkodee (*5*) developed a local trip generation rate model for multiuse (or mixed-use) developments in Kansas. The purpose of the model was to capture internal trips, which are different from pass-by and diverted-link trips. Data was collected from three mixed-use sites in Kansas following guidelines recommended by ITE. The field data collection involved intercept interviews and cordon traffic counts. A manual method was used for cordon traffic counts because they were not allowed to install any traffic counters at those sites. Models for internal capture rates were developed based on the collected data. The trip generation models for a.m. and p.m. peak hours are shown below:

$$AM : T = 1.28X;\ T_i = 0.76T;\ T_o = 0.24T \tag{3}$$
$$PM : T = 2.62X;\ T_i = 0.57T;\ T_o = 0.43T \tag{4}$$

where
$X = 1000$ sq ft of gross floor area;
$T = $ average vehicle trips;
$T_i = $ inbound vehicle trips; and
$T_o = $ outbound vehicle trips.

They compared the observed vehicle trips with those generated by the ITE Trip Generation Manual and found that the ITE Manual overestimated vehicle trips. In their study, the ITE estimates were generated by directly applying the ITE trip generation rates to individual land uses of the development site and adding the results together. Although the authors were able to interview people who were leaving the development, they were not allowed to interview people going from one property to another within the mixed-use development. Therefore, they could not collect data to directly estimate internal trips. They argued that internal trips may be estimated as the difference between the observed and ITE estimated vehicle trips.

Westrom et al. (*6*) developed a method called District DOT MXD+ for predicting the trip generation from the urban MXD. The District DOT MXD+ model consists of a person trip prediction module and a mode choice module. The data was collected by counting the number of persons and vehicles entering and exiting each site and conducting intercept surveys at 62 sites. The author first modeled person trips using linear regression, where the dependent variable is the observed person trips and independent variables include built environment (more specifically density of auto-oriented intersections) and vehicle trips estimated by the ITE Trip Generation Manual. They found that the ITE Trip Generation Manual provided a good basis for estimating person trips. With the predicted person trips, multinomial logistic (MNL) regression was used to predict mode share. The five mode choices considered were auto passenger, transit, bike, walk, and auto driver. The independent

6

variables used in MNL were distance to Metrorail (ft), employment share within 45 minutes by Metrorail, parking provided per service population, neighborhood population density, and employment within 1 mile.

Byrne (*7*) developed local trip generation models for the most common types of development in Vermont. The actual vehicle trip counts and vehicle turning movement counts were collected manually and were compared with those estimated by ITE. It was found that the ITE Manual often overestimated vehicle trips in rural and small urban areas, but the differences in many cases were not very significant. A main challenge brought up in this report was how to increase the local data sample size with a limited budget, suggesting the importance of developing automated data collection technologies.

Wilmot et al. (*8*) performed a study in Louisiana to identify to what extent population density, land use diversity, and road network density add to the accurate estimation of trip generation at strip malls. Manual vehicle counts were used as the ground truth. It was found that including additional contextual variables reduced the vehicle trip rate estimation error by 36% compared to the ITE method. The modification equation is shown below. An additional objective of this study is discussed in the following section.

$$MF_{day} = 30.11 - 32.43x_1 + 33.78x_2 + 9.21x_3 + 115.25x_4 \tag{5}$$

where

MF_{day} = Modification factor for whole day trip estimates from ITE,
X_1 = gross floor area (in units of 1,000 square feet of the site),
X_2 = total worker and residential density (in 1000s of population within 0.5-mile radius),
X_3 = JWR ratio (number of jobs to number of resident workers within 0.5-mile), and
X_4 = local connecting road density (miles of road within 0.5-mile).

2.3 Novel Data Collection Methods and Variable Exploration

Another common category of studies includes novel methods of collecting trip generation data. Furthermore, some studies also introduced and explored variables that may be relevant to developing more accurate trip generation estimates.
Clifton et al. (*9*) quantified the trip generation rates for affordable housing sites in Los Angeles and the San Francisco Bay Area. The data sources were retrieved from the on-site studies for 26 sites, mail-out household travel surveys for 109 sites, and the Caltrans 2012 Household Travel Survey (HTS). The collected information included household information, trip generation, VMT, mode used, person counts, vehicle counts, parking, vehicle ownership, etc. Among them, person and vehicle counts were collected manually on site following the guidelines in the third edition of the ITE Trip Generation Handbook. Based on the on-site data, the authors examined the relationship between person and vehicle trip generation rates and parking ratios, number of bedrooms, number of occupied dwelling units, and

employment density using linear regression. An intercept survey was conducted together with the person and vehicle trip counting to obtain additional information such as mode share, trip purpose, group size, and trip distance.

They further investigated the relationships between home-based vehicle and person trips, VMT, house type, income, place type, household size, automobile ownership, etc., using the 2012 California Household Travel Survey data. Finally, the author tried a mail-out household transportation survey to see whether it could be used to replace the expensive on-site survey. They found that the mail-out survey resulted in a very low response rate and disappointing data quality.

Shi and Zhu (*10*) estimated the commuting trip production and attraction rates for residential land and each subdivided housing type during the morning peak hour using mobile phone signaling data. Trip volume and OD were first derived from mobile phone signaling and cell tower data. Instead of using traditional TAZs, areas covered by neighboring cell towers were combined to create new TAZs for the analysis. Finally, multivariate linear regression was developed to predict the trip production and attraction rates based on land use information at the TAZ level. They pointed out several limitations of their study. They only used cell phone data from one mobile carrier. Travel mode information could not be directly obtained from cell phone data, although this may be estimated. Due to privacy, socioeconomic characteristics of mobile phone owners were not available to their study.

Clifton et al. (*11*) proposed a list of suggestions for improving the data collection for adjusting ITE trip generation rates in different urban contexts to better account for the increased non-automobile trips. Authors discussed in detail strategies related to person counts, vehicle counts, intercept survey of visitors, built environment/urban context, site-level attributes, background traffic, timing and duration of data collection, and new technologies (e.g., tablets) for data collection. These strategies can be useful in developing automated data collection methods in research to ensure that important practical factors are considered.

Kenchappagoudra (*12*) extracted the data for 254 Canadian sites from the 10th ITE Trip Generation Manual data using the Online Traffic Impact Study Software (OTISS Pro). They found that separating US and Canadian data could improve trip generation estimation accuracy. The author also concluded that trip generation estimation accuracy can be further significantly improved by combining the extracted Canadian data provided by ITE with local trip generation data.

Hard et al. (*13*) collected data from workplace and special generator travel surveys in Texas to (1) estimate local trip generation rates for different size categories of MPOs and (2) develop a Texas Trip Generation Manual similar to the ITE Trip Generation Manual. The collected data were grouped into 14 employment categories, and linear regression models were developed to estimate the commercial and noncommercial trips and analyze trip purposes and mode shares. They found the following explanatory variables to be strongly correlated with attraction trip rates: parking availability, MPO area size, traffic analysis zone area type, employment density, and establishment type. It is mentioned in this report that

TxDOT have been using Bluetooth, cellular, and global positioning system (GPS) data to estimate vehicle trips entering, leaving, and crossing a study area. However, no details were provided.

Currans (*14*) developed several methods to adjust ITE trip generation estimates for urban contexts using regional household travel survey (from the 2011 Oregon Household Activity Survey, 2006 Puget Sound Regional Travel Survey, and 2001 National Household Travel Survey Add-On Program) and built environments data (from Census Transportation Planning Package, Census Summary File 3, and more). Three adjustments were proposed to estimate automobile trips as mentioned earlier.

Schneider et al. (*15*) presented a multimodal trip generation data collection method for developments in smart-growth areas that has been used for 30 sites in California. This approach consisted of people entering and exiting counts (door counts) and intercept surveys from a sample of people exiting. The authors pointed out that door counts are important because (1) they cover trips from all modes (counting cars will miss pedestrian and bike trips); (2) they can capture trips for a particular land use in a mixed-use building; and (3) door counts combined with intercept survey can differentiate travel modes. With the proposed data collection methods, they found that the vehicle trip estimates based on the ITE Manual were 2.3 and 2.4 times higher than the observed values during morning and after peak hours, respectively. However, the total person trips estimated by the ITE method and observed were about the same.

Dock et al. (*16*) proposed a method to capture all trips to and from a site and the mode of all travelers. It was tested at mixed-use multifamily residential buildings. They developed a survey–count sheet to collect data in two aspects: (1) site- and area-specific data, including parking space counts, number of doors by type, and more; and (2) count and mode share data, including numbers of people entering and exiting through each door of the site and intercept survey of them. The count and mode share data at each door can be used to further calculate the trip counts by mode for each door and the entire site.

De Gruyter et al. (*17*) collected data for 933 residential developments to understand the relationship between a range of built environment variables and trip generation rates. The Trip Rate Information Computer System database was used to extract the data used in their study. In total, 65 independent variables were included representing location and housing, public transit, parking, etc. Ten linear regression models were developed separately for trip generation rates by mode (person, vehicle, public transport, pedestrian, and bicycle) and time (a.m. peak-hour and p.m. peak-hour).

$$TR_{ijk} = \beta_0 + \beta_1 LD_i + \beta_2 PT_i + \beta_3 DU_i + \beta_4 P_i + \beta_5 SD_i + \beta_6 TP_i + \varepsilon_i \qquad (6)$$

where

TR_{ijk} = trip generation rates for site i by using mode j during time k;

$LD_i, PT_i, DU_i, P_i, SD_i, TP_i$ = built environment variables for site i.

Tian et al. (*18*) used multilevel logit regression models to estimate the probability of internal trips (multilevel binomial logistic regression), walking on internal trips (multilevel binomial

logistic regression), and mode choices for external trips (multilevel multinomial logistic regression), by trip purpose. The data were collected from household travel surveys and other GIS databases for 622 mixed-use developments (MXDs) in 31 regions across the US. The authors identified 16 built environment variables and split them into three levels: (1) individual/household level, such as household size; (2) MXD explanatory variables, such as proportion of developed land; and (3) regional explanatory variables such as population. They found that well-designed MXDs had over 50% of internal trips made by walk mode.

Weinberger et al. (*19*) collected trip generation data via door counts and intercept surveys, and site-level data from the DC Economic Development Partnership database, Google Earth, Zillow.com, and the DC Taxpayer Service Center Real Property Database. The authors collected some key information within a 0.25-mile radius of each site, including office square footage, retail square footage, number of doors by type, bus shelter quality assessment, bicycle rack availability, parking utilization on-street, etc. They also prepared a field guide for future data collection efforts. The observed trip counts were compared with the predicted results of several well-known approaches, including the ITE Trip Generation Manual eighth edition (ITE), URBEMIS Trip Generation Module 2007, the EPA-MXD multiuse analysis method, the California Smart-Growth Trip Generation Model, and three models developed by Portland State University. None of these models were able to accurately predict the observed trips.

Bregman (*20*) collected data from 27 sites to derive trip generation rates for common infill developments in the urban area of California. This study used intercept surveys and people counts to collect trip generation data. The sites selected included mid/high-rise apartments, mid/high-rise resident condominiums/townhouses, general office buildings, specialty retail/shopping centers, and quality restaurants. The collected trip data were compared with those estimated by the ITE Trip Generation Manual. The ultimate goals of this study were to develop a California urban infill land use trip generation database and supplement the ITE Trip Generation data.

In the previous Louisiana study (*8*), another objective was to determine whether the observation of trip generation at a site could be automated. Two methods were used: (1) Wi-Fi and Bluetooth, and (2) video image processing. The Wi-Fi and Bluetooth option accuracy was affected by devices from nearby developments and pass-by vehicles, inconsistent detection radii among the individual Wi-Fi and Bluetooth scanners, and low detection rates (*21*). Extracting entering and exiting traffic from videos was able to produce highly accurate vehicle trip estimates. The trip counts from the video image processing approach were roughly 90% of the observed vehicle trip counts.

Tong et al. (*22*) proposed a comprehensive system (VeTrac) that employed widely deployed traffic cameras as a sensing network to track vehicle movements and reconstruct their trajectories on a large scale. VeTrac first employed a multidimensional similarity (MDS) block to combine identity estimations from license plate texts, vehicle appearance, and mobility causality to calculate the similarity score (MDS score) for every pair of snapshots. A graph convolution network was then adopted that took the MDS scores as input to cluster snapshots into individual vehicle trajectories. Finally, VeTrac generated a maximum

likelihood trajectory for each cluster of snapshots based on their timestamps. Such a model can potentially be used for capturing internal trips for mixed-use developments.

2.4 Comparison of Methodologies Used in Previous Studies

Lee et al. (*23*) identified eight methods for estimating smart growth trip generations. They compared five of them in terms of operation and accuracy, which were:

- ITE multiuse method,
- The EPA/SANDAG MXD multiuse analysis method,
- The NCHRP 8-51 method,
- Method proposed by the Metropolitan Transportation Commission in the San Francisco Bay Area (MTC Survey), and
- URBEMIS 2007

Eleven operational criteria were adopted to compare those methods. The percentage deviation between the observed and estimated trip generation rates was used to measure the accuracy. The traffic counts and built environment data were collected from 22 multiuse sites in California for the model evaluation. They concluded that all five models were more accurate than the ITE rates. However, none of them clearly stood out from the remaining models. The 22 sites were mainly for large multiuse suburban sites and individual urban infill projects. The authors suggested that trip generation data for additional smart growth sites should be collected.

Weinberger et al. (*24*) estimated trips using the ITE Trip Generation Manual and other popular models and compared the results with observed counts and intercept surveys at 16 sites in Washington, DC. In addition to people counts and surveys, the authors used databases including Google Earth, Zillow.com, etc., to collect the built environment and site data, such as number of residential units, parking space count, total square footage, office square footage, retail square footage, major use, and number of doors by type. The models compared in their study include:

- ITE trip generation manual (8th edition)
- URBEMIS trip generation module (TGM) 2007
- The EPA mixed-use development (MXD) multiuse analysis method
- The California smart-growth trip generation model
- Three Clifton and Currans Portland State University models

The authors concluded that the seven models and the ITE Trip Generation Manual all overestimated automobile trips and underestimated person trips. Additional data collection and model development was needed.

Fabregas et al. (*25*) performed a study on internal trips for four MXD sites in Florida. The data collected included: (1) MXD information, such as area; (2) establishment information,

such as land-use units; and (3) user information collected by exit interviews, such as inbound and outbound trips. Six trip generation methods were used including ITE Single Land Use, ITE Internal Trip Capture (ITC), NCHRP Report 684 (NCHRP 684), NCHRP 684 with Proximity, NCHRP+FDOT 2014, and NCHRP+FDOT 2014 with Proximity. They found the NCHRP+FDOT 2014 method to perform the best. The authors also suggested that a trip data repository for MXDs should be developed.

Currans (*26*) concluded that multimodal trip generation data collections were mostly done by counting overall person trips and visitor intercept surveys to calculate mode share and automobile occupancy rate. The total person trips, mode share, automobile occupancy rate, and site-related data were further used to estimate multimodal person trip counts and rates. She also pointed out that multimodal trips have been mainly estimated by adjusting the ITE trip generation rates using equations below.

$$P_{ITE} = \frac{Veh_{ITE}*OCC_{ITE}}{Veh\_\%_{ITE}} \tag{7}$$

$$T_{context,i} = \frac{P_{ITE}*Mode\_\%_{context,i}}{Mode\_OCC_{context,i}} \tag{8}$$

where

Veh_{ITE} = estimated vehicle trips by ITE;
OCC_{ITE} = vehicle occupancy by ITE;
$Veh\_\%_{ITE}$ = vehicle mode share by ITE;
P_{ITE} = estimated total person trips by ITE;
$Mode\_\%_{context,i}$ = share of mode i in a specific context;
$Mode\_OCC_{context,i}$ = occupancy of mode i in a specific context;
$T_{context,i}$ = trips for mode i in a specific context;

In this study, Currans also discussed methods for estimating urban trip generations, including:

- Urban context adjustment;
- Smart growth trip generation adjustment;
- Household travel survey urban context adjustment;
- NCHRP Report 758, National Cooperative Highway Research Program;
- Report 684, NCHRP, an updated version of the ITE Multiuse Method (ITE 2004) not discussed here;
- Environmental Protection Agency (EPA) MXD;
- MXD;
- Report 128, Transit Cooperative Research Program;
- Urban Emissions Model (URBEMIS);
- CalEEMod;
- San Francisco Traffic Impact Guidelines;
- New York City Transportation Guidelines—Section 311 (2014); and
- Washington DC Department of Transportation method.

The above methods are further summarized in Table 1 of (*26*). De Gruyter (*27*) provided a comprehensive review of methods for collecting multimodal trip generation data. He also summarized current and past practices of considering multimodal trip generation, issues associated with estimating and applying multimodal trip generation rates, and key knowledge gaps in this area. The main methods (*27*) for multimodal trip generation data collection include.

- Manual vehicle count
- Automatic vehicle count
- Person count
- Intercept survey
- Household travel survey
- Workplace/school travel survey

Among the key issues identified (*27*), the first two were (1) lack of multimodal trip generation data, and (2) data collection is resource-intensive and complex. The author suggested cross-jurisdictional comparison of multimodal trip generation data; exploration of alternative data collection methods such as Bluetooth, Wi-Fi, and smartphone tracking; standardization of data collection methods; etc.

This page left blank intentionally.

3.0 Development of Massachusetts Trip Generation Models

3.1 Data from Completed Development Projects

The team received trip generation data in PDF report format from MassDOT for 30 developments, covering the following land use categories:

- Casino
- Commercial
- Hotel
- Industrial
- Office
- Research and development
- Recreation
- Residential
- Restaurant
- Retail
- Service

Some of the reports were for mixed-use developments that include multiple land use types. With only 30 reports in total, the data points for each land use type (including mixed-use) are even more limited. Therefore, it is necessary to explore other nontraditional data sources for developing Massachusetts-specific trip generation models.

3.2 Nontraditional Data Sources and Their Advantages

The nontraditional data sources identified in this study include StreetLight, Cuebiq, Replica, and SafeGraph. These data sources all use location-based service (LBS) data. StreetLight allows users to define polygons and specify time windows (e.g., January through April). It then provides average trip generation information over the specified time window for each polygon by hour and day (e.g., Monday, Saturday). Although such data only reflects the visits of mobile device users to the selected polygons, there is very likely to be a strong correlation between the actual visits to those polygons and the StreetLight data. For example, there might exist a consistent ratio between the number of visits based on the LBS data and the true number of visits. If a reliable relationship can be established between polygon/site characteristics and the number of visits based on the LBS data, one can predict the number of visits for a new site. This predicted value can then be adjusted by the above ratio to account for the difference between LBS data and ground truth data.

Although it would be very interesting to conduct a comprehensive comparison of these data sources, this task is out of the scope of this project. Also, the research team only had access to the StreetLight data through MassDOT. Therefore, only the StreetLight data has been considered in this research. Because these data sources are similar to each other, the proposed model development procedure can be readily applied to other sources (like Replica or SafeGraph data) to derive another set of trip generation models.

The main advantages of these nontraditional LBS data sources include:

- They can cover a very large area and avoid the trouble of field data collection. With such data sources, one can focus on identifying development sites. Once the sites are selected, their trip generation data can be obtained very quickly.
- They can provide data to analyze time of day, day of week, and seasonal variations in trip generation rates. Such variations have not been considered in developing the ITE trip generation manual but can potentially be very useful.
- They also provide information about trip origins and destinations, trip lengths, time spent at a development, etc.
- With such data, trip generation estimation can be done using detailed local data such as similar developments within 20 miles, developments of the same size, and developments in areas with similar population densities.

As mentioned above, one of the main benefits of using LBS data sources like StreetLight is that the data can be collected in a very short amount of time. This allows for quick analysis to discover and analyze trip generation trends. An interesting way to showcase this benefit is to look at how trip generation rates changed during the pandemic. Land use code 221 (mid-rise multifamily housing) is used to demonstrate how certain residential areas reacted to a rise in COVID cases. Figure 3.1 below shows the results for Zone ID 381, which is in Malden, MA.

In Figure 3.1, new COVID cases are shown in red, and the total daily trips generated and attracted by Zone 381 are shown in blue. There was a clear drop in trips when the pandemic first started in March 2020: trips went from ~2800 down to ~2400 per day from March to May. This is seen again in December, 2020 when COVID cases were at an all-time high and the total daily trips dropped drastically. But after the spike in December 2020, trips returned to normal and even exceeded the previous normal trips. One possible explanation for this could be the administration of vaccines. The public began receiving vaccinations and thus people were more comfortable being out in public spaces.

Many other zones in land use 221 followed similar trends as in Figure 3.1. However, some outliers are also identified. One example is Zone ID 774, located in Lynnfield, MA. Its results are in Figure 3.2, which shows that there was no large initial drop in trips in March 2020. It is difficult to determine why this was the case without more detailed and comprehensive data (e.g., new buildings added). However, from June to December 2020, there was a consistent drop in trips, and from December 2020 onward, the number of trips recovered. The relationship between COVID cases and total daily trips during this period (June 2020 to June 2021) appears to be reasonable.
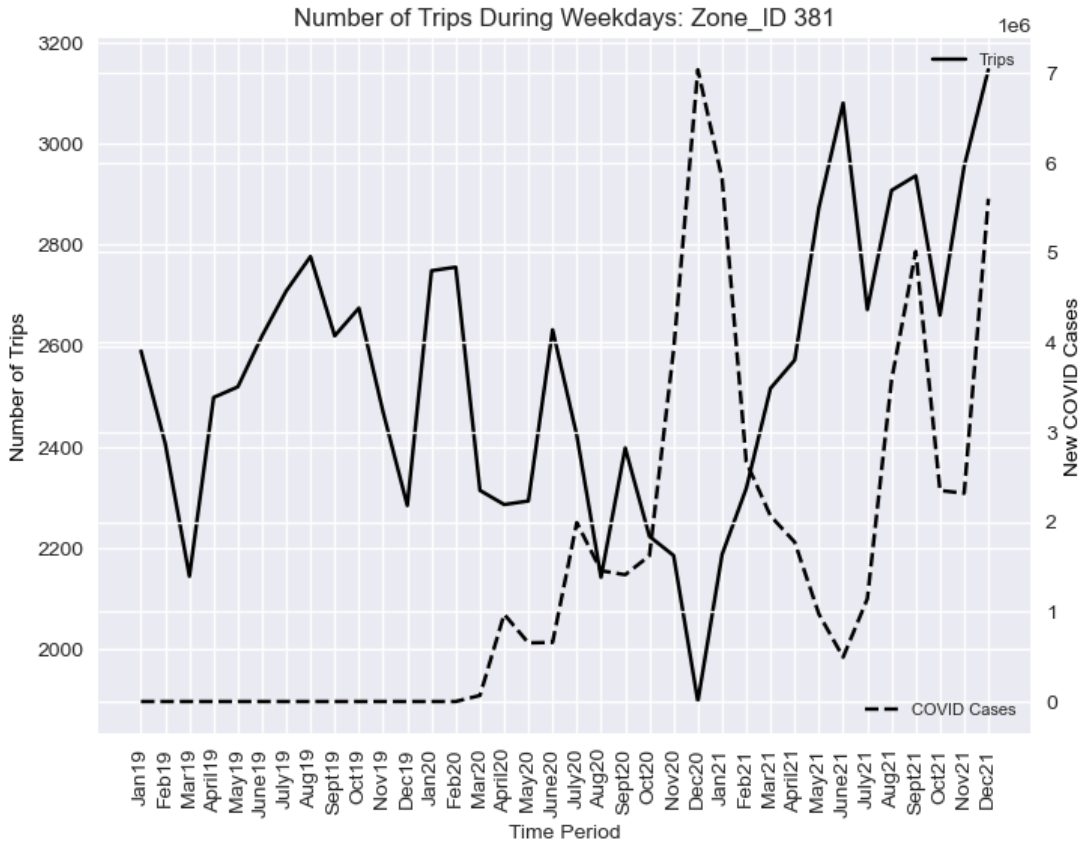
Figure 3.1: Trip generation results for Zone 381 (mid-rise multifamily housing)

The same trip generation analysis is conducted for another site that belongs to land use 850 (supermarkets). This site is in Lynn, MA, and its Zone ID is 89. The trip generation analysis results are shown in Figure 3.3. In this figure, there is a sharp increase in trips during March 2020 and a drastic decrease in April 2020. One possible explanation for this is that many people went to stock up on food and essential supplies in March 2020. They had their groceries settled for the month and minimized their grocery shopping trips. But once the initial fear of COVID dissipated, more and more people were comfortable going out for grocery shopping. Another observation is that supermarkets (land use code 850) did not follow the same trend as apartments (land use code 221) did during the COVID-19 case spike in December 2020—trips remained relatively constant rather than decreasing sharply.
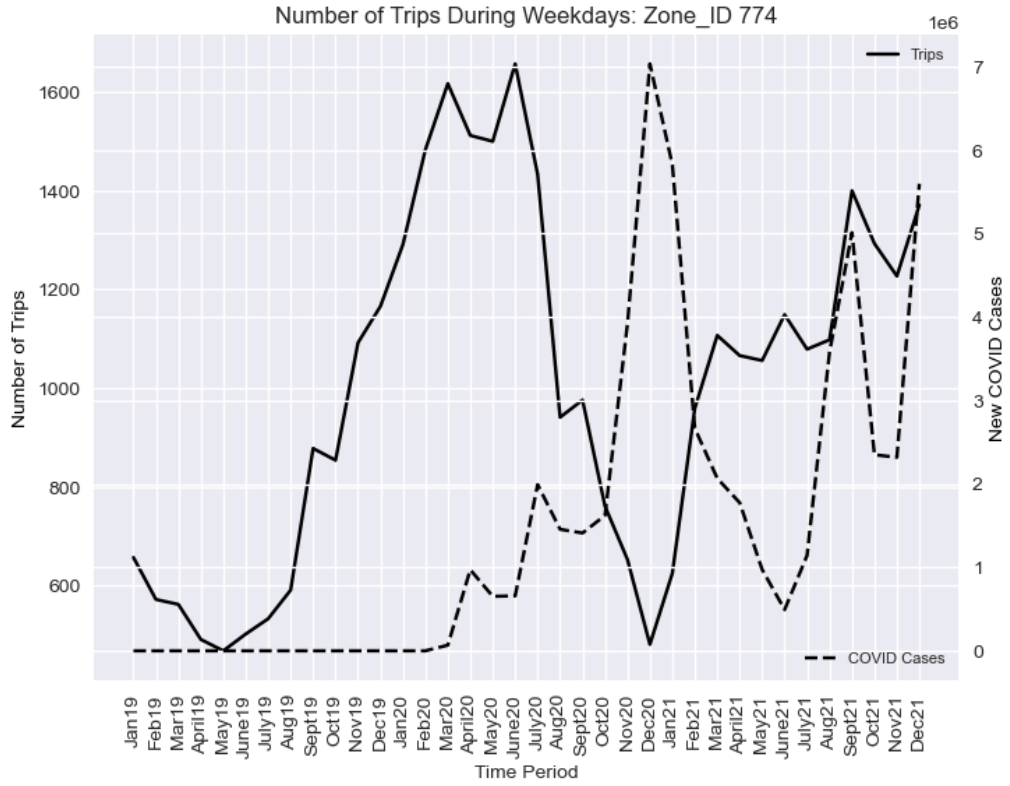
Figure 3.2: Trip generation results for Zone 774 (mid-rise multifamily housing)
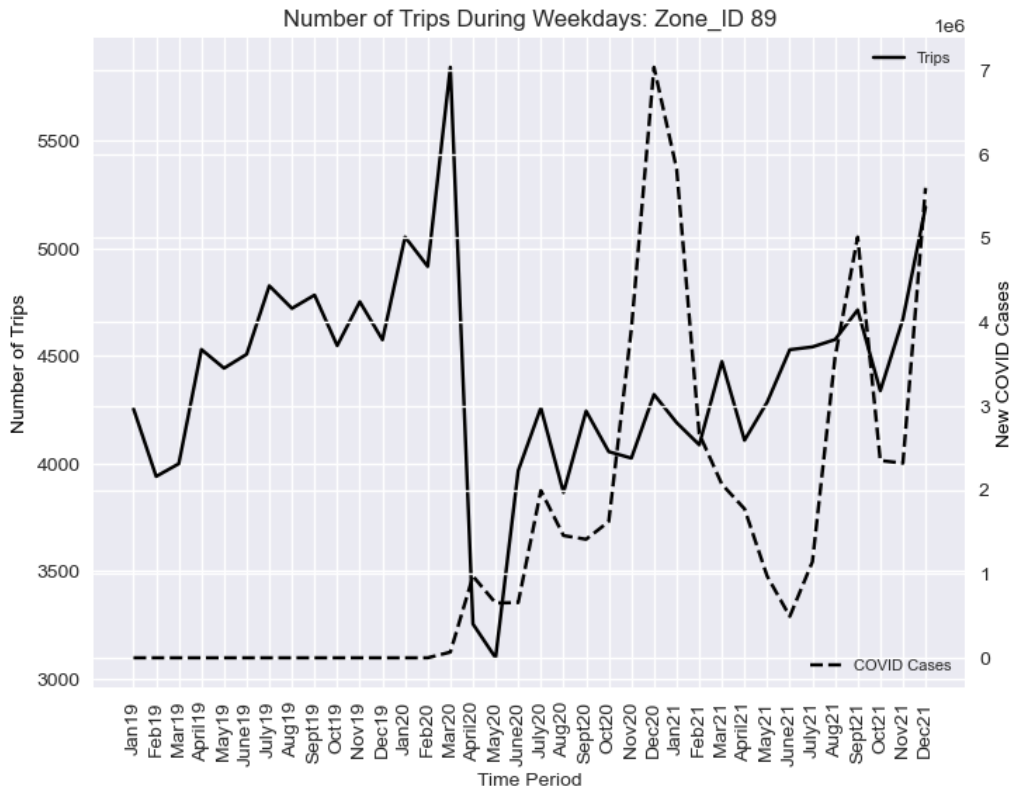
Figure 3.3: Trip generation results for Zone 89 (supermarkets)

18

One final example to show the advantages of using LBS data is for land use code 150 (high-cube fulfillment center warehouse such as Amazon warehouses). Warehouses are unlike the previous examples, and most people generally do not visit warehouses. Therefore, warehouses have different trip generation patterns. Figure 3.4 shows the trip generation results derived from StreetLight data for Zone ID 323, which is an Amazon fulfillment center in Milford, MA.

This warehouse is where Amazon temporarily stores packages that are sorted and later delivered to customers. In March 2020, there was a drastic increase in trips to this warehouse, which was probably because many people switched to online shopping due to COVID-19. As a comparison, Figure 3.5 shows the trips observed in Zone 492, which belongs to land use 820 (shopping centers). This figure essentially suggests an inverse relationship between trips generated by warehouses and shopping centers, particularly for March through May in 2020. Although this relationship is expected and reasonable, it is still very interesting to see data that supports such conjectures.
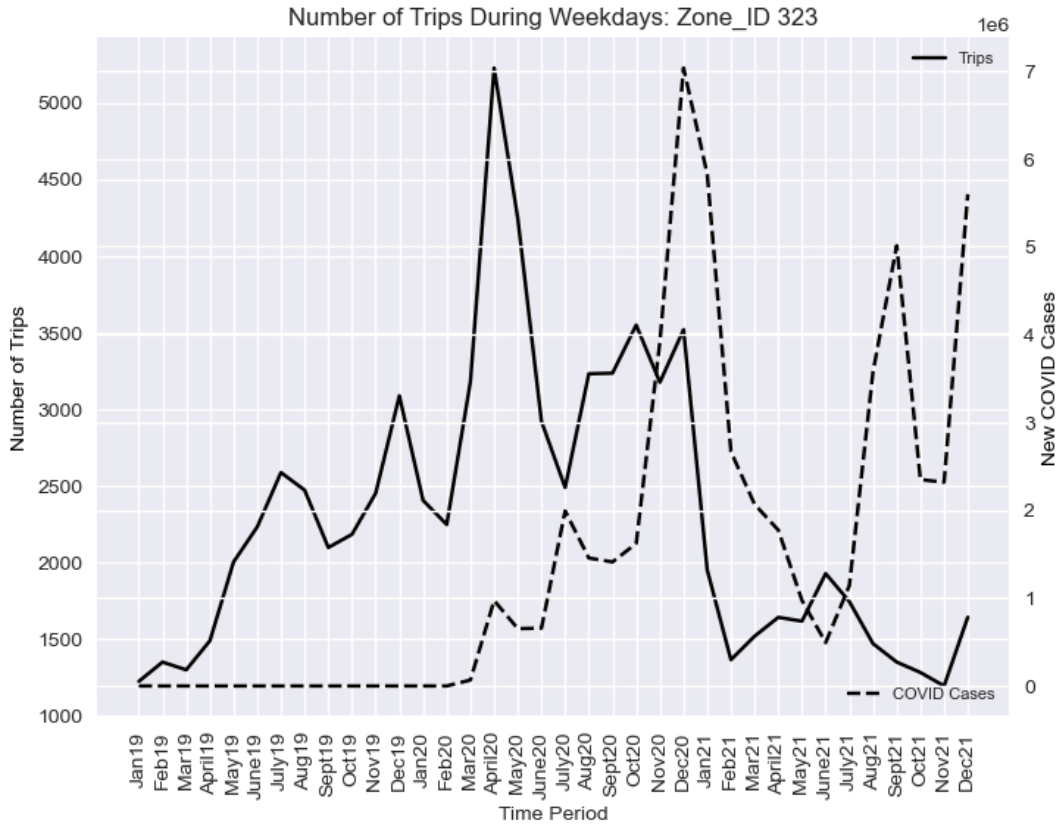
Figure 3.4: Trip generation results for Zone 323 (high-cube fulfillment center warehouse)

From these examples, the advantages of using LBS data for modeling trip generation are clear. LBS data allows analysts to quickly gather trip generation data for many development projects and saves a tremendous amount of the time typically required by traditional trip generation data collection methods. More importantly, the temporal changes in trip generation can be captured and the developed trip generation models can be easily updated as needed to reflect the impacts of major events such as COVID.
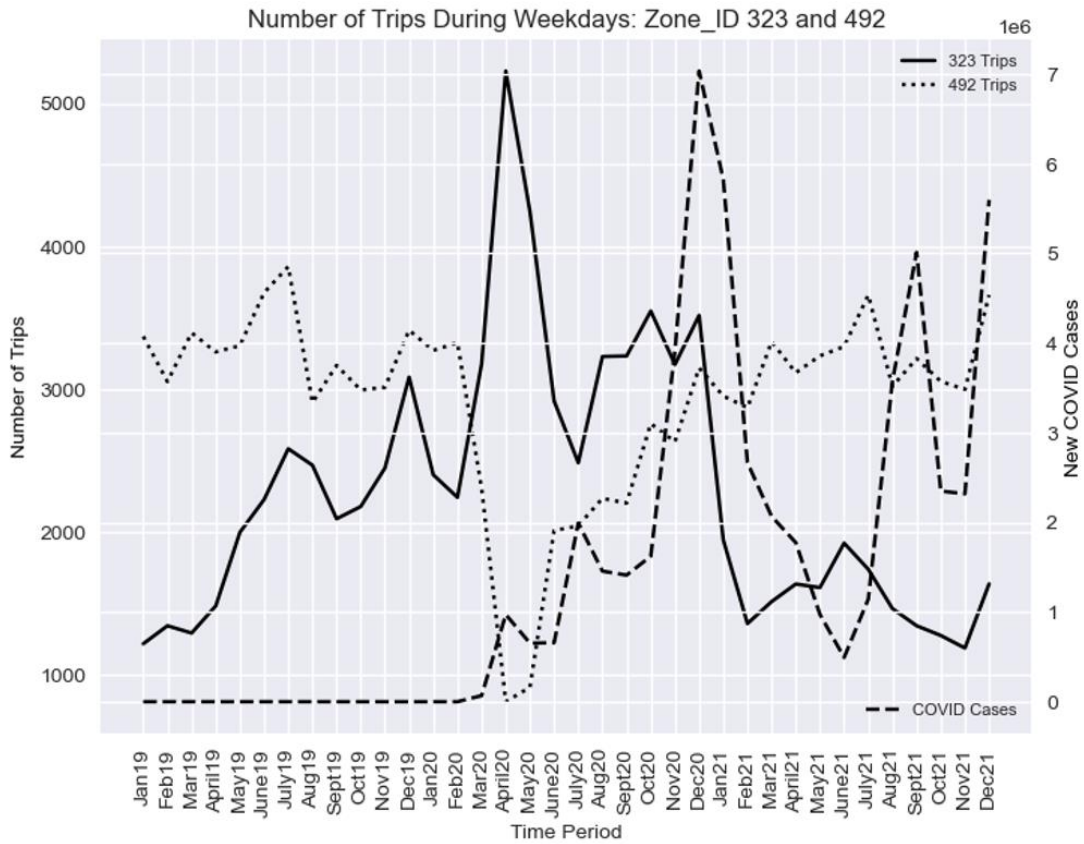
Figure 3.5: Trip generation results for Zones 323 (high-cube fulfillment center warehouse) and 492 (shopping centers)

3.3 Identified High-Priority Land Use Categories

Based on feedback from MassDOT, the following 10 high-priority land use categories are identified as shown in Table 3.1. Given these categories, sites/developments were randomly selected for developing Massachusetts-specific trip generation models. A polygon Shapefile was created for these sites and fed into StreetLight to obtain the numbers of trips for each site. The locations of these selected sites/polygons are shown in Figure 3.6, which suggests that these sites cover different parts of Massachusetts instead of being concentrated in selected areas.

Table 3.1: Ten high-priority land use categories

ID	ITE Code	# of Developments	Land Use Type
1	140	75	Industrial—manufacturing
2	150	52	Warehouse
3	155	19	High-cube fulfillment center warehouse (e.g., Amazon)
4	221	49	Mid-rise multifamily housing
5	710	72	General office building
6	760	41	Research and development
7	813	38	Freestanding discount superstore
7	815	12	Freestanding discount store
8	820	85	Shopping centers
9	850	84	Supermarket
10	960	73	Convenience market/gas station
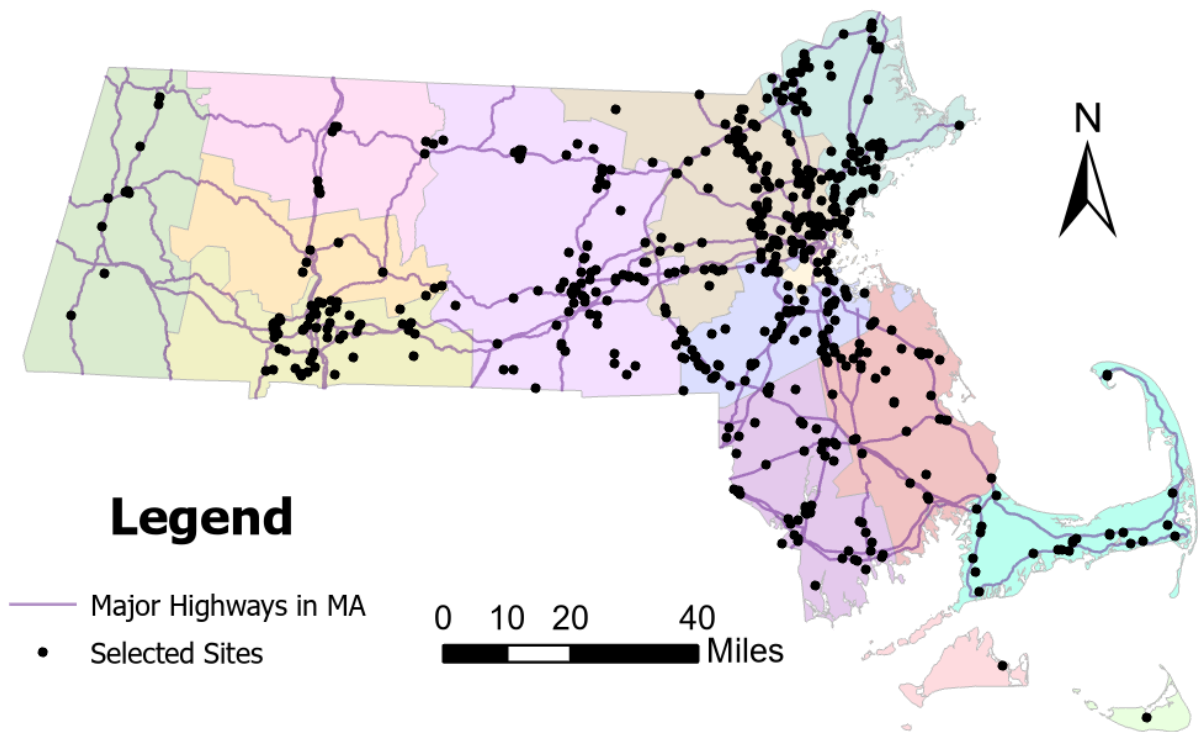
Figure 3.6: Selected sites/polygons for this study

3.4 StreetLight Data

Note that StreetLight data was selected for this study because MassDOT provided free StreetLight access to the team. Therefore, StreetLight data was used as an example to demonstrate the potential of LBS data for trip generation modeling. It does not necessarily mean that StreetLight is the best choice for this research. Other data sources such as SafeGraph and Replica can also be used to develop trip generation models by adjusting the procedure proposed in this study.

Table 3.2 is a sample StreetLight dataset showing the numbers of trips starting or ending in a user-defined polygon. Each polygon is for a specific development. To generate the results in Table 3.2, analysts need to provide the following information:

- A polygon shapefile. Each polygon in this file is for a specific development. It is important to include everything (e.g., parking lot, building) pertinent to a development in the polygon. How to draw polygons is detailed in Chapter 4.4, Identification of Polygons.
- The time windows for pulling zonal trip activity data from StreetLight. Users can specify multiple time windows in one analysis. However, the trip activities in those time windows will be aggregated and the average results will be provided for all chosen time windows. If a user wants to analyze seasonal trip generation variations, separate analyses should be conducted instead of one covering multiple time windows. For example, one analysis for spring and one for summer months.

The research team also conducted a preliminary study to validate the StreetLight trip data for trip generation analysis purposes. The results are presented in Table 3.3. Among the seven sites, only two sites had ITE estimates. The limited samples show that overall, StreetLight data matched the observed trip counts well. The StreetLight trip counts in Table 3.3 were extracted from different time periods in 2018 to match the corresponding observed data in terms of month to mitigate the impacts of seasonal variations. For example, the StreetLight data for the Southfield site was extracted using the May 2018 time window, and the corresponding ground truth data was observed in May (the same month) 2016.

StreetLight data is based on data collected from mobile devices. In general, data in more recent years is more comprehensive and accurate than data collected many years ago. However, data in 2020, 2021, and 2022 may be significantly affected by the COVID-19 pandemic. This problem can be viewed in a positive way. The skewed data in 2020 through 2022 can be used to analyze how the pandemic has affected trip generations for different land uses. For this type of longitudinal analysis, it is important to make sure that the businesses at a particular site have not changed over time. It is common that some businesses closed temporarily or permanently due to the pandemic, and this would certainly affect the corresponding site trip generation.

3.5 Developed Models

Selecting sites for model development is still a time-consuming process, although this is a much quicker process with LBS data than by conducting field data collection. The team first identified close to 800 sites. These sites went through several rounds of quality checks, which resulted in 600 sites as shown in Table 3.1 and Figure 3.6. Zonal trip activity data was pulled from StreetLight during the following periods.

- Dataset 1 (before COVID): 11/15/2018-1/15/2019, 2/1/2019-4/1/2019, 6/1/2019-8/1/2019, and 9/1/2019-11/1/2019 (it was initially thought that the system would generate four sets of trip generation data with one for each period. However, it generated one output averaging over the four periods).
- Dataset 2 (during COVID): 9/1/2021-11/1/2021
- Dataset 3 (toward the end of COVID): 1/1/2022-4/1/2022

Three sets of models have been developed based on the above datasets and the results are detailed in the following three subsections. The raw data from StreetLight must be reformatted for model development purposes. This step is described in Chapter 4.2, StreetLight Data Post-Processing.

Table 3.2: Sample StreetLight zonal vehicular trip activity data

Intersection Type	Zone ID	Zone Is Pass-Through	Zone Direction (degrees)	Zone Is Bi-Direction	Day Type	Day Part	Average Daily Zone Traffic (StL Volume)
Trip Start	7	no	N/A	no	0: All Days (M-Su)	00: All Day (12am-12am)	519
Trip Start	7	no	N/A	no	0: All Days (M-Su)	01: 12am (12am-1am)	2
Trip Start	7	no	N/A	no	0: All Days (M-Su)	02: 1am (1am-2am)	2
Trip Start	7	no	N/A	no	1: Monday (M-M)	00: All Day (12am-12am)	612
Trip Start	7	no	N/A	no	1: Monday (M-M)	01: 12am (12am-1am)	16
Trip Start	7	no	N/A	no	2: Tuesday (Tu-Tu)	00: All Day (12am-12am)	537
Trip Start	7	no	N/A	no	3: Wednesday (W-W)	00: All Day (12am-12am)	520
Trip Start	7	no	N/A	no	4: Thursday (Th-Th)	00: All Day (12am-12am)	459
Trip Start	7	no	N/A	no	4: Thursday (Th-Th)	02: 1am (1am-2am)	17
Trip Start	7	no	N/A	no	5: Friday (F-F)	00: All Day (12am-12am)	498
Trip Start	7	no	N/A	no	6: Saturday (Sa-Sa)	00: All Day (12am-12am)	498
Trip Start	7	no	N/A	no	7: Sunday (Su-Su)	00: All Day (12am-12am)	496
Trip End	7	no	N/A	no	0: All Days (M-Su)	00: All Day (12am-12am)	461
Trip End	7	no	N/A	no	0: All Days (M-Su)	03: 2am (2am-3am)	2
Trip End	7	no	N/A	no	1: Monday (M-M)	00: All Day (12am-12am)	500
Trip End	7	no	N/A	no	2: Tuesday (Tu-Tu)	00: All Day (12am-12am)	408
Trip End	7	no	N/A	no	2: Tuesday (Tu-Tu)	03: 2am (2am-3am)	15
Trip End	7	no	N/A	no	3: Wednesday (W-W)	00: All Day (12am-12am)	442
Trip End	7	no	N/A	no	4: Thursday (Th-Th)	00: All Day (12am-12am)	408
Trip End	7	no	N/A	no	5: Friday (F-F)	00: All Day (12am-12am)	371
Trip End	7	no	N/A	no	6: Saturday (Sa-Sa)	00: All Day (12am-12am)	568
Trip End	7	no	N/A	no	7: Sunday (Su-Su)	00: All Day (12am-12am)	533

Table 3.3: Validation of StreetLight data for trip generation analysis purposes

Site Name	Address	ITE Land Use Code	Obs[a] AM Peak Exit	Obs[a] AM Peak Enter	SL[b] AM Peak Exit	SL[b] AM Peak Enter	ITE AM Peak Exit	ITE AM Peak Enter	Obs[a] PM Peak Exit	Obs[a] PM Peak Enter	SL[b] PM Peak Exit	SL[b] PM Peak Enter	ITE PM Peak Exit	ITE PM Peak Enter	Date Data Was Observed	Data Collection Time
Southfield	200 Trotter Road, Weymouth	220	158	96	117	63	--	--	150	155	139	146	--	--	May 2016	Weekday AM Peak: 7-9am Weekday PM Peak: 4-6pm
Stoneham Crossing	225 Fallon Road, Stoneham	221	105	13	73	15	163	163	36	84	17	77	205	205	Nov/Dec 2018	Weekday AM Peak: 7-9am Weekday PM Peak: 4-6pm
Upland	1 Upland Woods Circle, Norwood	220	81	23	118	21	106	26	35	69	36	66	57	105	10/18/2018	Weekday AM Peak: 7-9am Weekday PM Peak: 4-6pm
Walmart	15 Tobey Road, Wareham	813	490	482	469	404	--	--	338	314	323	294	--	--	May and July 2018	Sat. Midday: 1-2pm Weekday PM Peak: 4-5pm
Walmart	1415 Curran Memorial Highway, North Adams	813	412	398	467	447	--	--	322	325	383	407	--	--	Sept and Oct 2016	Sat. Midday: 11am-2pm Weekday PM Peak: 4-6pm
Walmart	100 Valley Parkway, Northbridge	813	431	461	529	514	--	--	325	335	453	419	--	--	March in 2012	Sat. Midday: 11am-2pm Weekday PM Peak: 4-6pm
Wegmans	53 Third Avenue, Burlington	850	310	311	334	366	616	797	274	267	280	299	658	703	June in 2015	Sat. Midday: 11am-1pm Weekday PM Peak: 4-6pm

a: observed
b: StreetLight

The following variables have been considered in model fitting:

- Building square footage (sq ft): This was measured based on the tax parcel data and calibrated using Google satellite images.
- Employment opportunity density (workers/square mile): Based on the center of a site/polygon, the corresponding traffic analysis zone (TAZ) was identified. The TAZ's worker density was used as the employment opportunity density, which was calculated based on the 2012-2016 Census Transportation Planning Products Program (CTPP) data. More specifically, the A202100_Total_Workers table is used. Note that one polygon may overlap with multiple TAZs. Only the TAZ where the center of the polygon is located was considered.
- Population density (people/square mile): This was done using the same approach and data source as for calculating the employment opportunity density. The A101100_Total_Population table from the 2012-2016 CTPP database was used.

The model fitting results suggest that including employment opportunity density and population density does not help to improve model fitting. These two variables were statistically insignificant (at the 0.05 level) in almost all cases. Therefore, only the model fitting results based on the building square footage are reported here to keep this report concise. The results of other models considering all three variables are available upon request. In addition, log linear regression has been considered in this research. This approach only slightly improved the model fitting for land use category 155 and generated worse results for other land use categories. Therefore, the corresponding log linear regression results are not included here.

Table 3.4, Table 3.5, and Table 3.6 summarize the model fitting results. It can be seen from these tables that:

- As mentioned earlier in this report, the research team has checked all initially identified polygons (sites) and removed some of them to ensure good data quality. The cleaned dataset has 600 sites. Based on the cleaned dataset, the team fitted new models presented in Table 3.4. Table 3.5. and Table 3.6.. Overall, the new models generated higher R^2 values than the previous ones. However, in a few cases, they led to lower R^2 values than previous models. The team also found that the R^2 for 140 (weekend), 155, 710, 760, and 960 were significantly worse than that for other categories.
- For 140 (industrial—manufacturing), the R^2 values overall were low, and the research team may need to further divide them into subcategories of manufacturing (e.g., heavy machinery vs. high-tech).
- For 155 (high-cube fulfillment center warehouse), the team may need to contact each fulfillment center and see if they can be grouped into this category, which may improve the R^2 values.
- For 710 (general office building) and 760 (research and development), the models fitted based on 2018 (pre-COVID) data had decent R^2 (0.68 and 0.47 for weekdays), but the models fitted based on 9/2021 and 1/2022 data had poor R^2. This is probably because many people were working from home during the COVID pandemic.

- For 960 (convenience market/gas station), trip generations probably depend more on factors other than building square footage. The research team will likely need to consider additional variables, such as proximity to major roads, proximity to major businesses, and number of pumps, to improve the fitting.

Table 3.4: Model fitting R^2 results based on 2018 all-year data

Land Use Code	Land Use Type	Sample Size	Weekday	Weekday AM	Weekday PM	Weekend	Weekend AM	Weekend PM
140	Industrial—manufacturing	75	0.5	0.4	0.4	0.32	0.33	0.23
150	Warehouse	51	0.43	0.29	0.38	0.28	0.34	0.33
155	High-cube fulfillment center warehouse	19	0.06	0.1	0.2	0.09	0.18	0.23
221	Mid-rise multifamily housing	49	0.9	0.92	0.87	0.89	0.87	0.87
710	General office building	72	0.68	0.58	0.56	0.16	0.07	0.07
760	Research and development	41	0.47	0.45	0.55	0.04	0.14	0.02
820	Shopping centers	85	0.71	0.66	0.71	0.74	0.58	0.81
850	Supermarket	84	0.5	0.43	0.48	0.52	0.47	0.49
960	Convenience market/gas station	73	0.31	0.3	0.29	0.28	0.25	0.23
813 Or 815	Free standing discount store and superstore	50	0.64	0.63	0.65	0.66	0.68	0.68

Table 3.5: Model fitting R^2 results based on September 2021 data

Land Use Code	Land Use Type	Sample Size	Weekday	Weekday AM	Weekday PM	Weekend	Weekend AM	Weekend PM
140	Industrial—manufacturing	75	0.5	0.24	0.25	0.18	0.02	0.06
150	Warehouse	51	0.65	0.52	0.6	0.48	0.56	0.42
155	High-cube fulfillment center warehouse	19	0.14	0.05	0.22	0.33	0.08	0.27
221	Mid-rise multifamily housing	49	0.74	0.87	0.71	0.71	0.62	0.74
710	General office building	72	0.34	0.32	0.3	0.15	0.22	0.13
760	Research and development	41	0.25	0.24	0.15	0.11	0.17	0.11
820	Shopping centers	85	0.7	0.57	0.77	0.73	0.52	0.8
850	Supermarket	84	0.49	0.45	0.42	0.48	0.43	0.44
960	Convenience market/gas station	73	0.48	0.33	0.51	0.4	0.24	0.48
813 Or 815	Free standing discount store and superstore	50	0.7	0.64	0.72	0.72	0.67	0.76

Table 3.6: Model fitting R^2 results based on January 2022 data

Land Use Code	Land Use Type	Sample Size	Weekday	Weekday AM	Weekday PM	Weekend	Weekend AM	Weekend PM
140	Industrial—manufacturing	75	0.42	0.19	0.12	0.24	0.1	0.14
150	Warehouse	51	0.69	0.55	0.5	0.41	0.56	0.32
155	High-cube fulfillment center warehouse	19	0.12	0.02	0.17	0.14	0.01	0.13
221	Mid-rise multifamily housing	49	0.86	0.86	0.84	0.86	0.83	0.84
710	General office building	72	0.35	0.29	0.21	0.16	0.13	0.1
760	Research and development	41	0.29	0.28	0.23	0.16	0.09	0.11
820	Shopping centers	85	0.68	0.53	0.72	0.74	0.5	0.77
850	Supermarket	84	0.44	0.37	0.4	0.46	0.36	0.44
960	Convenience market/gas station	73	0.45	0.35	0.44	0.35	0.21	0.34
813 Or 815	Free standing discount store and superstore	50	0.68	0.61	0.72	0.71	0.69	0.71

Table 3.7 below is an example model fitting result table for ITE land use code 221 that shows the intercept and slope values (as well as p-values). This table can be used to calculate trip generation estimates given the day of week, time of day, and square footage of a site. Coefficients with a p-value greater than 0.1 are considered insignificant for trip prediction and are excluded from the final models. The remaining model fitting result tables are provided in Appendix D. Equation Tables for Dataset 3 (January 2022).

Table 3.7: Results for ITE Code=140 and Dataset 3.

Model Name	INTERCEPT (β_0) Coefficient	INTERCEPT (β_0) p-Value	SLOPE (β_1) Coefficient	SLOPE (β_1) p-Value
Weekday	54.27	0.04	0.77	0.00
Weekday AM	13.20	0.01	0.08	0.00
Weekday PM	14.51	0.00	0.05	0.00
Weekend	0.00*	0.12	0.14	0.00
Weekend AM	2.72	0.01	0.01	0.01
Weekend PM	1.85	0.09	0.02	0.00

* - set to zero due to being insignificant, p-value > 0.1.

Using the above table, estimated trips for land use 140 can be calculated as follows:

$$\text{Estimated Trips} = \beta_0 + \beta_1 X \tag{9}$$

where

X = site area in 1,000 sq ft, and
β_i = coefficients ($i = 0,1$)

4.0 Trip Generation Model Guidance

The overall steps for model development are as follows and are also illustrated in Figure 4.1:

1. Identify land use categories to be modeled. For each category, find about 50 developments. Create a Shapefile and generate a polygon for each development. Polygons for different land use categories can be included in the same Shapefile. In other words, there is no need to create multiple Shapefiles for different land use categories.
2. Upload the Shapefile to StreetLight. Specify the time frame for extracting trip generation data in StreetLight.
3. Download the StreetLight polygon-level trip generation results and process them using the StreetLight processing script (See Appendix A) into the format needed for the model fitting script (See Appendix B).
4. Collect additional variables for model fitting, which could include attribute information for the selected developments (i.e., building square footage in this research), population density, and employment data. The building square footage data can be obtained from the MassGIS Property Tax Parcels layer (*28*). The population density and employment data can be obtained from the CTPP database (*29*). In this study, the population density and employment attributes were found to be unhelpful. Therefore, only the building square footage data was used.
5. Use the trip generation data from Step #3 as the outputs and data from Step #4 above as the inputs to establish trip generation models (see Figure 4.1 below). These models can be created using either statistical methods or machine learning methods depending on how many data points (i.e., data records) are available. Given that there were less than 100 data points for each land use category, linear regression models were fitted (see Appendix D) for all land use categories.

Once models are developed, they can be applied to new developments to predict their trip generations. For example, with a given development, one only needs to collect the input data (Step #4) and then feed them into the model(s) developed. If users want to further evaluate the accuracy of the models developed, it can be done by following the steps below, which are also outlined in Figure 4.2:

i. Repeat Steps #1-4 above to collect model input data for additional developments.
ii. Apply the above developed models to predict trip generations for the newly selected developments.
iii. Compare the predicted trip generation rates with those provided by StreetLight (see Step #1 above).
iv. If needed, use the developed AI-based data collection tool to collect ground truth data at those selected developments and compare the predicted trip generation rates with those data collected from the field.
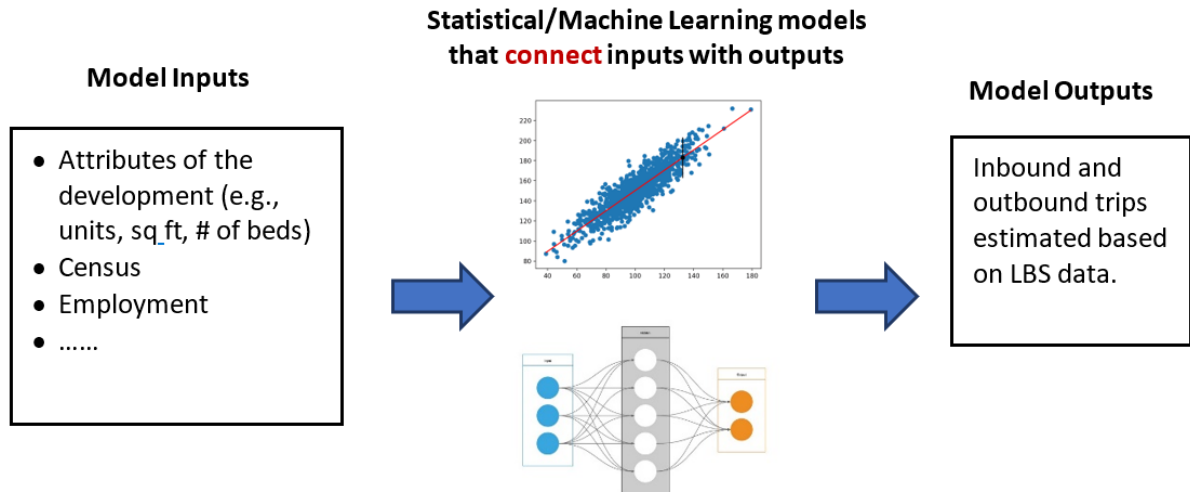
Figure 4.1: Trip generation models based on LBS data
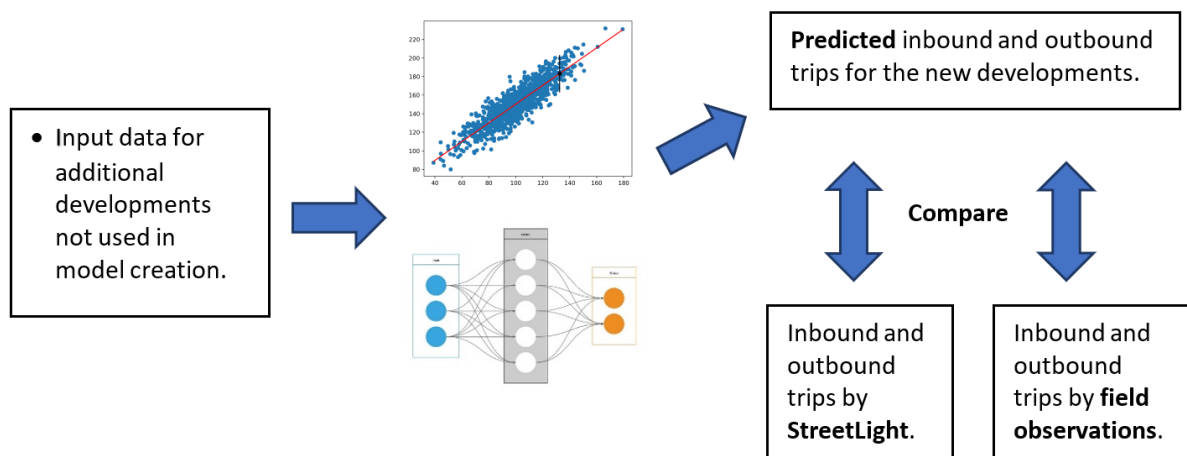
Figure 4.2: Evaluation of the developed trip generation models

A detailed explanation of how to run the model fitting script within ArcGIS Pro is provided in Chapter 4.5, Detailed Steps for Running Python Scripts in ArcGIS Pro.

4.1 StreetLight Data Preparation

To extract trip data from the StreetLight database, polygons for each development must be properly defined. The attributes of the Shapefile should look like the one in Figure 4.3. Among them,

- **FID** and **Shape** are attributes reserved by the Shapefile system.
- **id** is an attribute used to uniquely identify developments.

- **Name** is to store the name of a development; for example, Walmart in Tewksbury, MA.
- **is_pass** should be set to 0 for this research, which means trips that start or end in the polygon will all be considered.
- **direction** and **is_bidi** should both be set to 0 for this application.

FID	Shape	name	direction	is_pass	is_bidi	id
0	Polygon	Walmart	0	0	0	1
1	Polygon	Walmart	0	0	0	2
2	Polygon	Walmart	0	0	0	3
3	Polygon	Walmart	0	0	0	4
4	Polygon	Walmart	0	0	0	5
5	Polygon	Walmart	0	0	0	6
6	Polygon	Walmart	0	0	0	7
7	Polygon	Walmart	0	0	0	8
8	Polygon	Walmart	0	0	0	9
9	Polygon	Target	0	0	0	10
10	Polygon	Walmart	0	0	0	11

Figure 4.3: Attributes of the Shapefile for defining zones used by StreetLight

This Shapefile can be created online using the StreetLight platform by choosing "Zones" on the main page, then "Create New." Another option is to utilize commercial GIS software following the data structure shown in Figure 4.3 using the World Geodetic System (WGS) 1984 Geographic Coordinate System.

After completing the Shapefile, users need to go to "Analyses" from the main page and choose "Create New Analysis," then "Trips to or from Pre-set Geography." This will take users to the web interface shown in Figure 4.4 through Figure 4.7.

For basic information (Figure 4.4), the "Name" field is required to uniquely identify the analysis to be performed. "Mode of Travel" should be set to "All Vehicles LBS+." Although StreetLight also provides trip generation information for truck, bicycle, and pedestrian, the results seem unreliable given the relatively small shares of such travel modes. Also, the data from completed development projects do not cover such modes. Therefore, the analyses for this research so far have considered only the "All Vehicles LSB+" mode. For "Output Type," there are the following four options based on the StreetLight website (*30*). For this research, the "StreetLight All Vehicles Volume (Vehicle Trips)" option was chosen. The remaining settings on this basic information page are quite intuitive and can be easily configured.

- **StreetLight All Vehicles Volume (Vehicle Trips):** Provides an estimated number of vehicle trips that are comparable to real-world count data. StreetLight Volume is based on an algorithm trained with real-world data and seasonal factors. This is the best count information for seasonal, monthly, hourly, and specific date traffic values and is the best output type when doing time trends or comparisons across time.
- **StreetLight Index (Device Trips):** Provides comparative, normalized values based on the sample trip counts, but is not the estimated real-world counts.

- **Single Factor Calibrated Index Using StreetLight AADT (Vehicle Trips):** Provides scaled estimated counts based on a process to create a single normalization factor that is an average of the StreetLight Sample Trip Counts compared to StreetLight AADT estimates.
- **Single Factor Calibrated Index Using User Counts (Vehicle Trips):** Provides scaled estimated counts based on a process to create a single normalization factor that is an average of the StreetLight Sample Trip Count compared to the user's count data. This factor is then applied to the StreetLight index, which scales it to estimated counts.

Figure 4.4: Basic information for creating a new StreetLight zone analysis

Figure 4.5: Time periods information for creating a new StreetLight zone analysis

For time periods information (Figure 4.5), users can specify "Data Periods," "Day Types," and Day Parts." For this research's modeling purposes, "Day Types" should be set to "Daily" and "Day Parts" should be set to "Hourly." For "Data Periods," if multiple time periods are selected, the trip rates in them will be averaged. If users want to analyze the trip generation rates for different time periods (e.g., spring, summer, winter) separately, different analyses should be created instead of choosing multiple time periods in one analysis.

For zones information in Figure 4.6, users have three options to create the zones for trip generation analysis. The "Existing Zones" option allows users to choose a zone set (i.e., a Shapefile) previously uploaded using the "Zones" → "Create New" method. The "Draw" option allows users to draw zones on the screen. The "Upload" option works the same as the "Zones" → "Create New" method and allows users to upload a Shapefile consisting of zones.
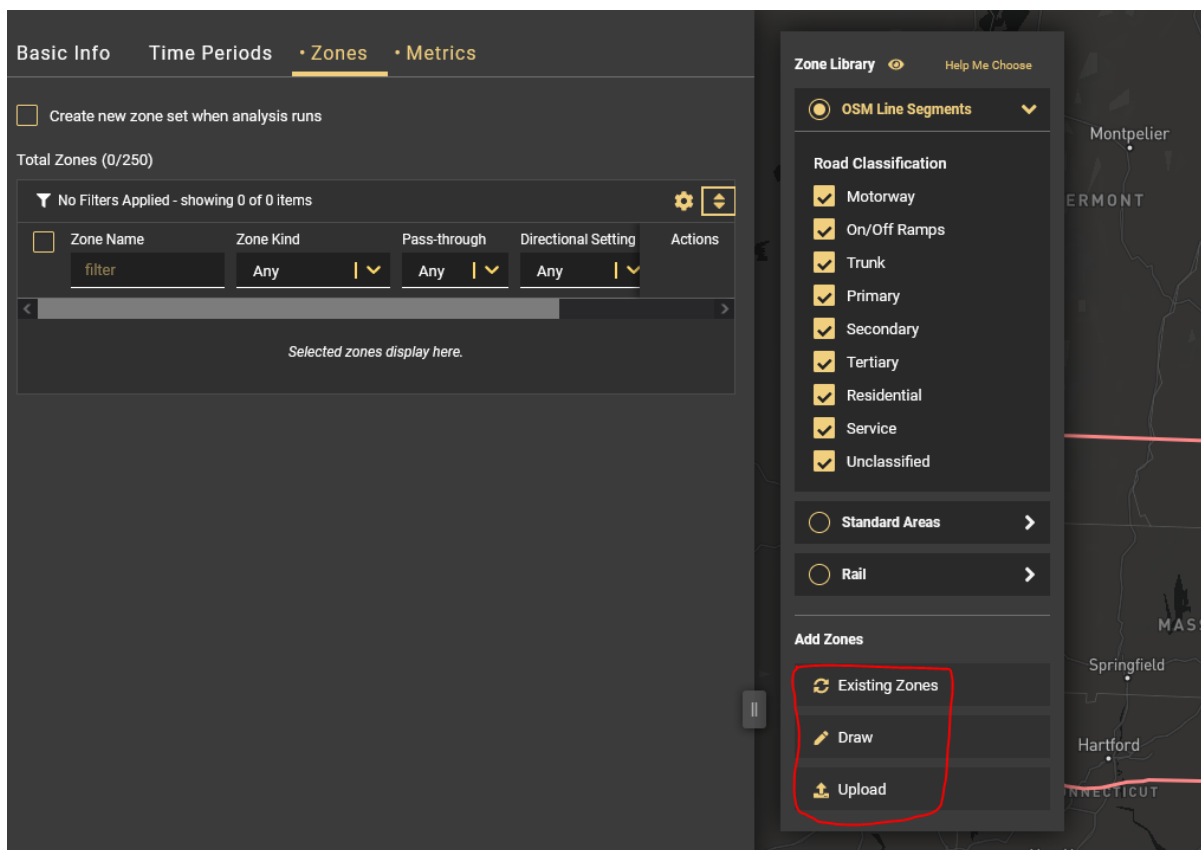
Figure 4.6: Zones information for creating a new StreetLight zone analysis

For metrics information in Figure 4.7, the "Geographies" dropdown list has three options: 2020 US Census TAZ, 2020 US Census Block Groups, and 2020 US Census Zip Codes. This will decide how the trip origin and destination information are organized. For example, the trips starting in a selected development will go to different destinations. Also, trips ending in that development are from many different origins. The "Geographies" option will affect how these origins and destinations are represented (e.g., 2010 TAZ, 2020 Block Groups). The focus of this study is to find the numbers of trips that start and end in a development, not their origins and destinations. Therefore, any one of the three options will work. For the analyses in this report, the 2010 US Census TAZ option was used. "Get Trip Metrics" and

"Get Traveler Metrics" can be left unselected. Choosing these two will provide detailed information (e.g., trip length, traveler demographics) about those trips, which was not necessary for this study.

Figure 4.7: Metrics information for creating a new StreetLight zone analysis

4.2 StreetLight Data Post-Processing

A sample of the trip generation outputs from StreetLight is provided in Table 3.2. It needs to be converted into the format shown in Table 4.1 to facilitate model developments. The results in Table 4.1 are for the total average numbers of trips (both inbound and outbound) for weekday (Monday-Friday) all day, weekday a.m. and p.m. peaks, weekend (Saturday and Sunday) all day, and weekend a.m. and p.m. peaks. A Python script has been developed to perform this data conversion. A full explanation for running the script is explained in Chapter 4.5, Detailed Steps for Running Python Scripts in ArcGIS Pro. The script is provided in Appendix A. Streetlight Processing Script.

Table 4.1: Total trips for each development

Site/Development ID	Weekday All Day Trips	Weekday AM Trips	Weekday PM Trips	Weekend All Day Trips	Weekend AM Trips	Weekend PM Trips
1	3280	160	331	4130	322	370
2	5174	260	543	7411	545	792
3	2934	170	301	4184	379	392
4	8122	397	775	10130	547	963
5	5394	338	540	7240	440	694
6	5619	262	560	7632	511	696
7	4632	225	474	6404	336	536
8	4305	215	419	6093	302	578
9	5405	321	492	7859	442	733

4.3 CTPP Data Processing

The Census Transportation Planning Products Program (CTPP) provides a variety of variables that could potentially be used for predicting trip generation. These variables are categorized into five main groups as in Figure 4.8. Among these variables, the two most relevant ones are population and employment. Therefore, total population (A101100 in Figure 4.9[a]) and total workers (A202100 in Figure 4.9[b]) were chosen for this study. These two variables are obtained at the Traffic Analysis Zone (TAZ) level and further divided by TAZ area measured in square miles.
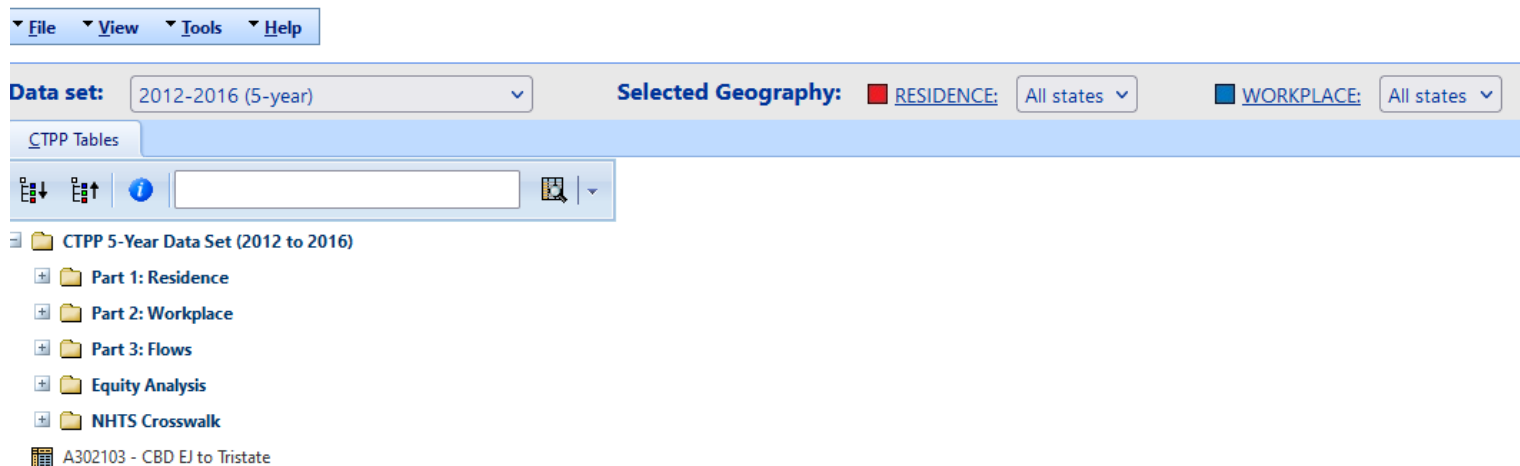
Figure 4.8: Overview of variables in the CTPP database
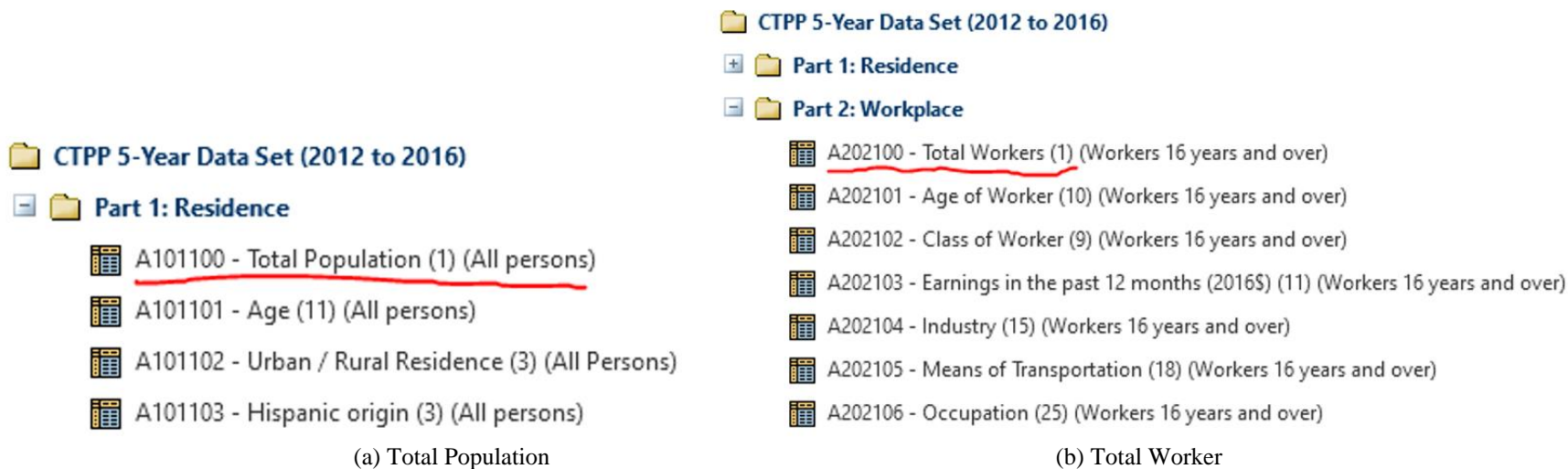
(a) Total Population

(b) Total Worker

Figure 4.9: CTPP variables chosen for this study
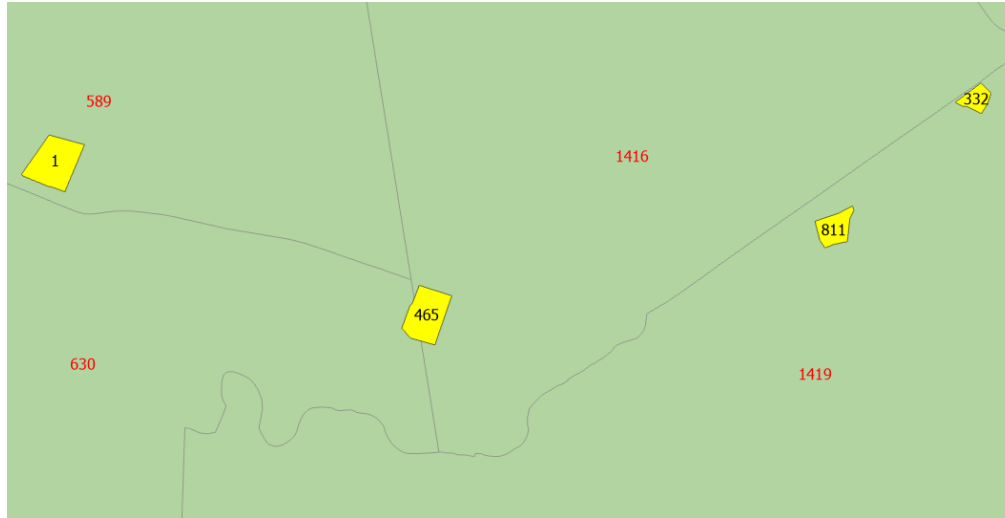
Figure 4.10: Match developments with TAZ

To match developments with the TAZ-level population and worker data, the strategy illustrated in Figure 4.10 was used. Yellow polygons represent developments and green polygons are for TAZs. If a development is completely within a TAZ, the corresponding TAZ's population and worker densities are assigned to that development. If a development overlaps with multiple TAZs, the center of the development is used to make the final decision. For development #465 in Figure 4.10, its center was in TAZ #1416. Therefore, the population and worker densities of TAZ #1416 were assigned to development #465.

As discussed in Chapter 3.5, both population density (people/square mile) and employment density (number of workers/square mile) were found to be statistically insignificant at the 0.05 level in the trip generation model development. Therefore, these variables were not included in the research team's final models. In future research, additional variables not in the CTPP database should be considered, such as the number of pumps at a gas station.

4.4 Identification of Polygons

The following definitions were used to identify polygons to be fed into StreetLight. The research team primarily used Google Maps, Google Satellite Images, Google Street Views, Google Earth, and MassGIS Property Tax Parcels layers to identify appropriate polygons for each land use category and determine their building square footage values. This is a relatively time-consuming process and requires the analysts to be familiar with the ITE land use codes.

Some main challenges faced during this process included businesses being opened or closed, inconsistent building square footage results in the property tax parcel database, buildings shared by multiple businesses, and parking lots shared by multiple businesses. These factors made it a difficult task to draw an accurate polygon to clearly separate the trips from and to a

particular development. Below are some general guidelines used in this research to identify polygons.

- 710 General office building: A general office building houses multiple tenants; it is a location where affairs of businesses, commercial or industrial organizations, or professional persons or firms are conducted. An office building or buildings may contain a mixture of tenants including professional services, insurance companies, investment brokers, and tenant services, such as a bank or savings and loan institution, a restaurant or cafeteria, and service retail facilities. A general office building with a gross floor area of 5,000 square feet or less is classified as a small office building (Land Use 712).
- 760 R&D: Single building or complex of buildings devoted to research and development. May contain offices and light fabrication facilities.
- 820 Shopping Center: A shopping center is an integrated group of commercial establishments that is planned, developed, owned, and managed as a unit. A shopping center's composition is related to its market area in terms of size, location, and type of store. A shopping center also provides on-site parking facilities sufficient to serve its own parking demands. Factory outlet center (Land Use 823) is a related use.
- 150 Warehouse: A warehouse is primarily devoted to the storage of materials, but it may also include office and maintenance areas. Long-term storage facility.
- 155 High-Cube Fulfillment Center Warehouse: High-cube fulfillment center warehouses include warehouses characterized by a significant storage function and direct distribution of e-commerce products to end users. These facilities typically handle smaller packages and quantities than other types of HCWs and often contain multiple mezzanine levels. More detailed descriptions provided by ITE also note that the range of sizes studied for this use are between 818,000 sq ft to 1,466,000 sq ft. Short-term, automated warehouse, direct distribution of e-commerce products to end users (e.g., Amazon).
- 960 Convenience Market/Gas Station: Use when fueling positions > 10 and building size > 3,000 sq ft.
- 140 Manufacturing: A manufacturing facility is an area where the primary activity is the conversion of raw materials or parts into finished products. Size and type of activity may vary substantially from one facility to another. In addition to the actual production of goods, manufacturing facilities generally also have office, warehouse, research, and associated functions.
- 220 Multifamily Housing (Mid-Rise up to Two Floors): Mid-rise multifamily housing includes apartments, townhouses, and condominiums located within the same building with at least three other dwelling units and that have between three and 10 levels (floors).
- 850 Supermarket: Freestanding retail stores selling a complete assortment of food, food preparation and wrapping materials, and household cleaning and servicing items. Supermarkets may also contain facilities such as money machines, photo centers, pharmacies, and video rental areas.
- 813 Freestanding Discount Superstore: Big box store with full grocery department (e.g., Super Walmart, Super Target)
- 815 Freestanding Discount Store: Big box store without full grocery department (e.g., Walmart, Target).

For the MassGIS property tax parcel database, the research team mainly used the L3_TAXPAR_POLY layer and L3_ASSESS table. The L3_TAXPAR_POLY layer was attribute joined to the L3_ASSESS table by the LOC_ID field. The L3_ASSESS table contains important information such as BLD_AREA, RES_AREA, UNITS, NUM_ROOMS, STORIES_NUM, and STORIES. However, the team found that some values were inaccurate and inconsistent and, therefore, could not do a simple spatial join to transfer these values into the Shapefile prepared earlier. The research team reviewed the identified polygons one by one and examined the corresponding records in the property tax parcel database. BLD_AREA and RES_AREA in many cases were the same. When they were not the same, Google products were used as a reference to determine which value was more reasonable and that was used for both BLD_AREA and RES_AREA. In the final Shapefile, these two fields were renamed as "TotalBLD_A" and "TotalRES_A" and they had the same value. In addition to these two new fields, the research team also added "Worker_Den" and "Pop_Den," although these two variables were later found to be statistically insignificant at the 0.05 level.

The MassGIS property tax parcel database was found to be useful to identify additional polygons. The L3_TAXPAR_POLY layer had an attribute field called LU_CODES (*31*), which is not the same as the ITE land use code system. However, it helped to guide the research team to the right parcels to create polygons.

4.5 Detailed Steps for Running Python Scripts in ArcGIS Pro

To set up and run the Python scripts within ArcGIS Pro, users can follow the steps below:

1. Cloning and setting up virtual environments in ArcGIS Pro
   a. First navigate to "Project" in the top left corner, then "Package Manager." Click on the cog on the right in the screenshot below in Figure 4.11.

Figure 4.11: ArcGIS package manager

b. The default ArcGIS Pro environment must be cloned. A window called "Environment Manager" should open, which will have only one environment (see Figure 4.12). Click on the clipboard icon to clone the default environment and name the clone. Avoid using spaces in the file path and name as that may cause issues with ArcGIS.

Figure 4.12: ArcGIS cloning default environment from Environment Manager

c. The cloned environment should then be set to be active, as in Figure 4.13.
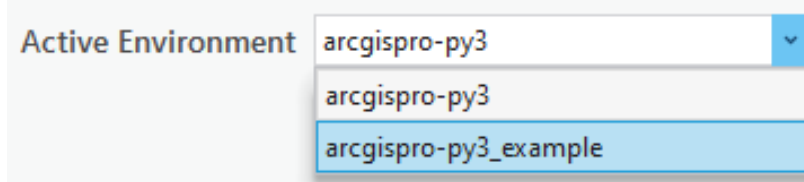
Figure 4.13: Setting the active environment to the cloned environment

d. The final step is to add the required packages. Navigate to the "Add Packages" tab, which is highlighted in the screenshot below (see Figure 4.14). In the search bar, type in "statsmodel" and install it by clicking the install button on the right.
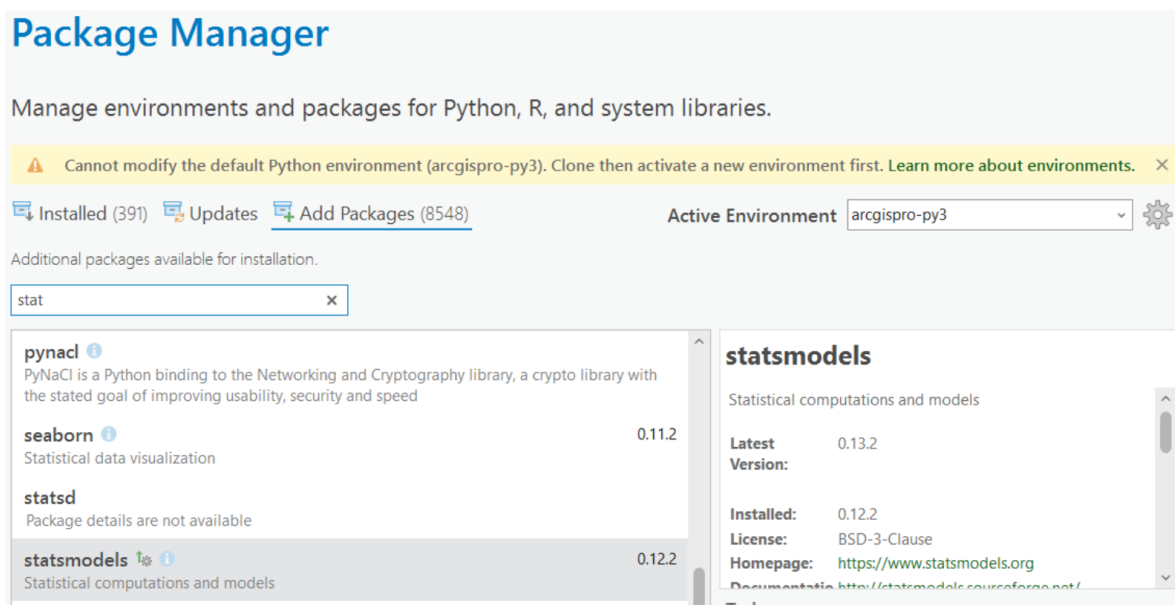
Figure 4.14: Adding new statsmodels to the cloned environment

e. Whenever model fitting needs to be run, this cloned environment must be set to "active." If it is not, Python will not have the required packages installed to run and an "ImportError: No module named statsmodels" will occur.

2. Importing shapefiles and Python scripts:
   a. After creating the shapefile using StreetLight following the steps above, it must be imported into ArcGIS. A folder connection needs to be established in the catalog pane. The folder should contain the shapefile, which can then be dragged and dropped onto the map.
   b. The StreetLight processing and model fitting Python scripts then need to be imported. First create two new notebooks by navigating to "Insert," then "New Notebook" (see Figure 4.15).

43
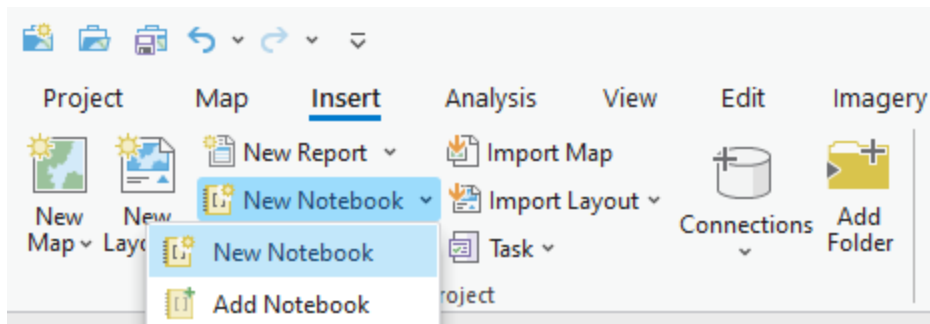
Figure 4.15: Inserting new notebooks to ArcGIS

    c.   The code can then be copied and pasted to the notebooks (see Figure 4.16); these can be renamed "Streetlight_Processsing" and "Model_Fitting."
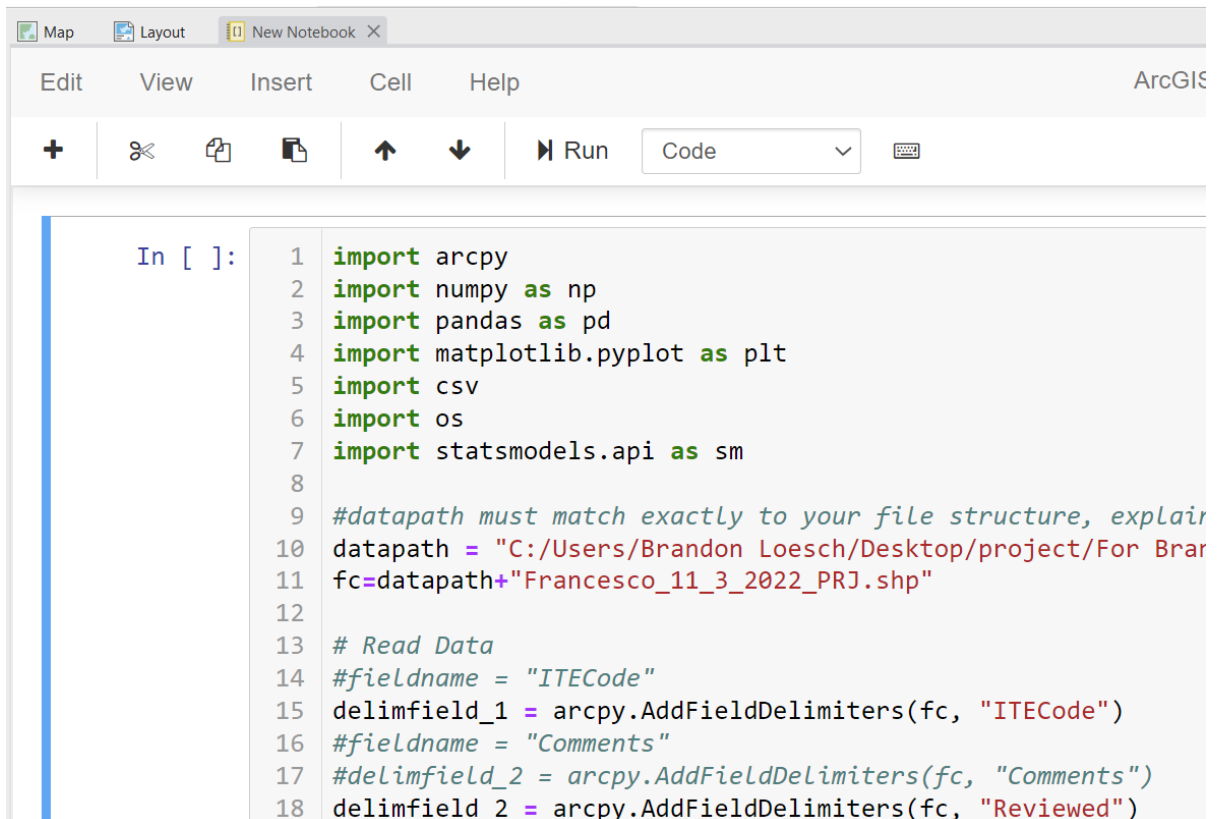
Figure 4.16: Copy and paste Python code to ArcGIS notebook

3. Editing file paths in Python scripts:
    a.   First the StreetLight processing script needs to be edited. On Line 8, the CSV file (see Figure 4.16) that contains the raw data from StreetLight is read. This file path needs to be changed to match where the file is saved on the user's computer. Navigate to the folder where it is saved and copy the path (see Figure 4.17).

44
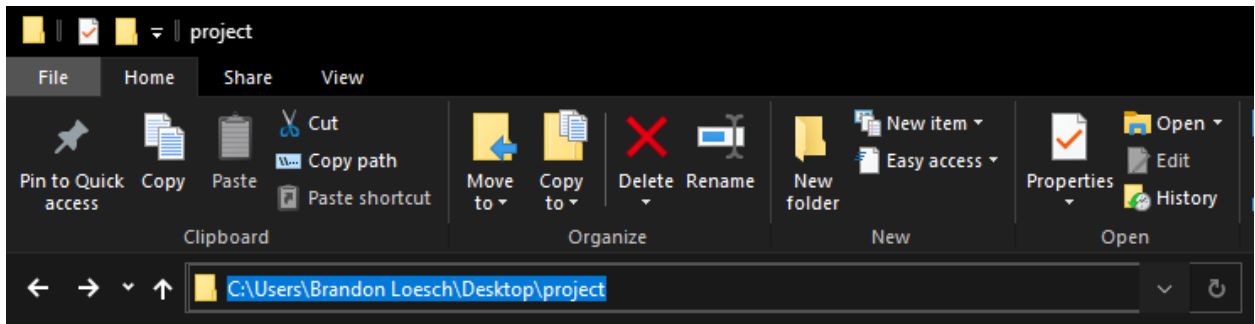
Figure 4.17: Copying file path from Windows file explorer

    b. In ArcGIS, open the "Streetlight_Processing" notebook and replace the current file path with the copied path. Python syntax requires the code to use a forward slash (/) while Windows uses a backslash (\), so the slashes need to be replaced. When unzipping the raw data downloaded from StreetLight, there will be several files. The file needed will be in the "Zone Activity" folder and the file name will end with "zone_odg_all" as in Figure 4.18.

```
In [1]:   1  import pandas as pd
          2  import numpy as np
          3  import datetime
          4  import arcpy
          5
          6  columns = ["Intersection Type", "Zone ID", "Day Type", "Day Part", "Average Daily Zone Traffic
          7  df = pd.read_csv("C:/Users/Yuanchang_Xie/PycharmProjects/ITE_Prj/949481_Copy_of_Zone_740_6_15_z
```

Figure 4.18: Streetlight_Processing notebook

    c. Do the same thing for saving the processed data to a new file. This occurs at the end of the script, Line 273. The file path can be saved in the same folder as the input file but should be named differently.

    d. This is also required for the "Model_Fitting" notebook, and a similar process can be followed. As shown in Figure 4.19, Line 9 should be edited to the user's file path where the shapefile is saved (but not including the shapefile). Line 10 should be edited to the user's shapefile name. Line 37 should also be edited to the processed StreetLight data file name.

```
In [ ]:   1  import arcpy
          2  import numpy as np
          3  import pandas as pd
          4  import matplotlib.pyplot as plt
          5  import csv
          6  import os
          7  import statsmodels.api as sm
          8
          9  #datapath must match exactly to your file structure, explained in report
         10  datapath = "C:/Users/Brandon Loesch/Desktop/project/For Brandon 2-22-2023/
         11  fc=datapath+"Francesco_11_3_2022_PRJ.shp"
```

Figure 4.19: Model_Fitting notebook

45

e. Within the folder that is written in Line 9 (Figure 4.19), create a folder called "ModelResults"; this is where all the results (including figures, R^2, and linear regression) will be saved.

4.6 Updating the Developed Models

To update the developed models, users can follow the steps below:

- Update the Shapefile by adding or removing polygons.
- Upload the Shapefile to StreetLight and create a new analysis. Choose the appropriate time periods and other settings (see Chapter 4.1).
- Use the Script in **Appendix A. Streetlight Processing Script**, to convert the data.
- Add attribute fields such as "TotalBLD_A," "TotalRES_A," "Worker_Den," and "Pop_Den" to the Shapefile as needed and make necessary changes to the "Model_Fitting" Script in **Appendix B. Model Fitting Script**.
- Run the Script in **Appendix B. Model Fitting Script** to get the new models in Excel files.

5.0 Evaluation of the Accuracy of LBS Estimates

Although deriving trip generation rates from LBS data is very convenient, it is important to collect ground truth data to validate the derived trip generation rates. The data generated by StreetLight are just estimates, so they cannot be used with complete certainty for developing trip generation models. A comparison should be made between StreetLight and ground truth data. While the ideal scenario for generating trip generation models would be a 100% match between StreetLight and ground truth trip data, a consistent ratio between StreetLight and ground truth data would suffice.

A challenge here is to collect ground truth data. Cameras have been widely used to collect traffic data. However, manually reviewing such data to extract trip information is an extremely time-consuming process. A complete computer vision and deep learning-based solution to automatically extract trip rates from traffic videos has been developed as part of this study. This cost-effective tool can be used by transportation agencies for trip generation data collection ranging from a few days to several weeks. It works well during both day and night.

This chapter first describes the traffic video data collection efforts to validate the StreetLight data used in this research. It further presents a deep learning model developed to automatically process the collected video data. At the end, the ground truth trip generation results derived from the traffic videos are compared to those obtained from StreetLight.

5.1 Site Selection Methodology for StreetLight Data Collection

To request StreetLight trip data, approximately 800 trip analysis polygons (TAPs) were first drawn in ArcGIS Pro. The geometry and building area attribute data of each TAP synthesized the real-world practical boundary of the property in question with the corresponding tax parcel polygon (TPP) data in the MassGIS Property Tax Parcels layer (*28*). This synthesis was accomplished by incorporating and/or spatial joining information from ArcGIS Pro, Google Maps, and Google Street View. The primary objectives of each TAP were to:

1. Enclose a property dedicated to one ITE land use code classification and isolate and capture all the automobile trips induced/attracted solely by the property.
2. Establish an accurate value for the building area within the TAP.

Properties for which no TAPs could satisfy both objective 1 and objective 2 were excluded from the analysis. To draw an appropriate TAP for a particular property, the TAP should capture all trips directly related to the property and exclude trips related to neighboring properties. This is particularly important for properties in dense urban areas, where one parking lot may serve multiple properties. In this case, if the neighboring properties all have

the same ITE land use code, these properties and the parking lot should all be included in the TAP. If these properties have different land use codes and they all share the same parking lot, it is difficult to split the parking lot into multiple TAPs with different land use codes. In this case, they should still be grouped into one TAP, which can be used to study trip generation for mixed-use developments.

Consider hypothetical properties 1 and 2, each of which has a parking lot adjacent to its respective building. If a single access road connects both parking lots to the larger transportation network, the TAP for property 1 should only include the parking lot for property 1 and should not include any of the access road. If property 1 and property 2 share the same parking lot, it is impossible to isolate the trips for property 1 from the trips for property 2. Thus, both property 1 and property 2 should be excluded from the analysis in this research unless both properties have the same ITE land use code. If they share the same ITE Code, their properties and trip data may be aggregated into one TAP.

If one building contains multiple private enterprises, the property could be included or excluded from the analysis depending on whether all those enterprises belong to the same ITE land use code classification. If all the enterprises belong to the same classification, the property would be included within a TAP. If any one of the enterprises deviates from the others in terms of its classification, the property would be excluded from this research.

After identifying an initial set of TAPs, the suitability of each TAP was evaluated visually using Google Maps and Google Street View. Google Maps shows the names and websites of the private enterprises within each TAP. If a TAP included enterprises belonging to different land use codes in the same building, this TAP was removed from the initial set. If a TAP included areas belonging to different ITE land use code classifications, it was flagged either for redrawing (if this could clearly exclude areas that were not in the same land use classification) or for removing (if there was no way to clearly separate them; for example, they share a surface parking lot). Any TAP that passed the above inspection would then be evaluated according to objective 2 mentioned at the beginning of this section.

Accurately obtaining the building area of a TAP was not a straightforward process. The building area attribute data in the MassGIS Property Tax Parcels layer sometimes was clearly inaccurate. In such cases, the building area was measured manually by using ArcGIS Pro and Google Maps.

5.2 Ground Truth Data Collection Tools and Methods

Low-resolution trail cameras were used to record videos at the entrances and exits of sites. They were installed at a site for several days to gather recordings. These cameras are motion-activated, so whenever a vehicle entered the frame, it would be activated and would record for one minute. The cameras were strapped to trees or poles that were available by the site. It's important to set the camera angle properly. If the camera only captures a fraction of an entering or exiting vehicle (e.g., the wheel or roof), the derived vehicle counts may be

inaccurate. Also, if the camera also captures vehicles on adjacent main roads, it will generate unnecessary challenges to the data post-processing.

Table 5.1: Summary of traffic video data collection efforts

Site ID	Coordinates	Status	Number of attempts
213	71.2824802°W 42.5491356°N	Processed	2
256	71.3590126°W 42.6275635°N	Processed	2
373	71.1440693°W 42.7444794°N	Processed	1
374	71.1491822°W 42.7264368°N	Processed	1
376	71.1183916°W 42.6679587°N	Processed	1
379	70.8752611°W 42.5709926°N	Processed	1
381	71.0292762°W 42.4377476°N	Data lost	1
383	71.0189033°W 42.4375883°N	Data lost	1
408	71.2779737°W 42.5491485°N	Processed	2
409	71.2024584°W 42.7028040°N	Processed	1
410	71.1515691°W 42.4652884°N	Processed	1
430	70.9872773°W 42.3784298°N	Data lost	1
431	70.9780675°W 42.4300244°N	Card stolen	1
432	70.9384489°W 42.4710439°N	Data lost	2
434	70.9474399°W 42.5653927°N	Processed	1
436	71.1047262°W 42.5259379°N	Data lost	1
437	71.1106157°W 42.4685095°N	Processed	1
443	71.1178784°W 42.4204972°N	Processed	1
445	71.2357860°W 42.2932602°N	Processed	1
682	71.2529766°W 42.3230864°N	Processed	1
690	71.2305427°W 42.4395680°N	Processed	1
707	71.1258121°W 42.5065097°N	Processed	1
761	71.3188037°W 42.5908063°N	Processed	2
774	71.0058265°W 42.5109148°N	Processed	1

Another possibility of inaccurate vehicle counting exists if a vehicle enters or exits the frame at the end of the minute recording. The camera had a one-minute recording limitation. At a high-volume site, vehicles may be entering and exiting constantly, so a recording can always be occurring. It is possible that a vehicle may enter the frame at the end of a recording and still be in the frame at the start of the next recording. Similarly, depending on the time between ending and starting a recording, a vehicle could pass through without being counted. This could lead to inaccuracies in the vehicle counting.

Table 5.1 summarizing sites where traffic video data was collected. During data collection, some attempts were unsuccessful. The camera location and angle needed to be readjusted to get better results. The research team had to visit those sites twice. For several sites, videos were lost due to damaged/stolen memory cards or similar issues.

5.3 Recorded Video Data Processing

Deriving trip generation data from the recorded traffic videos is essentially an object detection and tracking problem, which has been an important topic in computer vision for many years. The advancement of deep learning has provided a powerful tool for object detection and tracking. In this research, a widely used deep learning model called YOLOv8 was adopted for vehicle detection, and the Deep-SORT algorithm was used for vehicle tracking. YOLO stands for You Only Look Once and is one of the state-of-the-art models for object detection. YOLOv8 builds upon the success of previous YOLO versions by incorporating new features such as spatial pyramid pooling (SPP), SPP-Net, Feature Pyramid Network (FPN), and PANet to improve the detection accuracy and reduce false positives. YOLOv8 is an anchor-free object detection algorithm. Unlike anchor-based methods, which use predefined anchors or priors to predict object locations and shapes, anchor-free methods directly predict the object locations without requiring anchor boxes. YOLOv8 employs a center-based mechanism to detect objects, which makes it more efficient and accurate than anchor-based methods.

YOLOv8 comes in various sizes, each designed for different use cases and hardware constraints. The smallest size is YOLOv8-Nano, which has the least number of model parameters, making it suitable for deployment on edge devices with limited computational resources. YOLOv8-Small has a slightly larger network size and more parameters than YOLOv8-Nano, making it suitable for deployment on low-end hardware such as mobile devices. The YOLOv8-Medium size is designed for mid-level hardware, such as desktop computers or laptops, and offers a balance between accuracy and speed. YOLOv8-Large has the largest network size and the most parameters, making it suitable for high-end hardware such as server-grade graphics processing units (GPU) or cloud computing. It provides the highest accuracy and can handle complex scenes with multiple objects. Choosing the right YOLOv8 size depends on several factors such as the target hardware, the complexity of the scene, and the desired speed and accuracy. Using a larger size will generally result in higher accuracy, but at the cost of slower inference times and higher computational requirements. In the experiments for this research, the research team used YOLOv8-Medium as it provided the right balance between speed and accuracy. Also, it should be noted that the larger models did not perform clearly better than the YOLOv8-Medium model.

Deep-SORT is a popular tracking algorithm that has been widely used in the computer vision community. It is a time-tested algorithm and is known for its high tracking accuracy and robustness. Unlike other tracking algorithms that rely solely on appearance features, Deep-SORT combines appearance features with motion information to improve tracking performance. It achieves this by using a deep neural network to extract appearance features, and a Kalman filter to model the motion information. Additionally, Deep-SORT employs a

matching algorithm to associate detections across frames, which further improves tracking accuracy. Overall, Deep-SORT is a reliable and effective tracking algorithm that remains relevant in the field of computer vision. By utilizing the YOLOv8 and Deep-SORT algorithms in tandem, it is possible to enhance the accuracy and speed of the vehicle detection and tracking process.

(a) Very few vehicles are detected in the RGB image.

(b) Several vehicles are detected in the above IR image.

Figure 5.1: Comparison of detection results based on RGB and IR cameras

In this study, both RGB and Infrared (IR) videos were tested as inputs for vehicle detection and tracking. RGB video captures the visible spectrum of light, which is the same spectrum that the human eye can see. Therefore, it is suitable for daytime applications when there is ample natural light. RGB cameras perform best in environments with good lighting conditions, such as outdoor scenes or well-lit indoor spaces. In contrast, IR video captures the infrared spectrum of light. This type of video is particularly useful in low-light and nighttime conditions. They can provide enhanced visibility and detection capabilities in challenging lighting conditions. Depending on whether it is daytime or night, the use of either RGB or IR videos can provide more reliable and robust detection and tracking results, as it can compensate for the limitations of each video type. Figure 5.1 shows how the use of IR cameras can improve detection in low-light conditions.

Deep learning models usually are trained using large datasets such as COCO or OpenImages, which contain tens of thousands of annotated images covering hundreds of object categories. These datasets are used to train the neural network to recognize and localize objects accurately in a variety of contexts and situations. The default YOLOv8 models are trained

using a supervised learning approach, which involves presenting the neural network with a set of input images and their corresponding ground-truth labels and adjusting the network's weights and biases to minimize the difference between the predicted and ground-truth labels. The model architecture uses a combination of convolutional and pooling layers to extract features from the input images, followed by several fully connected layers to perform object classification and localization. During training, the YOLOv8 algorithm learns to identify object features, such as edges, corners, and textures, and to use them to recognize and classify objects accurately. The default YOLOv8 models are trained using a GPU-accelerated implementation, which allows for faster training times and higher model accuracy.

YOLOv8 can accurately localize and identify partially appearing objects in a variety of contexts and scenarios. Figure 5.2 shows two such instances based on videos collected as part of this study. Also, when the conditions are right, even smaller objects such as a backpack can be detected with high confidence.

Figure 5.2: Nighttime detection results using IR camera

Once the YOLOv8 algorithm has processed the input videos and detected vehicles, the output typically includes a set of bounding boxes that enclose each detected object, along with a confidence score indicating the algorithm's confidence in the detection, as shown in Figure 5.2. These bounding boxes can be used to track the detected vehicles over time and to estimate their trajectories and velocities. Deep-SORT can associate the detected bounding boxes with unique vehicle identities and track them over multiple frames of the input video,

even in cases where the vehicles are occluded or partially hidden from view. By using the output of YOLOv8 as input to Deep-SORT, it is possible to achieve highly accurate and reliable vehicle tracking results in real-world scenarios. Figure 5.3 shows how tracking results can be visualized using RGB videos collected on the UMass Lowell campus.

Figure 5.3: Visualization of tracking on detection results

5.4 Comparison of Ground Truth Data with StreetLight Estimates

The ground truth data collected from the trail cameras was then compared with StreetLight and the ITE's Trip Generation Manual estimates. This section focuses on ITE land use code 221, mid-rise multifamily housing. Therefore, some sites listed in Table 5.1 are omitted here, as they do not belong to land use code 221. StreetLight data for the selected sites was available from 2019 to 2022, and data in 2023 was unavailable at the time of this study. ITE estimates were taken from equations given in the 11th edition of the ITE Trip Generation Manual. These equations are below:

$$AM\ Peak: Trips = 0.44 * (Dwelling\ Units) - 11.61 \qquad (10)$$
$$PM\ Peak: Trips = 0.39 * (Dwelling\ Units) + 0.34 \qquad (11)$$

The numbers of dwelling units for each site were collected from public GIS property databases, including the MassGIS Property Tax Parcels layer. Not all sites had complete data, so some estimations were made regarding dwelling units when necessary. It is important to note that the ground truth data was only collected for several days at each site. Therefore, some fluctuations or randomness in the collected trip generation data is expected.

The video processing program (see Chapter 5.3) yielded over- or under-counting at some locations due to reasons mentioned at the beginning of Chapter 5.2. Therefore, the videos for peak hours were reviewed manually to ensure the ground truth data was accurate. Figure 5.4 shows a comparison of data for site #373, which is in Methuen, MA. The peak hour counts generally look the same between the ground truth data and StreetLight data while the ITE estimates seem to overestimate. The comparison results for other developments belonging to

land use code 221 are presented in Appendix E. From the comparison figures, it can be observed that:

- The ITE Trip Generation Manual in general overestimates trips compared to the collected ground truth data (See sites 373, 374, 379AM, 408AM, 434, 690).
- Overall, the 2019 StreetLight data matched the ground truth better than the 2020 and 2022 data. This is likely due to the impacts of COVID. One example is Zone/Site 443. The 2023 StreetLight data for April and May (when the ground truth data was collected) was not ready yet. It would be interesting to bring the 2023 data into the comparison when it is available.
- The a.m. data for Zone 409 suggests that the number of trips can vary significantly within rush hours. However, such trends were not captured by the ITE method.
- The 2019, 2020, and 2022 StreetLight data were only for April, not the entire year. Overall, the daily trends for these years were noisy. Adding data from other months may help to make the daily trends much smoother.
- The trends in different figures should not be compared directly, since the y axes of these figures have different ranges.

Table 5.2 shows the absolute percentage errors of the 2022 StreetLight data and ITE estimates compared to the ground truth. The StreetLight data has an average error of 67% while the ITE data has an average error of 94%. This gives StreetLight data a slight advantage compared to ITE estimates, suggesting that using StreetLight data for estimating trip generation rates is feasible and promising.

For future research, the following additional topics can be explored:

- Add StreetLight data from additional months to the comparison.
- Add 2023 StreetLight data to the comparison when it is available.
- Separate weekday and weekend ground truth data.
- Compare data from other land use categories. This will require collecting additional traffic videos and processing them using the developed AI tool.
- Take weather information into consideration.
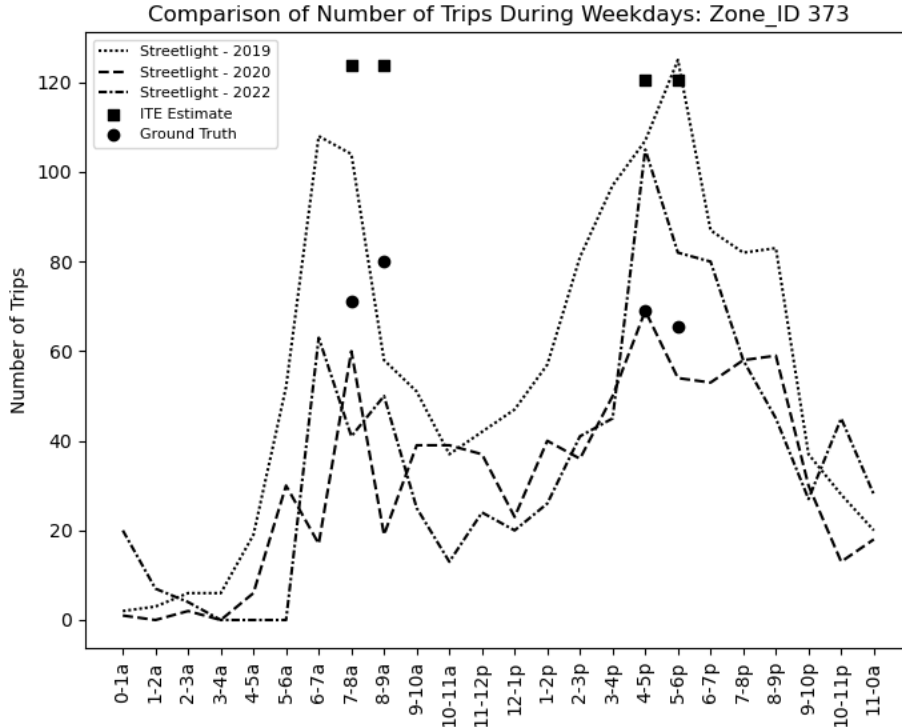
Figure 5.4: Comparison of StreetLight (2019-2022), ITE, and ground truth data for ID 373

Table 5.2: Absolute percentage errors of StreetLight and ITE data compared to ground truth

Site	Streetlight Data 7-8 AM	Streetlight Data 8-9 AM	Streetlight Data 4-5 PM	Streetlight Data 5-6 PM	ITE AM	ITE PM
373	42%	38%	52%	25%	64%	79%
374	85%	71%	34%	30%	207%	25%
376	267%	37%	118%	145%	98%	15%
379	201%	115%	5%	42%	117%	14%
408	24%	56%	47%	45%	219%	5%
409	56%	91%	100%	100%	281%	150%
410	18%	77%	143%	111%	1%	67%
434	63%	0%	8%	0%	166%	49%
437	71%	83%	73%	57%	115%	30%
443	69%	66%	72%	88%	23%	18%
682	67%	79%	48%	9%	45%	9%
690	100%	0%	28%	35%	424%	131%
774	222%	9%	34%	34%	60%	14%
Average Across Sites	99%	56%	59%	55%	140%	47%

Average for each method: StreetLight data, 67%; ITE, 94%

This page left blank intentionally.

6.0 Conclusions and Discussion

This project aims to develop Massachusetts-specific trip generation models for high-priority land uses. Given the difficulty in collecting trip generation data, this research proposes an innovative approach of utilizing LBS data for trip generation model development. Utilizing the StreetLight data, a popular LBS data source, this research has developed a set of trip generation models for ten high-priority land uses for Massachusetts. It also provides detailed guidance in how to develop and update trip generation models using StreetLight data. This guidance can be adjusted to take other LBS data sources as the input instead of the StreetLight data. In addition, an AI tool has been developed to automatically extract vehicle counts from traffic videos. Such data can be used to either directly develop trip generation models or to validate the performance of models based on LBS data.

Building square footage, employment opportunity density, and population density were considered as the explanatory variables. However, only the building square footage was found to be statistically significant in estimating trip generations. Three sets of models were developed using data in 2018, 2021, and 2022. An interesting finding was that the 2018 models for 710 (general office building) and 760 (research and development) had better goodness-of-fit than the corresponding models fitted based on the 2021 and 2022 data. A possible reason is that many people were allowed to work from home during COVID. This shows that using LBS data makes it possible to capture changes in trip generation due to major events, which is a significant advantage compared to using the traditional ITE Trip Generation Manual method for estimating trips. Overall, models for weekdays had better goodness-of-fit than those for weekends. Additionally, it was found that:

- The R^2 results for the following land use categories were **good**: 221 (mid-rise multifamily housing), 820 (shopping centers), and 813 and 815 (freestanding discount stores and super stores);
- The R^2 results for the following land use categories were **reasonable**: 960 (convenience market/gas station), 850 (supermarkets), and 150 (warehouses); and
- The R^2 results for the following land use categories were **poor**: 140 (Industrial—manufacturing), 155 (high-cube fulfillment center warehouse such as Amazon), 710 (general office building), and 760 (research and development).

The step-by-step guidance includes detailed instructions on how to retrieve StreetLight trip data, data processing, and model fitting. Python scripts are provided as part of the deliverables of this research to allow MassDOT to (1) update the developed models when new data is available and (2) develop new models for other types of land use in the future.

Trail cameras were used to collect traffic videos from selected sites (primarily for land use code 221—mid-rise multifamily housing). These cameras were installed at the entrances and exits of the locations selected. These videos were then processed using computer vision algorithms, producing automatic vehicle counts. These videos were further processed by the developed AI tool. Overall, the AI tool performed very well and was able to accurately extract vehicle counts. Some of the accuracy issues could be more easily addressed by

changing camera mounting angles rather than further improving the AI tool. To ensure accuracy, the trip counts during peak hours were also validated by manually reviewing the collected videos to generate the ground truth trip generation data. The ground truth data for mid-rise multifamily housing was then compared with the StreetLight trip counts and trips estimated by the ITE Trip Generation Manual. The comparison results suggest that the StreetLight data overall was slightly closer to the ground truth data than the ITE estimates, demonstrating the feasibility and potential of using LBS data for developing trip generation models. Note that the ground truth data was collected in 2023. However, the StreetLight data was for 2019, 2020, and 2022, and the 2023 StreetLight data was not available yet.

Overall, the proposed method using LBS data for developing trip generation models is feasible and promising, although more work is needed to further validate the LBS data for other land use categories. With the developed AI algorithm to automate video data processing, it is also possible to collect ground truth data on a large scale for validating the developed models. In addition, the ground truth data can be used directly to develop Massachusetts-specific trip generation models.

For future work, a larger sample size is needed to further validate the developed models. The current sample size was small (only 13 sites) due to limited time and some unexpected issues. For example, some of the cameras were removed by residents. Also, it would be helpful to develop a custom camera solution that can record continuously as long as there is a motion activation. In addition, it would be very interesting to evaluate data from other LBS data sources.

7.0 References

1   Clifton, K. J., K. M. Currans, and C. D. Muhs. Adjusting ITE's Trip Generation Handbook for Urban Context. *Journal of Transport and Land Use*, Vol. 8, No. 1, 2015, pp. 5–29. http://dx.doi.org/10.5198/jtlu.2015.378

2   Currans, K. M., and K. J. Clifton. Using Household Travel Surveys to Adjust ITE Trip Generation Rates. *Journal of Transport and Land Use*, Vol. 8, No. 1, 2015, pp. 85–119. http://dx.doi.org/10.5198/jtlu.2015.470

3   Clifton, K., and K. M. Currans. Characterizing the Trip Generation Profiles of Multifamily Housing. 2019. https://pdxscholar.library.pdx.edu/cgi/viewcontent.cgi?article=1199&context=trec_reports

4   Gulden, J., J. P. Goates, and R. Ewing. Mixed-Use Development Trip Generation Model. *Transportation Research Record*, Vol. 2344, No. 1, 2013, pp. 98–106. https://doi.org/10.3141/2344-11

5   Givechi, M., and G. Sarkodee. Establishment of Local Trip Generation Rates or Equations for Mixed-Use Developments in Kansas (No. K-TRAN: KU-11-7). Kansas Dept. of Transportation, 2012.

6   Westrom, R., S. Dock, J. Henson, M. Watten, A. Bakhru, M. Ridgway, ... and R. Paradkar. Multimodal Trip Generation Model to Assess Travel Impacts of Urban Developments in the District of Columbia. *Transportation Research Record*, Vol. 2668, No. 1, 2017, pp. 29–37. https://doi.org/10.3141/2668-04

7   Byrne, B. *Vermont Trip Generation Manual* (No. 2010-5). 2010.

8   Wilmot, C., P. Stopher, A. Antipova, R. Gudishala, S. Doulabi, and M. Majumder. ITE Trip Generation Modification Factors for Louisiana. Final Project #FHWA/LA.17/646, LTRC Project Number: 18-4SS. Louisiana Department of Transportation and Development, 2021.

9   Clifton, K. J., K. M. Currans, R. Schneider, S. Handy, A. Howell, G. Abou-Zeid, ... and S. R. Gehrke. Affordable Housing Trip Generation Strategies and Rates (No. CA18-2465). 2018. https://dot.ca.gov/-/media/dot-media/programs/research-innovation-system-information/documents/final-reports/ca18-2465-finalreport-a11y.pdf

10   Shi, F., and L. Zhu. Analysis of Trip Generation Rates in Residential Commuting Based on Mobile Phone Signaling Data. *Journal of Transport and Land Use*, Vol. 12, No. 1, 2019, pp. 201–220. http://dx.doi.org/10.5198/jtlu.2019.1431

11   Clifton, K. J., K. M. Currans, and C. D. Muhs. Evolving ITE Trip Generation Handbook: Proposal for Collecting Multimodal, Multicontext, Establishment-Level Data. *Transportation research record*, Vol. 2344, No. 1, 2013, pp. 107–117. https://doi.org/10.3141/2344-12

12   Kenchappagoudra, M. Improving Trip Generation Estimates for Canadian Sites Using Aggregation and Extraction Techniques (Poster). In *Transportation Association of Canada and ITS Canada 2019 Joint Conference and Exhibition*. 2019. https://www.tac-atc.ca/sites/default/files/conf_papers/tac-poster-mallikarjuna-kenchappagoudra-co.pdf

13   Hard, E., C. Bhat, B. Chigoy, L. Green, S. Dubey, D. Pearson, ... and P. C. Moore. Improved Trip Generation Data for Texas Using Workplace and Special Generator Survey Data (No. FHWA/TX-15/0-6760-1). Texas A&M Transportation Institute, 2015.

14   Currans, K. M. Improving Vehicle Trip Generation Estimations for Urban Contexts: A Method Using Household Travel Surveys to Adjust ITE Trip Generation Rates. 2013. https://pdxscholar.library.pdx.edu/cgi/viewcontent.cgi?article=1986&context=open_access_etds

15   Schneider, R. J., K. Shafizadeh, B. R. Sperry, and S. L. Handy. Methodology to Gather Multimodal Trip Generation Data in Smart-Growth Areas. *Transportation Research Record*, Vol. 2354, No. 1, 2013, pp. 68–85.

16   Dock, S., L. Cohen, J. D. Rogers, J. Henson, R. Weinberger, J. Schrieber, and K. Ricks. Methodology to Gather Multimodal Urban Trip Generation Data. *Transportation Research Record*, Vol. 2500, No. 1, 2015, pp. 48–58. https://doi.org/10.3141/2500-06

17   De Gruyter, C., S. M. Zahraee, and N. Shiwakoti. Site Characteristics Associated with Multimodal Trip Generation Rates at Residential Developments. *Transport Policy*, No. 103, 2021, pp. 127–145. https://doi.org/10.1016/j.tranpol.2021.01.019

18   Tian, G., K. Park, R. Ewing, M. Watten, and J. Walters. Traffic Generated by Mixed-Use Developments—A Follow-Up 31-Region Study. *Transportation Research Part D: Transport and Environment*, No. 78, 2020, p. 102205. https://doi.org/10.1016/j.trd.2019.102205

19   Weinberger, R., K. Ricks, J. Schrieber, L. Cohen, and Symmetra Design, LLC. Trip Generation Data Collection in Urban Areas. 2014.

20   Bregman, G. Trip-Generation Rates for Urban Infill Land Uses in California. 2009.

21   Doulabi, S. Contextual Adjustment of the ITE Trip Generation Rates Using Wi-Fi and Bluetooth Technologies. MS Thesis, Department of Civil Engineering, Louisiana State University, 2020.

22   Tong, P., M. Li, M. Li, J. Huang, and X. Hua. Large-Scale Vehicle Trajectory Reconstruction with Camera Sensing Network. In *MobiCom*, 2021, pp. 188–200.

23   Lee, R., J. Miller, R. Maiss, M. Campbell, K. Shafizadeh, D. Niemeier, ... and T. Parker. Evaluation of the Operation and Accuracy of Five Available Smart Growth Trip Generation Methodologies. 2011.

24   Weinberger, R., S. Dock, L. Cohen, J. D. Rogers, and J. Henson. Predicting Travel Impacts of New Development in America's Major Cities: Testing Alternative Trip Generation Models. *Transportation Research Record*, Vol. 2500, No. 1, 2015, pp. 36–47.

25   Fabregas, A., P. S. Lin, V. Koneru, M. K. Seggerman, A. Pinjari, and M. B. Bochner. Verification of Trip Generation and Trip Internalization Estimation Procedures for Mixed-Use Developments (No. 15-4881). 2015.

26   Currans, K. M. Issues in Trip Generation Methods for Transportation Impact Estimation of Land Use Development: A Review and Discussion of the State-of-the-Art Approaches. *Journal of Planning Literature*, Vol. 32, No. 4, 2017, pp. 335–345. https://doi.org/10.1177/0885412217706505

27   De Gruyter, C. Multimodal Trip Generation from Land Use Developments: International Synthesis and Future Directions. *Transportation Research Record*, Vol. 2673, No. 3, 2019, pp. 136–152.

28   MassGIS Data: Property Tax Parcels https://www.mass.gov/info-details/massgis-data-property-tax-parcels. Accessed on December 28, 2022.
29   Census Transportation Planning Products Program (CTPP) data. https://ctpp.transportation.org/ctpp-data-set-information/. Accessed on December 28, 2022.
30   StreetLight—Choose Your Output Type. https://support.streetlightdata.com/hc/en-us/articles/360029642992. Accessed on December 28, 2022.
31   Massachusetts Property Type Classification Codes. https://www.mass.gov/files/documents/2016/08/wr/classificationcodebook.pdf. Accessed on December 28, 2022.

This page left blank intentionally.

8.0 Appendices

Appendix A. Streetlight Processing Script

```
1    import pandas as pd
2    import numpy as np
3
4    columns = ["Intersection Type", "Zone ID", "Day Type", "Day Part", "Average Daily Zone
     Traffic (StL Volume)"]
5
6
7    df = pd.read_csv("C:/Users/Brandon
     Loesch/Desktop/project/1097333_2021_Sept_Data_Zone_740_6_15_zone_odg_all.csv",
     usecols=columns)
8
9
10   temp = list(df["Zone ID"])
11   ZoneID = []
12   for i in range(830):
13      if temp.count(i) > 1:
14         ZoneID.append(i)
15   res = np.zeros((len(ZoneID), 7))
16
17   for i in range(len(ZoneID)):
18      # ----------------- weekday data goes here --------------------------------
19
20      weekday_allday_in = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
     End') \
21         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
     (df["Day Type"] != '7: Sunday (Su-Su)') \
22         & (df["Day Part"] == '00: All Day (12am-12am)'), ["Average Daily Zone Traffic (StL
     Volume)"]]
23
24      weekday_allday_out = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
     Start') \
25         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
     (df["Day Type"] != '7: Sunday (Su-Su)') \
26         & (df["Day Part"] == '00: All Day (12am-12am)'), ["Average Daily Zone Traffic (StL
     Volume)"]]
27
28      weekday_AM_in_6 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
     End') \
29         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
     (df["Day Type"] != '7: Sunday (Su-Su)') \
30         & (df["Day Part"] == '07: 6am (6am-7am)'), ["Average Daily Zone Traffic (StL Volume)"]]
31      weekday_AM_in_7 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
     End') \
32         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
     (df["Day Type"] != '7: Sunday (Su-Su)') \
```

```
33          & (df["Day Part"] == '08: 7am (7am-8am)'), ["Average Daily Zone Traffic (StL Volume)"]]
34      weekday_AM_in_8 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
    End') \
35          & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
    (df["Day Type"] != '7: Sunday (Su-Su)') \
36          & (df["Day Part"] == '09: 8am (8am-9am)'), ["Average Daily Zone Traffic (StL Volume)"]]
37      weekday_AM_in_9 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
    End') \
38          & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
    (df["Day Type"] != '7: Sunday (Su-Su)') \
39          & (df["Day Part"] == '10: 9am (9am-10am)'), ["Average Daily Zone Traffic (StL Volume)"]]
40      weekday_AM_in_10 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
    End') \
41          & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
    (df["Day Type"] != '7: Sunday (Su-Su)') \
42          & (df["Day Part"] == '11: 10am (10am-11am)'), ["Average Daily Zone Traffic (StL
    Volume)"]]
43
44      weekday_AM_out_6 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
    Start') \
45          & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
    (df["Day Type"] != '7: Sunday (Su-Su)') \
46          & (df["Day Part"] == '07: 6am (6am-7am)'), ["Average Daily Zone Traffic (StL Volume)"]]
47      weekday_AM_out_7 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
    Start') \
48          & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
    (df["Day Type"] != '7: Sunday (Su-Su)') \
49          & (df["Day Part"] == '08: 7am (7am-8am)'), ["Average Daily Zone Traffic (StL Volume)"]]
50      weekday_AM_out_8 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
    Start') \
51          & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
    (df["Day Type"] != '7: Sunday (Su-Su)') \
52          & (df["Day Part"] == '09: 8am (8am-9am)'), ["Average Daily Zone Traffic (StL Volume)"]]
53      weekday_AM_out_9 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
    Start') \
54          & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
    (df["Day Type"] != '7: Sunday (Su-Su)') \
55          & (df["Day Part"] == '10: 9am (9am-10am)'), ["Average Daily Zone Traffic (StL Volume)"]]
56      weekday_AM_out_10 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
    Start') \
57          & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
    (df["Day Type"] != '7: Sunday (Su-Su)') \
58          & (df["Day Part"] == '11: 10am (10am-11am)'), ["Average Daily Zone Traffic (StL
    Volume)"]]
59
60      temp = np.zeros((5, 2))
61      temp[0][0] = float(weekday_AM_in_6.sum()) / 5
62      temp[0][1] = float(weekday_AM_out_6.sum()) / 5
63      temp[1][0] = float(weekday_AM_in_7.sum()) / 5
64      temp[1][1] = float(weekday_AM_out_7.sum()) / 5
65      temp[2][0] = float(weekday_AM_in_8.sum()) / 5
66      temp[2][1] = float(weekday_AM_out_8.sum()) / 5
67      temp[3][0] = float(weekday_AM_in_9.sum()) / 5
```

```
68      temp[3][1] = float(weekday_AM_out_9.sum()) / 5
69      temp[4][0] = float(weekday_AM_in_10.sum()) / 5
70      temp[4][1] = float(weekday_AM_out_10.sum()) / 5
71
72      weekday_AM = []
73      for j in range(5):
74         weekday_AM.append(temp[j][0] + temp [j][1])
75      weekday_AM_Max = max(weekday_AM)
76      tempind = weekday_AM.index(weekday_AM_Max)
77      weekday_AM_in = temp[tempind][0]
78      weekday_AM_out = temp[tempind][1]
79
80      weekday_PM_in_6 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
   End') \
81         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
   (df["Day Type"] != '7: Sunday (Su-Su)') \
82         & (df["Day Part"] == '16: 3pm (3pm-4pm)'), ["Average Daily Zone Traffic (StL Volume)"]]
83      weekday_PM_in_7 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
   End') \
84         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
   (df["Day Type"] != '7: Sunday (Su-Su)') \
85         & (df["Day Part"] == '17: 4pm (4pm-5pm)'), ["Average Daily Zone Traffic (StL Volume)"]]
86      weekday_PM_in_8 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
   End') \
87         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
   (df["Day Type"] != '7: Sunday (Su-Su)') \
88         & (df["Day Part"] == '18: 5pm (5pm-6pm)'), ["Average Daily Zone Traffic (StL Volume)"]]
89      weekday_PM_in_9 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
   End') \
90         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
   (df["Day Type"] != '7: Sunday (Su-Su)') \
91         & (df["Day Part"] == '19: 6pm (6pm-7pm)'), ["Average Daily Zone Traffic (StL Volume)"]]
92      weekday_PM_in_10 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
   End') \
93         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
   (df["Day Type"] != '7: Sunday (Su-Su)') \
94         & (df["Day Part"] == '20: 7pm (7pm-8pm)'), ["Average Daily Zone Traffic (StL Volume)"]]
95
96      weekday_PM_out_6 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
   Start') \
97         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
   (df["Day Type"] != '7: Sunday (Su-Su)') \
98         & (df["Day Part"] == '16: 3pm (3pm-4pm)'), ["Average Daily Zone Traffic (StL Volume)"]]
99      weekday_PM_out_7 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
   Start') \
100        & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
   (df["Day Type"] != '7: Sunday (Su-Su)') \
101        & (df["Day Part"] == '17: 4pm (4pm-5pm)'), ["Average Daily Zone Traffic (StL Volume)"]]
102     weekday_PM_out_8 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
   Start') \
103        & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
   (df["Day Type"] != '7: Sunday (Su-Su)') \
104        & (df["Day Part"] == '18: 5pm (5pm-6pm)'), ["Average Daily Zone Traffic (StL Volume)"]]
```

```
105     weekday_PM_out_9 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
        Start') \
106         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
        (df["Day Type"] != '7: Sunday (Su-Su)') \
107         & (df["Day Part"] == '19: 6pm (6pm-7pm)'), ["Average Daily Zone Traffic (StL Volume)"]]
108     weekday_PM_out_10 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
        Start') \
109         & (df["Day Type"] != '0: All Days (M-Su)') & (df["Day Type"] != '6: Saturday (Sa-Sa)') &
        (df["Day Type"] != '7: Sunday (Su-Su)') \
110         & (df["Day Part"] == '20: 7pm (7pm-8pm)'), ["Average Daily Zone Traffic (StL Volume)"]]
111
112     temp = np.zeros((5, 2))
113     temp[0][0] = float(weekday_PM_in_6.sum()) / 5
114     temp[0][1] = float(weekday_PM_out_6.sum()) / 5
115     temp[1][0] = float(weekday_PM_in_7.sum()) / 5
116     temp[1][1] = float(weekday_PM_out_7.sum()) / 5
117     temp[2][0] = float(weekday_PM_in_8.sum()) / 5
118     temp[2][1] = float(weekday_PM_out_8.sum()) / 5
119     temp[3][0] = float(weekday_PM_in_9.sum()) / 5
120     temp[3][1] = float(weekday_PM_out_9.sum()) / 5
121     temp[4][0] = float(weekday_PM_in_10.sum()) / 5
122     temp[4][1] = float(weekday_PM_out_10.sum()) / 5
123
124     weekday_PM = []
125     for j in range(5):
126         weekday_PM.append(temp[j][0] + temp [j][1])
127     weekday_PM_Max = max(weekday_PM)
128     tempind = weekday_PM.index(weekday_PM_Max)
129     weekday_PM_in = temp[tempind][0]
130     weekday_PM_out = temp[tempind][1]
131
132     # ----------------- weekend data goes here --------------------------------
133
134     weekend_allday_in = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
        End') \
135                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
        (Su-Su)')) \
136                     & (df["Day Part"] == '00: All Day (12am-12am)'), ["Average Daily Zone Traffic
        (StL Volume)"]]
137
138     weekend_allday_out = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
        Start') \
139                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
        (Su-Su)')) \
140                     & (df["Day Part"] == '00: All Day (12am-12am)'), ["Average Daily Zone Traffic
        (StL Volume)"]]
141
142     weekend_AM_in_6 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
        End') \
143                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
        (Su-Su)')) \
144                     & (df["Day Part"] == '07: 6am (6am-7am)'), ["Average Daily Zone Traffic (StL
        Volume)"]]
```

```
145    weekend_AM_in_7 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       End') \
146                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
147                     & (df["Day Part"] == '08: 7am (7am-8am)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
148    weekend_AM_in_8 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       End') \
149                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
150                     & (df["Day Part"] == '09: 8am (8am-9am)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
151    weekend_AM_in_9 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       End') \
152                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
153                     & (df["Day Part"] == '10: 9am (9am-10am)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
154    weekend_AM_in_10 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       End') \
155                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
156                     & (df["Day Part"] == '11: 10am (10am-11am)'), ["Average Daily Zone Traffic
       (StL Volume)"]]
157
158    weekend_AM_out_6 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       Start') \
159                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
160                     & (df["Day Part"] == '07: 6am (6am-7am)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
161    weekend_AM_out_7 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       Start') \
162                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
163                     & (df["Day Part"] == '08: 7am (7am-8am)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
164    weekend_AM_out_8 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       Start') \
165                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
166                     & (df["Day Part"] == '09: 8am (8am-9am)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
167    weekend_AM_out_9 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       Start') \
168                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
169                     & (df["Day Part"] == '10: 9am (9am-10am)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
170    weekend_AM_out_10 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       Start') \
171                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
```

```
172                    & (df["Day Part"] == '11: 10am (10am-11am)'), ["Average Daily Zone Traffic
       (StL Volume)"]]
173
174    temp = np.zeros((5, 2))
175    temp[0][0] = float(weekend_AM_in_6.sum()) / 2
176    temp[0][1] = float(weekend_AM_out_6.sum()) / 2
177    temp[1][0] = float(weekend_AM_in_7.sum()) / 2
178    temp[1][1] = float(weekend_AM_out_7.sum()) / 2
179    temp[2][0] = float(weekend_AM_in_8.sum()) / 2
180    temp[2][1] = float(weekend_AM_out_8.sum()) / 2
181    temp[3][0] = float(weekend_AM_in_9.sum()) / 2
182    temp[3][1] = float(weekend_AM_out_9.sum()) / 2
183    temp[4][0] = float(weekend_AM_in_10.sum()) / 2
184    temp[4][1] = float(weekend_AM_out_10.sum()) / 2
185
186    weekend_AM = []
187    for j in range(5):
188        weekend_AM.append(temp[j][0] + temp[j][1])
189    weekend_AM_Max = max(weekend_AM)
190    tempind = weekend_AM.index(weekend_AM_Max)
191    weekend_AM_in = temp[tempind][0]
192    weekend_AM_out = temp[tempind][1]
193
194    weekend_PM_in_6 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       End') \
195                    & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
196                    & (df["Day Part"] == '16: 3pm (3pm-4pm)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
197    weekend_PM_in_7 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       End') \
198                    & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
199                    & (df["Day Part"] == '17: 4pm (4pm-5pm)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
200    weekend_PM_in_8 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       End') \
201                    & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
202                    & (df["Day Part"] == '18: 5pm (5pm-6pm)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
203    weekend_PM_in_9 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       End') \
204                    & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
205                    & (df["Day Part"] == '19: 6pm (6pm-7pm)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
206    weekend_PM_in_10 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
       End') \
207                    & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
       (Su-Su)')) \
208                    & (df["Day Part"] == '20: 7pm (7pm-8pm)'), ["Average Daily Zone Traffic (StL
       Volume)"]]
```

```
209
210     weekend_PM_out_6 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
        Start') \
211                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
        (Su-Su)')) \
212                     & (df["Day Part"] == '16: 3pm (3pm-4pm)'), ["Average Daily Zone Traffic (StL
        Volume)"]]
213     weekend_PM_out_7 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
        Start') \
214                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
        (Su-Su)')) \
215                     & (df["Day Part"] == '17: 4pm (4pm-5pm)'), ["Average Daily Zone Traffic (StL
        Volume)"]]
216     weekend_PM_out_8 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
        Start') \
217                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
        (Su-Su)')) \
218                     & (df["Day Part"] == '18: 5pm (5pm-6pm)'), ["Average Daily Zone Traffic (StL
        Volume)"]]
219     weekend_PM_out_9 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
        Start') \
220                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
        (Su-Su)')) \
221                     & (df["Day Part"] == '19: 6pm (6pm-7pm)'), ["Average Daily Zone Traffic (StL
        Volume)"]]
222     weekend_PM_out_10 = df.loc[(df["Zone ID"] == ZoneID[i]) & (df["Intersection Type"] == 'Trip
        Start') \
223                     & ((df["Day Type"] == '6: Saturday (Sa-Sa)') | (df["Day Type"] == '7: Sunday
        (Su-Su)')) \
224                     & (df["Day Part"] == '20: 7pm (7pm-8pm)'), ["Average Daily Zone Traffic (StL
        Volume)"]]
225
226     temp = np.zeros((5, 2))
227     temp[0][0] = float(weekend_PM_in_6.sum()) / 2
228     temp[0][1] = float(weekend_PM_out_6.sum()) / 2
229     temp[1][0] = float(weekend_PM_in_7.sum()) / 2
230     temp[1][1] = float(weekend_PM_out_7.sum()) / 2
231     temp[2][0] = float(weekend_PM_in_8.sum()) / 2
232     temp[2][1] = float(weekend_PM_out_8.sum()) / 2
233     temp[3][0] = float(weekend_PM_in_9.sum()) / 2
234     temp[3][1] = float(weekend_PM_out_9.sum()) / 2
235     temp[4][0] = float(weekend_PM_in_10.sum()) / 2
236     temp[4][1] = float(weekend_PM_out_10.sum()) / 2
237
238     weekend_PM = []
239     for j in range(5):
240         weekend_PM.append(temp[j][0] + temp[j][1])
241     weekend_PM_Max = max(weekend_PM)
242     tempind = weekend_PM.index(weekend_PM_Max)
243     weekend_PM_in = temp[tempind][0]
244     weekend_PM_out = temp[tempind][1]
245
246     '''
```

```
247     res[i][0] = ZoneID[i]
248     res[i][1] = float(weekday_allday_in.sum()) / 5
249     res[i][2] = float(weekday_allday_out.sum()) / 5
250     res[i][3] = weekday_AM_in
251     res[i][4] = weekday_AM_out
252     res[i][5] = weekday_PM_in
253     res[i][6] = weekday_PM_out
254     res[i][7] = float(weekend_allday_in.sum()) / 2
255     res[i][8] = float(weekend_allday_out.sum()) / 2
256     res[i][9] = weekend_AM_in
257     res[i][10] = weekend_AM_out
258     res[i][11] = weekend_PM_in
259     res[i][12] = weekend_PM_out
260     '''
261     res[i][0] = ZoneID[i]
262     res[i][1] = float(weekday_allday_in.sum()) / 5 + float(weekday_allday_out.sum()) / 5
263     res[i][2] = weekday_AM_in + weekday_AM_out
264     res[i][3] = weekday_PM_in + weekday_PM_out
265     res[i][4] = float(weekend_allday_in.sum()) / 2 + float(weekend_allday_out.sum()) / 2
266     res[i][5] = weekend_AM_in + weekend_AM_out
267     res[i][6] = weekend_PM_in + weekend_PM_out
268
269     print(str(i))
270
271 res1 = np.around(res).astype(int)
272
273 np.savetxt('C:/Users/Brandon Loesch/Desktop/project/Sept_2021_Results_test.csv', res1,
        delimiter=',', header = "Zone_ID, Wday_All, \
274     Wday_AM_All, Wday_PM_All, Wend_All, Wend_AM_All, Wend_PM_All", fmt=
        '%d',comments='')
```

Appendix B. Model Fitting Script

```
1    import arcpy
2    import numpy as np
3    import pandas as pd
4    import matplotlib.pyplot as plt
5    import csv
6    import os
7    import statsmodels.api as sm
8
9    #datapath must match exactly to your file structure, explained in report
10   datapath = "C:/Users/Brandon Loesch/Desktop/project/For Brandon 2-22-2023/For Brandon 2-22-
     2023/"
11   fc=datapath+"Francesco_11_3_2022_PRJ.shp"
12
13   # Read Data
14   #fieldname = "ITECode"
15   delimfield_1 = arcpy.AddFieldDelimiters(fc, "ITECode")
16   #fieldname = "Comments"
17   #delimfield_2 = arcpy.AddFieldDelimiters(fc, "Comments")
18   delimfield_2 = arcpy.AddFieldDelimiters(fc, "Reviewed")
19   LUCode = [140, 150, 155, 221, 710, 760, 820, 850, 960, 813, 815]
20   #LUCode = [760, 813, 815]
21
22   # specify ArcGIS Pro SQL
23   sqlwhere={}
24   LUName={}
25   for i in range(len(LUCode)-2):
26      sqlwhere[i] = delimfield_1 + " ='" + str(LUCode[i]) + "' And "
27      LUName[i] = "ITECode = "+ str(LUCode[i])
28    #  sqlwhere[i] = sqlwhere [i]+ delimfield_2 + " = ' '"
29      sqlwhere[i] = sqlwhere [i]+ delimfield_2 + " = 'y'"
30
31
32   sqlwhere[len(LUCode)-2] = "(" + delimfield_1 + " = '813' Or " + delimfield_1 + " = '815') And "
33   LUName[len(LUCode)-2] = "ITECode = 813 Or ITECode = 815"
34   #sqlwhere[len(LUCode)-2] = sqlwhere[len(LUCode)-2] + delimfield_2 + " = ' '"
35   sqlwhere[len(LUCode)-2] = sqlwhere[len(LUCode)-2] + delimfield_2 + " = 'y'"
36
37   # read trips from .csv file, only 1 dataname should be active at a time, use a "#" to deactivate the
     other files
38   dataname = "Jan_2022_Results"
39   #dataname = "Sept_2021_Results"
40   #dataname = "2018_Results"
41
42   f_trip = datapath + dataname + ".csv"
43   with open(f_trip, 'r') as ft:
44      reader = csv.reader(ft, delimiter=',')
45      headers = next(reader)
46      ydata = np.array(list(reader)).astype(int)
47
48   # create an empty array (# of land use code x 6 models) to save rsquared results
```

```
49    my_rows, my_cols = (len(LUName), 6+1)  # last column saves sample size
50    R2_Res = [[0.00]*my_cols]*my_rows
51    R2_Res = np.array(R2_Res)
52
53    # output file path for modeling results
54    write_path = datapath + "ModelResults/" + dataname + "__modelsummary.csv"
55    fout = open(write_path,'w')
56
57    # read data from shapefile and do regression
58    for k in range(len(sqlwhere)):
59      invar = []
60      with arcpy.da.SearchCursor(fc, ["id", "TotalBLD_A", "TotalRES_A", "Worker_Den",
      "Pop_Den"], sqlwhere[k]) as cursor:
61        num_row=0
62        for row in cursor:
63          invar = invar + list(row)
64          num_row+=1
65
66      invar = list(map(int, invar))
67      invar = np.array(invar)
68      invar = invar.reshape(num_row, 5)
69
70      # Fit Model
71      x = invar[:,2]
72      x = x/1000
73      x_t = np.transpose(x)
74      x = sm.add_constant(x)
75
76      ModelName = ["Weekday",
77          "Weekday AM",
78          "Weekday PM",
79          "Weekend",
80          "Weekend AM",
81          "Weekend PM"]
82
83      # read trip data
84      ZoneID = invar[:,0]
85      R2_Res[k,my_cols-1] = len(ZoneID)
86      yvar = []
87
88      for i in ZoneID:
89        for j in ydata:
90          if j[0] == i:
91            yvar = yvar + list(j)
92
93      yvar = np.array(yvar)
94      yvar = yvar.reshape(len(ZoneID), 7)
95
96      #set up plots
97
98      fig, axes = plt.subplots(2, 3, figsize=(15, 10))
99      fig.suptitle("Results for " + LUName[k])
100     plt.setp(axes[:,:], xlabel='Area in 1000 sqft')
```

72

```
101      plt.setp(axes[:,:], ylabel='# of trips')
102      plt.subplots_adjust(wspace=0.33,hspace=0.33)
103
104      #set up empty coefficient, intercept table
105
106      equation_table = np.empty([6,4])
107      equation_table_arrange = np.empty([1,4])
108
109      for j in range(6):
110         y=yvar[:,j+1]
111         model = sm.OLS(y, x)
112         res = model.fit()
113
114         #grab coefficients, intercepts and p-values and append to empty numpy array
115         results = res.params
116         pvalues = res.pvalues
117         res_err = np.hstack((results,pvalues))
118         equation_table[j] = np.around(res_err,2)
119
120         R2_Res[k,j]=res.rsquared
121         fout.write("\n\n\n\n This is the modeling results for " + LUName[k] + " ------" + ModelName[j]
     + "\n\n\n")
122         #   f.write(res.summary().as_csv())
123         fout.write(res.summary().as_csv())
124         zonelb=invar[:,0]  # zone label -- zone ID
125         p1 = int(j/3)
126         p2 = j%3
127         axes[p1,p2].scatter(x_t, y)
128         axes[p1,p2].set_title(ModelName[j])
129
130      #create folder structure for equation csvs and save to that path
131      eqfolder = datapath + "ModelResults/" + dataname + '_equations'
132      if not os.path.exists(eqfolder):
133         os.makedirs(eqfolder)
134      #saving model equation results to csv, both clean and uncleaned version
135      write_path_res = eqfolder + '/' + LUName[k] + '__' + 'equation.csv'
136      with open(write_path_res, "w", newline='') as csvfile:
137         wri = csv.writer(csvfile, delimiter=',')
138         wri.writerow([dataname + ' - ' + LUName[k]])
139         wri.writerow([' ','INTERCEPT',' ','SLOPE',' '])
140         wri.writerow(['Model Name', 'Coefficient', 'P-Value', 'Coefficient', 'P-Value'])
141         for i in range(len(ModelName)):
142            wri.writerow([ModelName[i], equation_table[i,0], equation_table[i,2], equation_table[i,1],
     equation_table[i,3]])
143         wri.writerow([' '])
144         wri.writerow(['P-Values considered in equation'])
145         wri.writerow([dataname + ' - ' + LUName[k]])
146         wri.writerow([' ','INTERCEPT',' ','SLOPE',' '])
147         wri.writerow(['Model Name', 'Coefficient', 'P-Value', 'Coefficient', 'P-Value'])
148         for i in range(len(ModelName)):
149            if equation_table[i,2] and equation_table[i,3] >= 0.1:
150               wri.writerow([ModelName[i], 0, equation_table[i,2], 0, equation_table[i,3]])
151            elif equation_table[i,2] >= 0.1:
```

73

```
152            wri.writerow([ModelName[i], 0, equation_table[i,2], equation_table[i,1],
      equation_table[i,3]])
153         elif equation_table[i,3] >= 0.1:
154            wri.writerow([ModelName[i], equation_table[i,0], equation_table[i,2], 0,
      equation_table[i,3]])
155         else:
156            wri.writerow([ModelName[i], equation_table[i,0], equation_table[i,2], equation_table[i,1],
      equation_table[i,3]])
157       wri.writerow(['* - set to zero due to being insignificant, p-value > 0.1.'])
158
159    #create folder structure for figure pngs and save to that path
160    figfolder = datapath + "ModelResults/" + dataname + '_figures'
161    if not os.path.exists(figfolder):
162       os.makedirs(figfolder)
163    plt.savefig(figfolder + "/" + LUName[k] + ".png")
164    plt.close()
165
166
167 fout.close()
168
169 #save r-squared table
170 write_path = datapath + "ModelResults/" + dataname + "__" + "RSquared.csv"
171 R2_Res = np.around(R2_Res, decimals = 2)
172
173 with open(write_path, "w", newline='') as csvfile:
174    wri = csv.writer(csvfile, delimiter=',')
175    wri.writerow(['Land Use Code', 'Sample Size', 'Wday_All', 'Wday_AM_All', 'Wday_PM_All',
      'Wend_All', 'Wend_AM_All', 'Wend_PM_All'])
176    for i in range(len(LUName)):
177       wri.writerow([LUName[i], R2_Res[i,6], R2_Res[i,0], R2_Res[i,1], R2_Res[i,2], R2_Res[i,3],
      R2_Res[i,4], R2_Res[i,5]])
178
179 print("Complete!")
180
```

Appendix C. Trip vs. Square Footage Figures (January 2022)
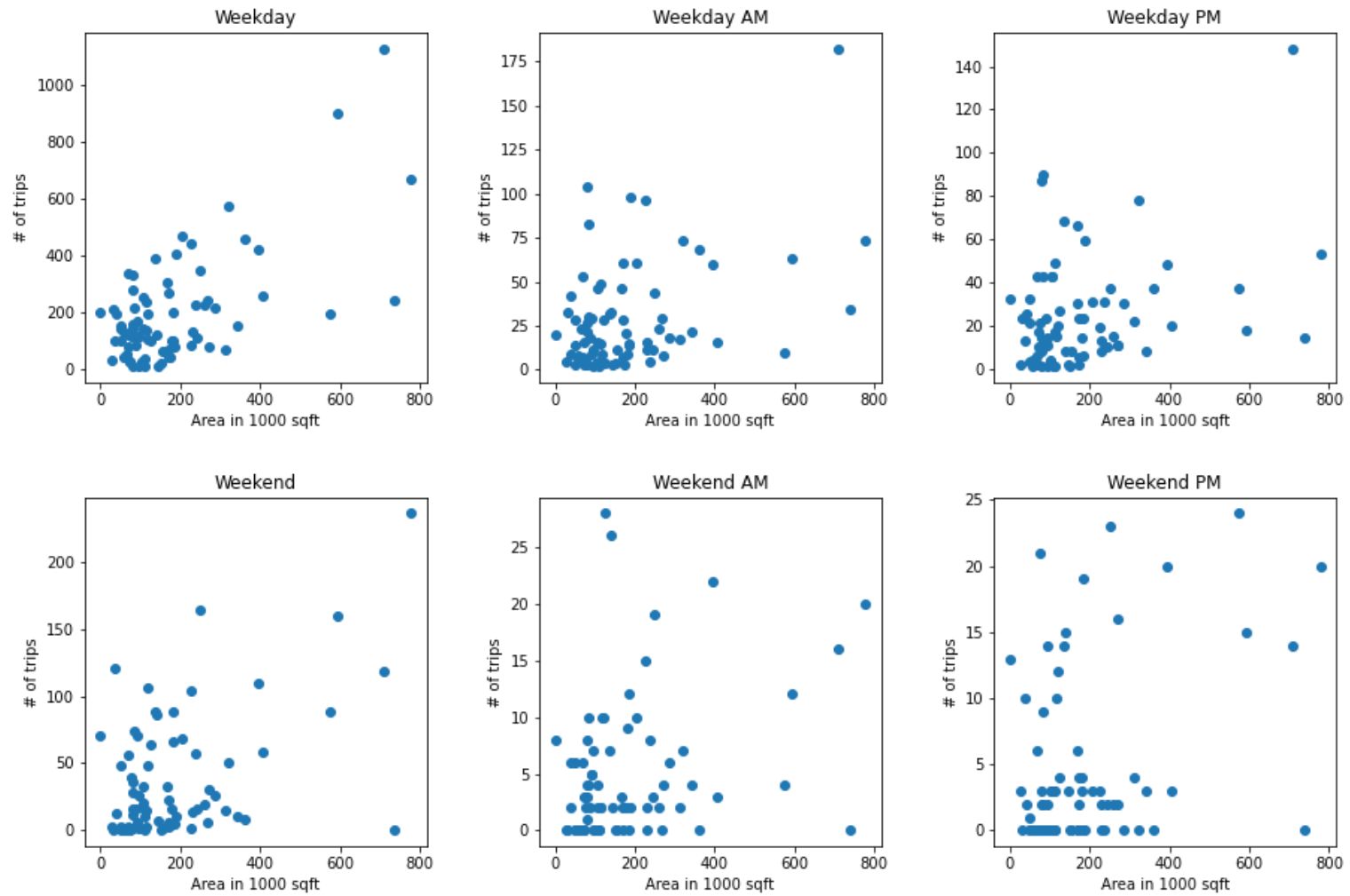
Figure 8.1 Trip and area data for ITECode=140
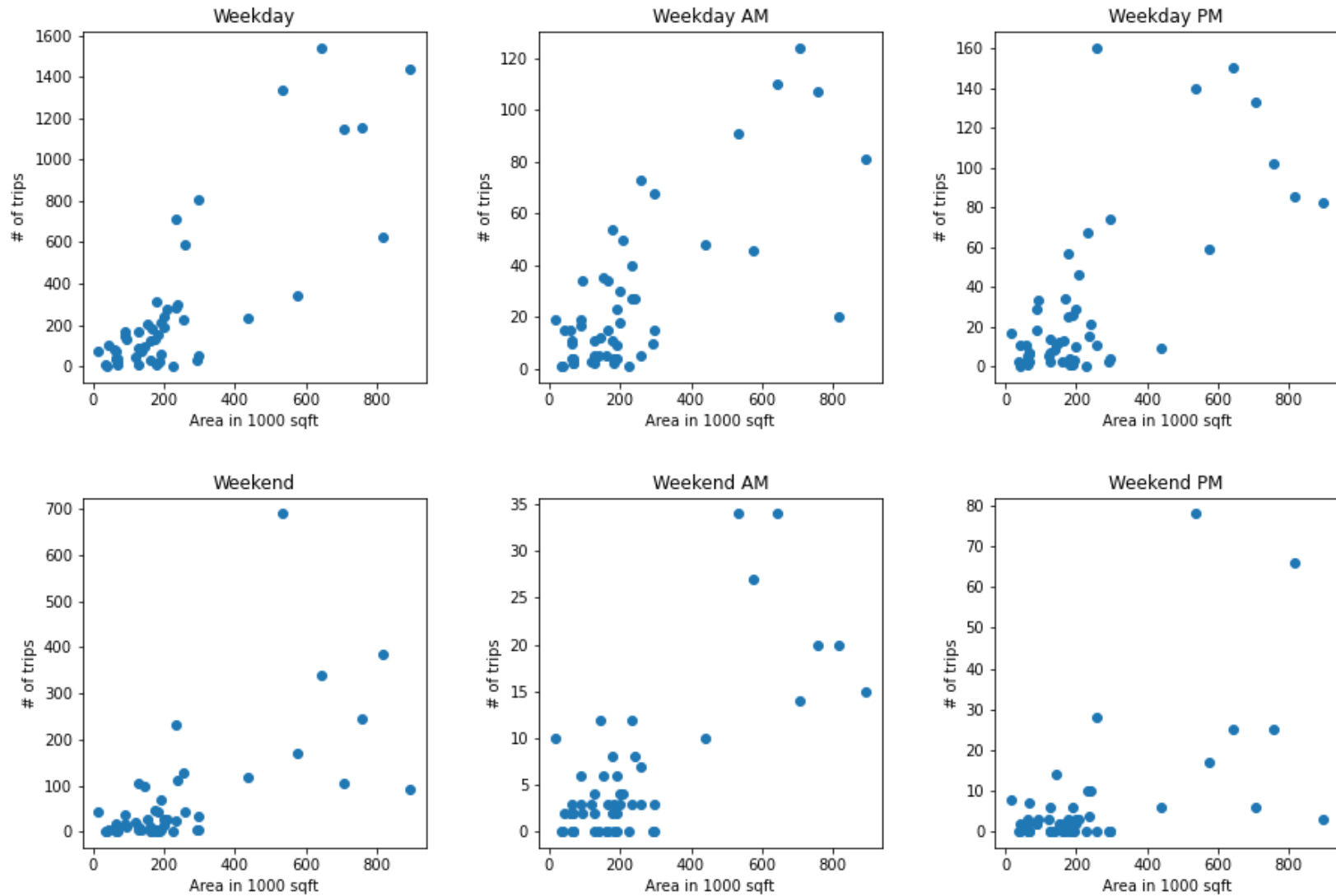
Figure 8.2: Trip and area data for ITECode=150
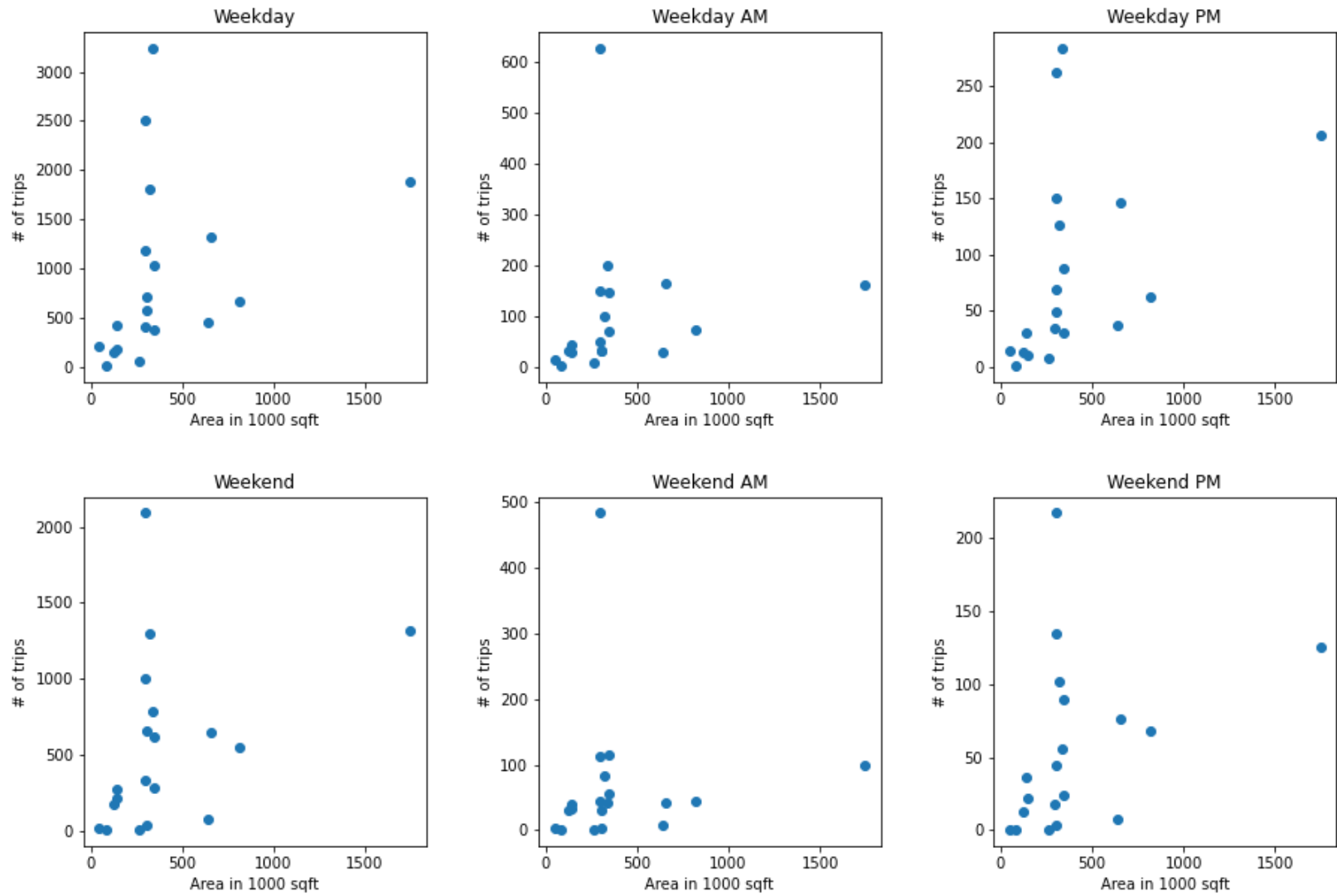
Figure 8.3: Trip and area data for ITECode=155
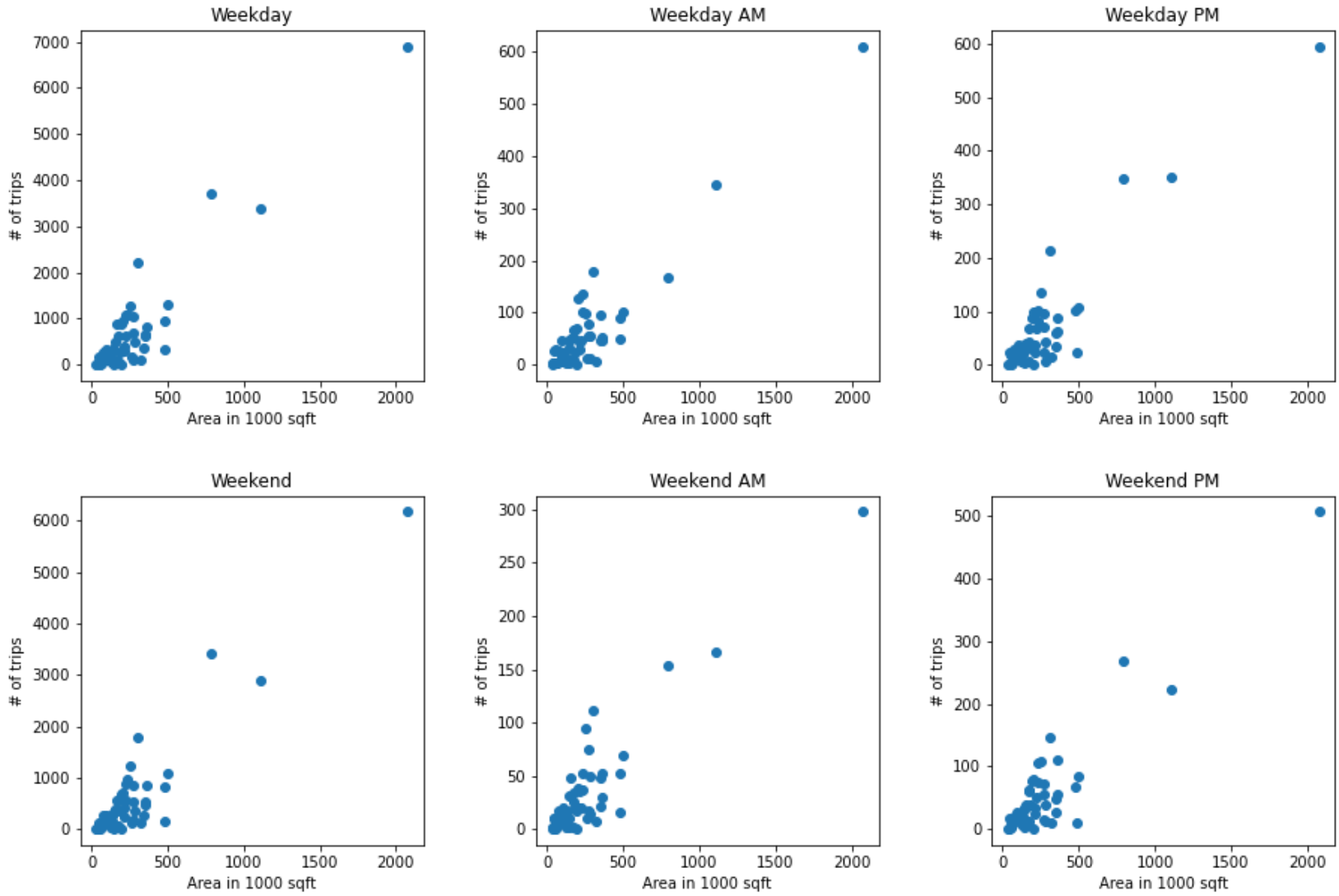
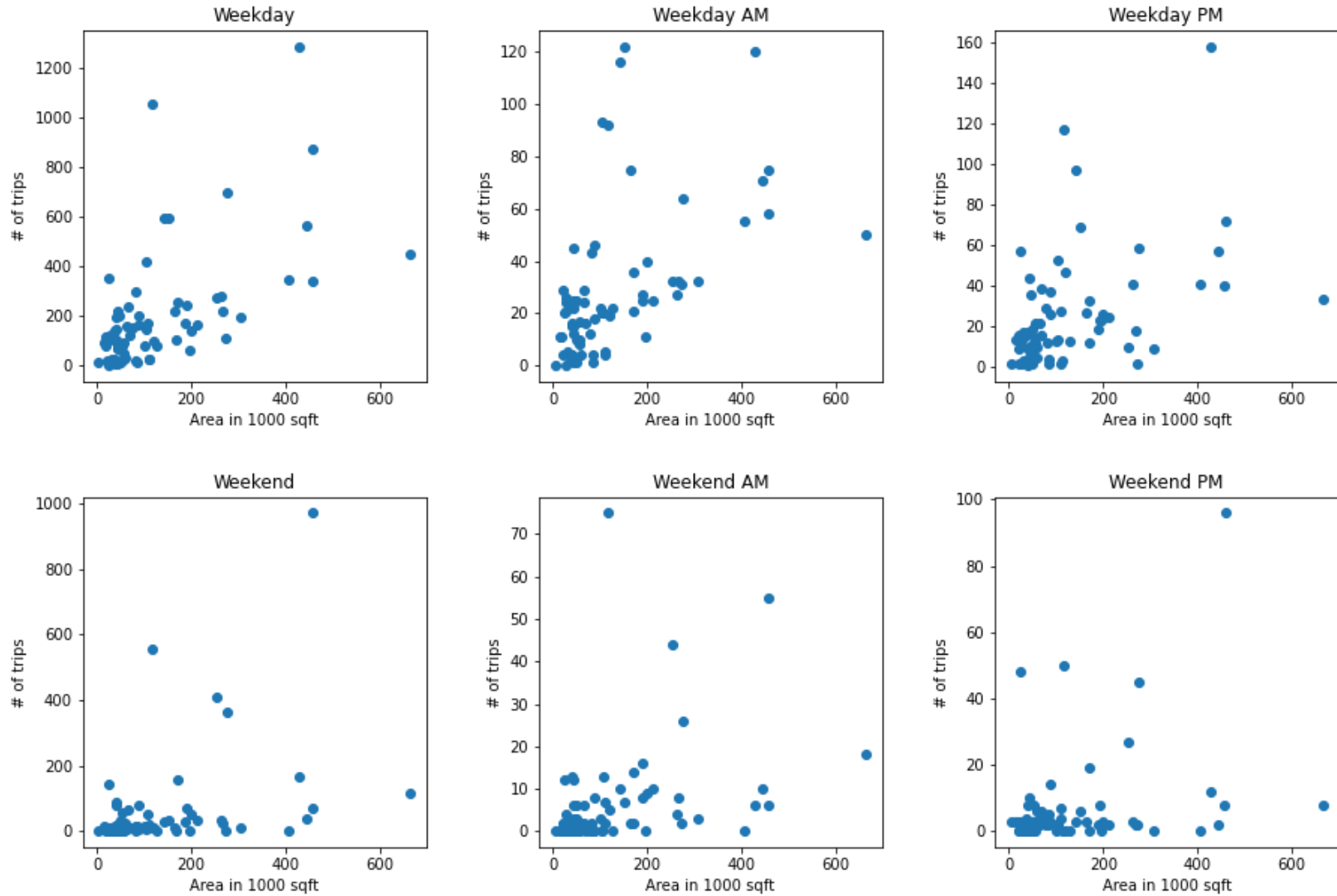**Figure 8.4: Trip and area data for ITECode=221**
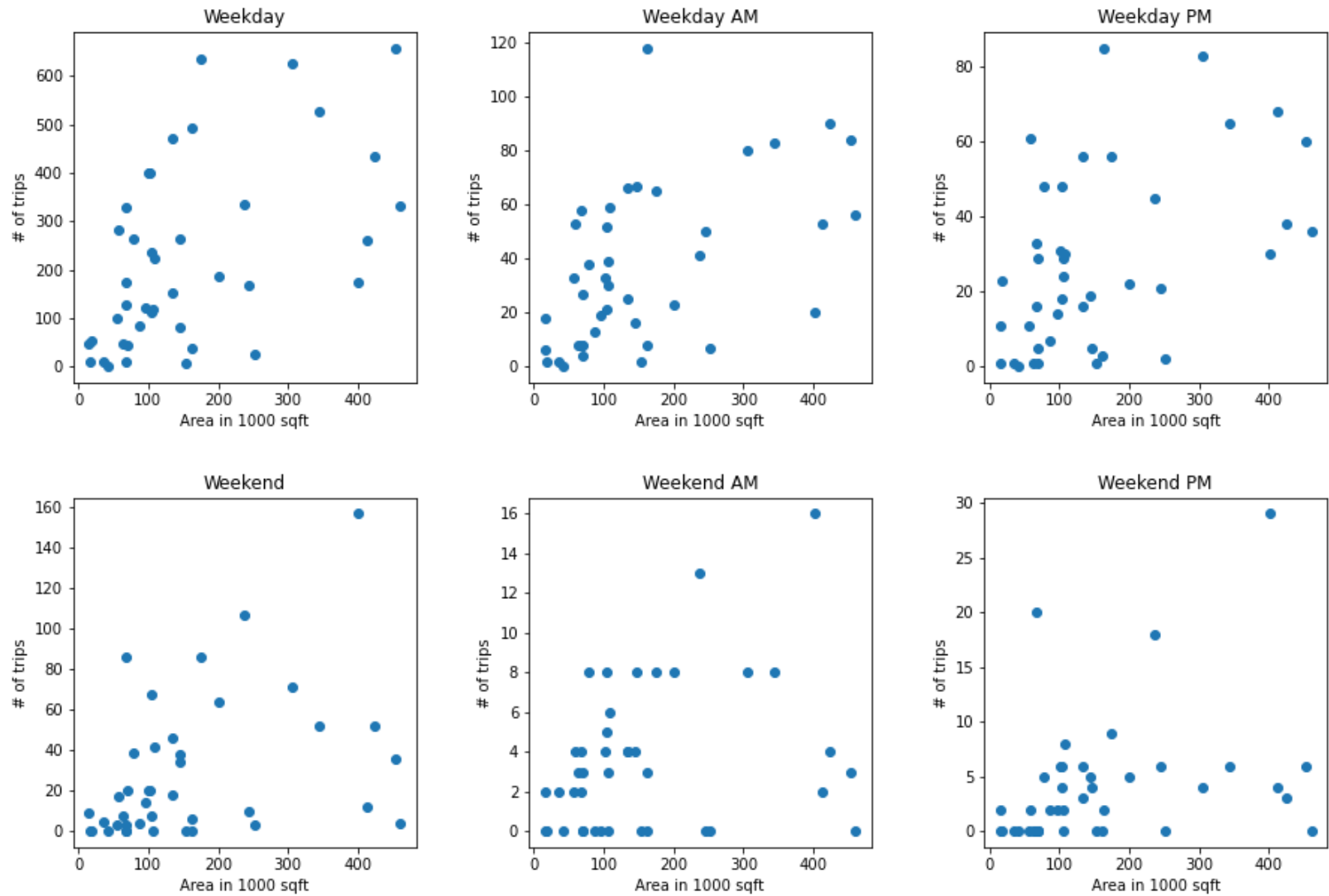
Figure 8.5: Trip and area data for ITECode=710
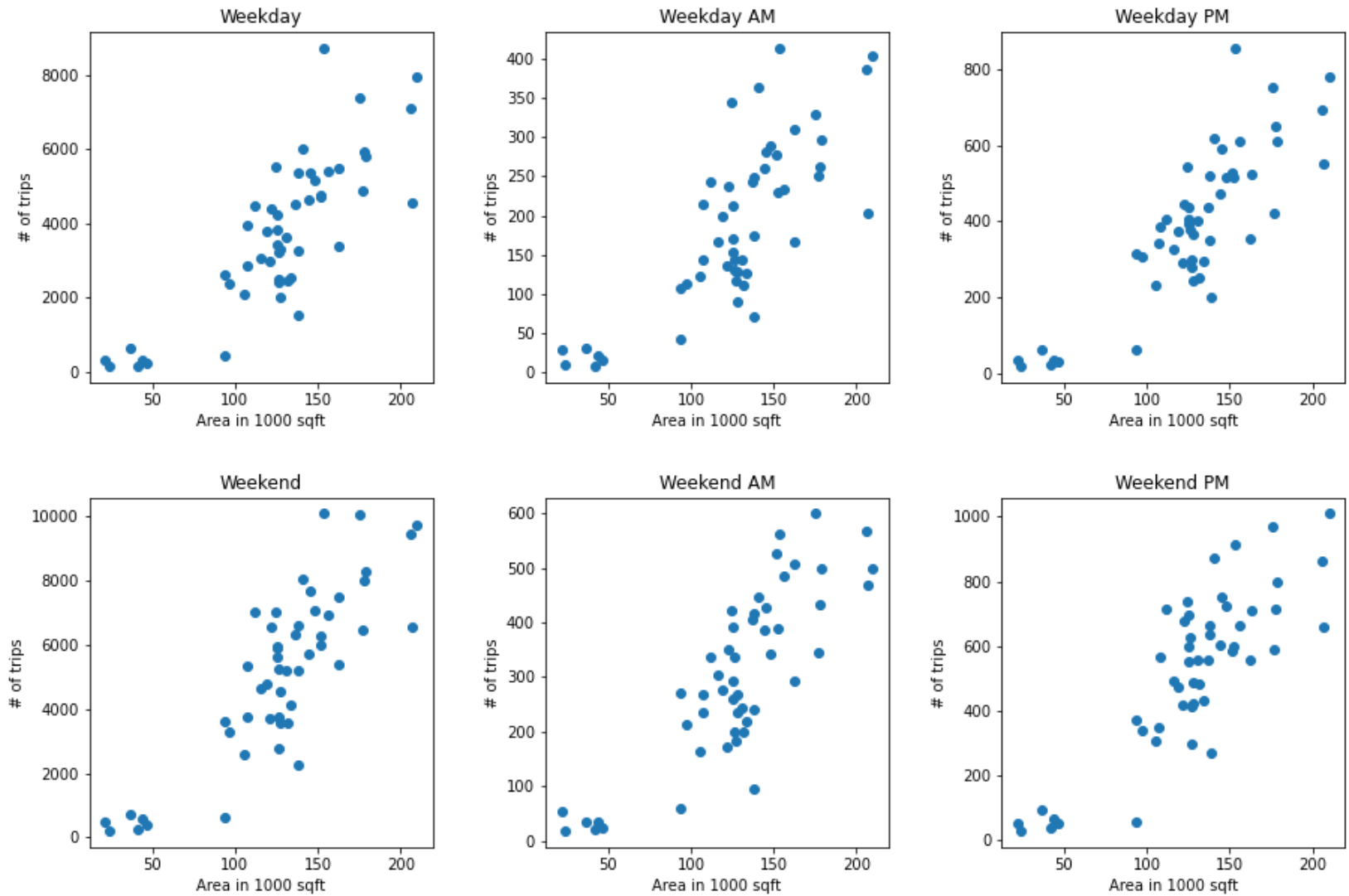
Figure 8.6: Trip and area data for ITECode=760
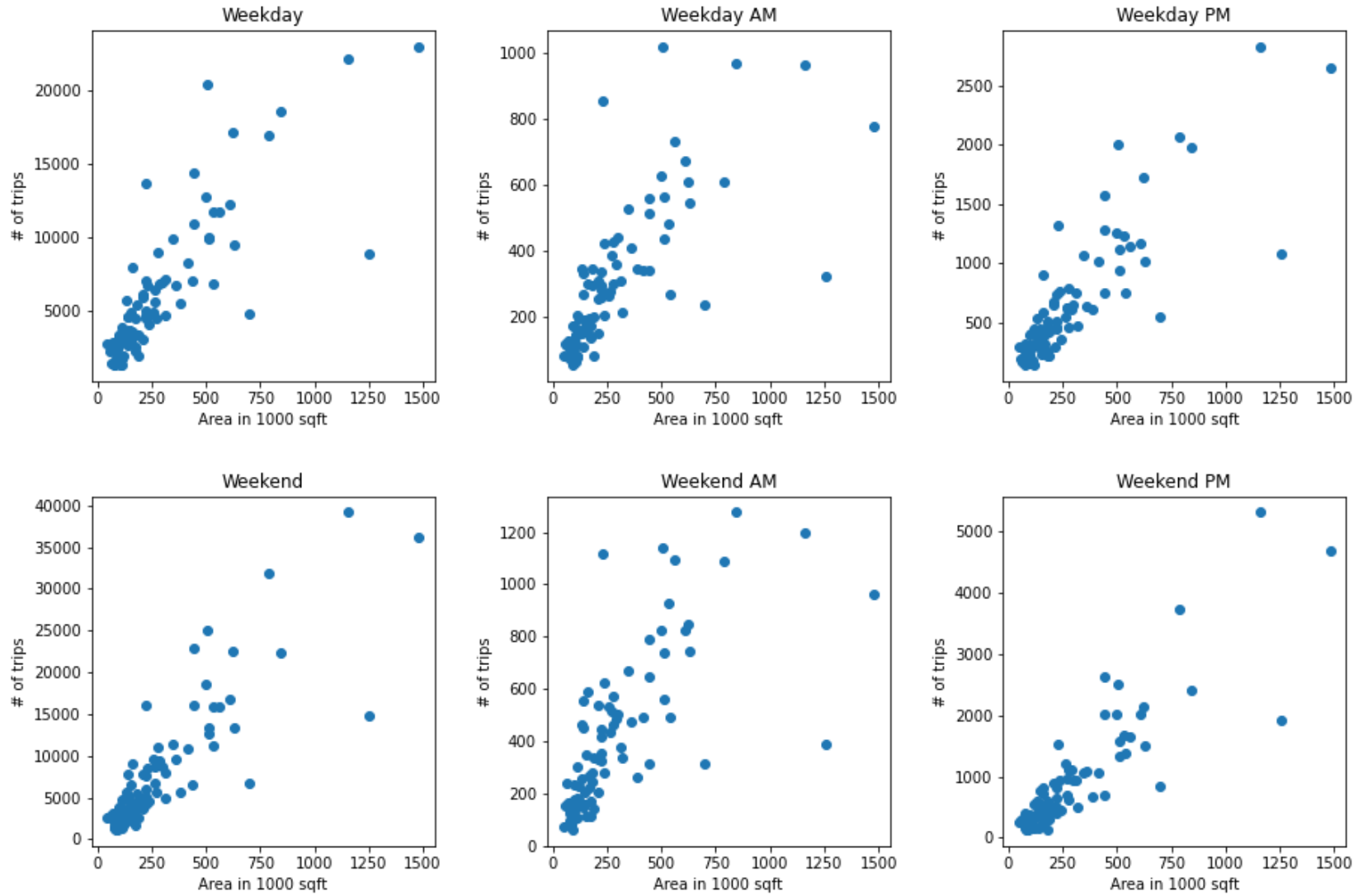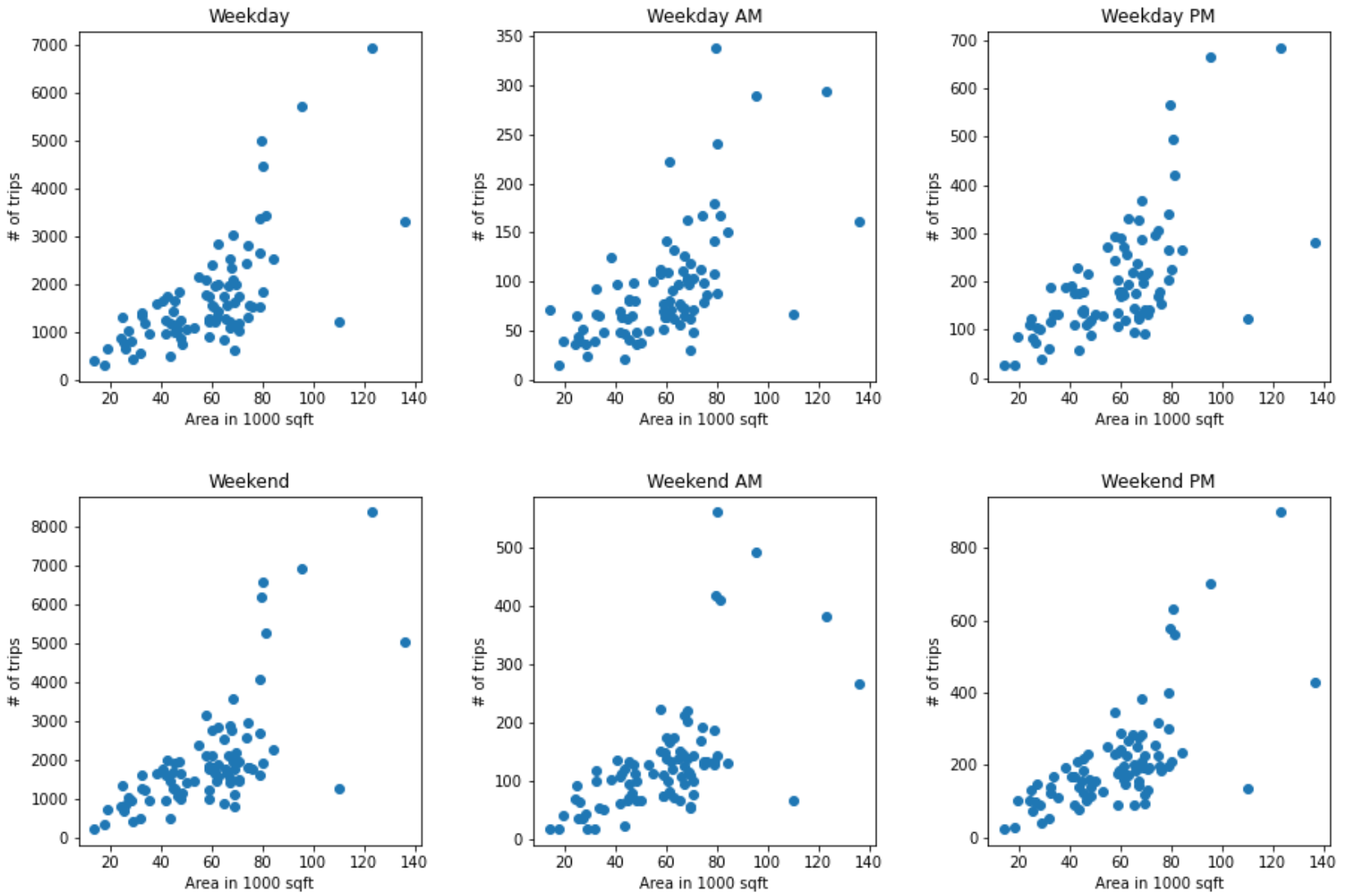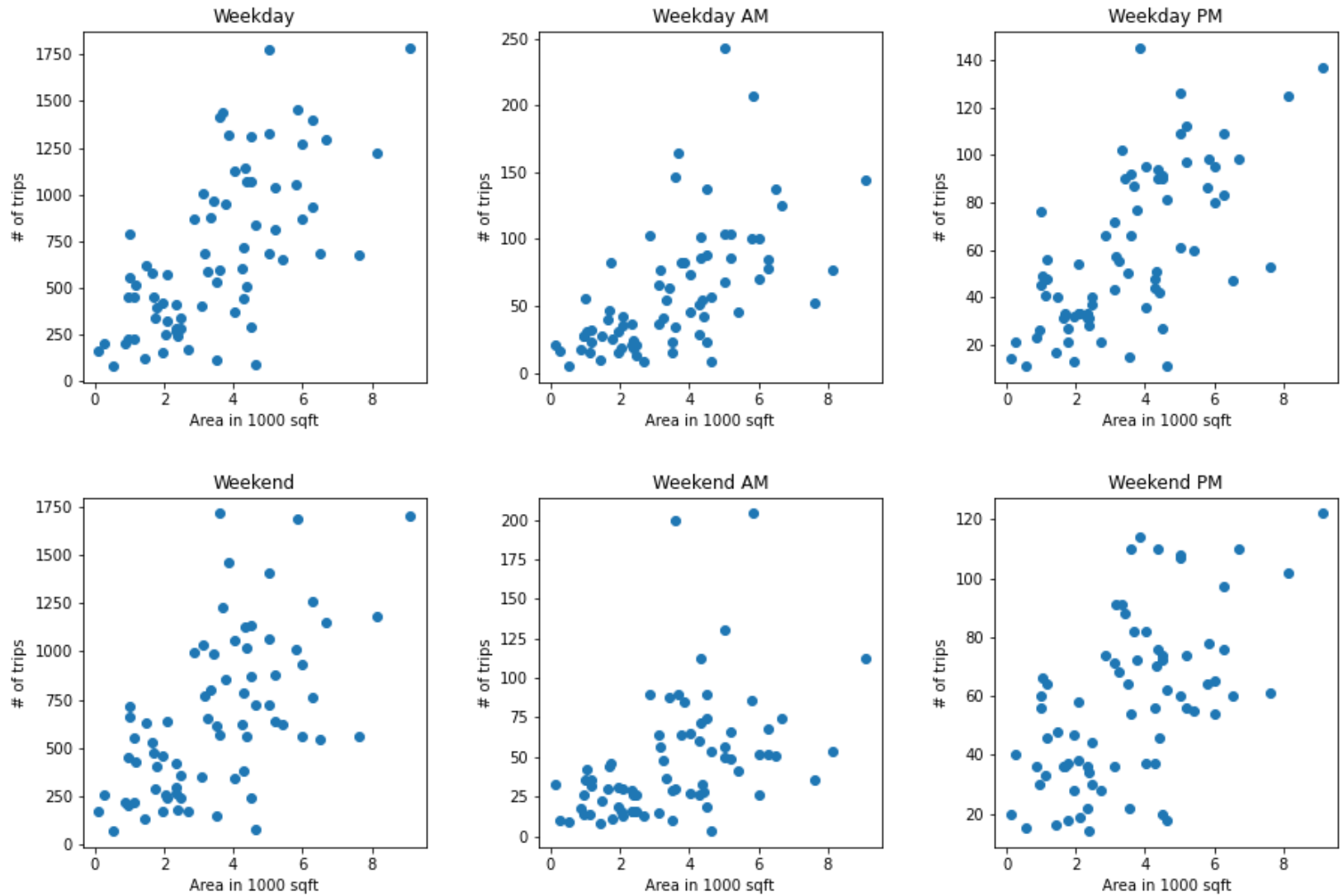
Figure 8.7: Trip and area data for ITECode=813 or 815

Figure 8.8: Trip and area data for ITECode=820

Figure 8.9: Trip and area data for ITECode=850

Figure 8.10: Trip and area data for ITECode=960

Appendix D. Equation Tables for Dataset 3 (January 2022)

Table 8.1: Results for ITE Code=150 and Dataset 3

Model Name	INTERCEPT Coefficient	INTERCEPT P-Value	SLOPE Coefficient	SLOPE P-Value
Weekday	0.00*	0.11	1.54	0.00
Weekday AM	0.00*	0.69	0.11	0.00
Weekday PM	0.00*	0.74	0.14	0.00
Weekend	0.00*	0.31	0.38	0.00
Weekend AM	0.00*	0.49	0.03	0.00
Weekend PM	0.00*	0.38	0.04	0.00

* - set to zero due to being insignificant, p-value > 0.1.

Table 8.2: Results for ITE Code=155 and Dataset 3

Model Name	INTERCEPT Coefficient	INTERCEPT P-Value	SLOPE Coefficient	SLOPE P-Value
Weekday	589.51	0.06	0.00*	0.15
Weekday AM	0.00*	0.11	0.00*	0.53
Weekday PM	48.51	0.10	0.09	0.08
Weekend	330.59	0.08	0.00*	0.12
Weekend AM	0.00*	0.13	0.00*	0.76
Weekend PM	32.68	0.10	0.00*	0.13

* - set to zero due to being insignificant, p-value > 0.1.

Table 8.3: Results for ITE Code=221 and Dataset 3

Model Name	INTERCEPT Coefficient	INTERCEPT P-Value	SLOPE Coefficient	SLOPE P-Value
Weekday	-184.76	0.03	3.35	0.00
Weekday AM	-13.36	0.06	0.29	0.00
Weekday PM	0.00*	0.12	0.30	0.00
Weekend	-179.45	0.02	2.99	0.00
Weekend AM	0.00*	0.50	0.15	0.00
Weekend PM	0.00*	0.30	0.24	0.00

* - set to zero due to being insignificant, p-value > 0.1.

Table 8.4: Results for ITE Code=710 and Dataset 3

Model Name	INTERCEPT Coefficient	INTERCEPT P-Value	SLOPE Coefficient	SLOPE P-Value
Weekday	65.16	0.05	1.07	0.00
Weekday AM	13.22	0.00	0.12	0.00
Weekday PM	12.51	0.00	0.1	0.00
Weekend	0.00*	0.94	0.44	0.00
Weekend AM	0.00*	0.29	0.03	0.00
Weekend PM	0.00*	0.34	0.03	0.01

* - set to zero due to being insignificant, p-value > 0.1.

Table 8.5: Results for ITE Code=760 and Dataset 3

Model Name	INTERCEPT Coefficient	INTERCEPT P-Value	SLOPE Coefficient	SLOPE Coefficient
Weekday	94.97	0.02	0.80	0.00
Weekday AM	17.02	0.01	0.12	0.00
Weekday PM	13.79	0.01	0.09	0.00
Weekend	0.00*	0.19	0.11	0.01
Weekend AM	2.19	0.02	0.01	0.05
Weekend PM	0.00*	0.25	0.02	0.04

* - set to zero due to being insignificant, p-value > 0.1.

Table 8.6: Results for ITE Code=813 or 815 and Dataset 3

Model Name	INTERCEPT Coefficient	INTERCEPT P-Value	SLOPE Coefficient	SLOPE P-Value
Weekday	-1230.43	0.02	39.07	0.00
Weekday AM	-55.47	0.07	1.92	0.00
Weekday PM	-113.18	0.02	3.98	0.00
Weekend	-1466.14	0.02	51.83	0.00
Weekend AM	-84.77	0.04	3.04	0.00
Weekend PM	-104.25	0.09	4.93	0.00

Table 8.7: Results for ITE Code=820 and Dataset 3

Model Name	INTERCEPT Coefficient	INTERCEPT P-Value	SLOPE Coefficient	SLOPE P-Value
Weekday	1674.99	0.00	15.36	0.00
Weekday AM	128.17	0.00	0.61	0.00
Weekday PM	150.27	0.00	1.76	0.00
Weekend	0.00*	0.24	25.02	0.00
Weekend AM	177.9	0.00	0.80	0.00
Weekend PM	0.00*	0.88	3.12	0.00

* - set to zero due to being insignificant, p-value > 0.1.

Table 8.8: Results for ITE Code=850 and Dataset 3

Model Name	INTERCEPT Coefficient	INTERCEPT P-Value	SLOPE Coefficient	SLOPE P-Value
Weekday	0.00*	0.31	33.82	0.00
Weekday AM	0.00*	0.81	1.66	0.00
Weekday PM	0.00*	0.83	3.51	0.00
Weekend	-603.2	0.07	44.53	0.00
Weekend AM	0.00*	0.31	2.67	0.00
Weekend PM	0.00*	0.20	4.37	0.00

* - set to zero due to being insignificant, p-value > 0.1.

Table 8.9: Results for ITE Code=960 and Dataset 3

Model Name	INTERCEPT Coefficient	INTERCEPT P-Value	SLOPE Coefficient	SLOPE P-Value
Weekday	171.83	0.03	147.77	0.00
Weekday AM	0.00*	0.28	14.31	0.00
Weekday PM	19.77	0.00	11.38	0.00
Weekend	218.03	0.01	124.78	0.00
Weekend AM	16.25	0.05	8.87	0.00
Weekend PM	28.33	0.00	8.33	0.00

* - set to zero due to being insignificant, p-value > 0.1.

This page left blank intentionally.

Appendix E. Ground Truth Comparison Figures
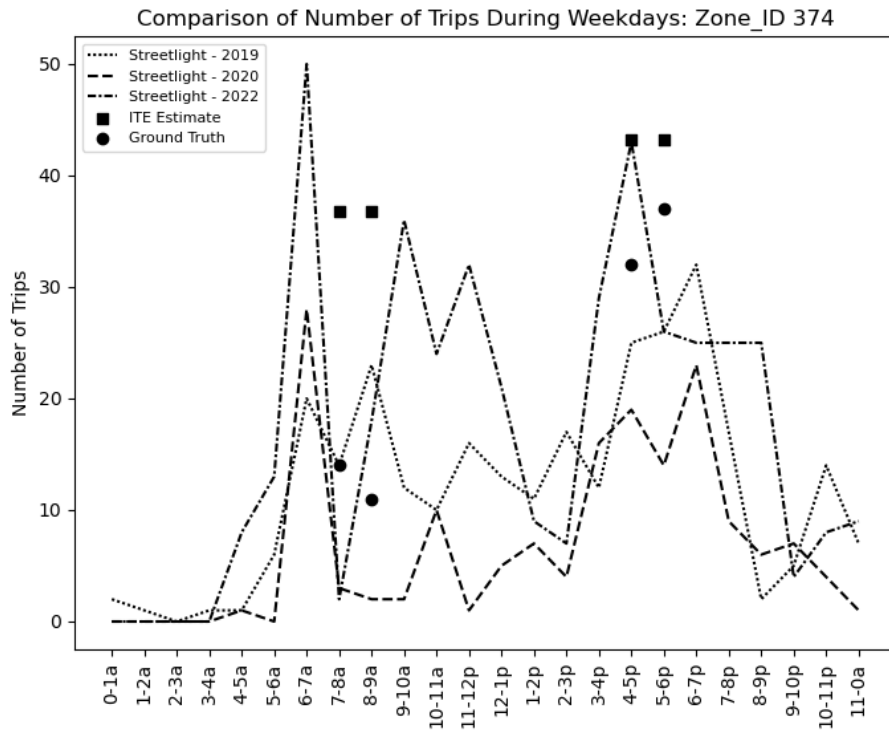
Figure 8.11: Trip comparison for Zone_ID 374
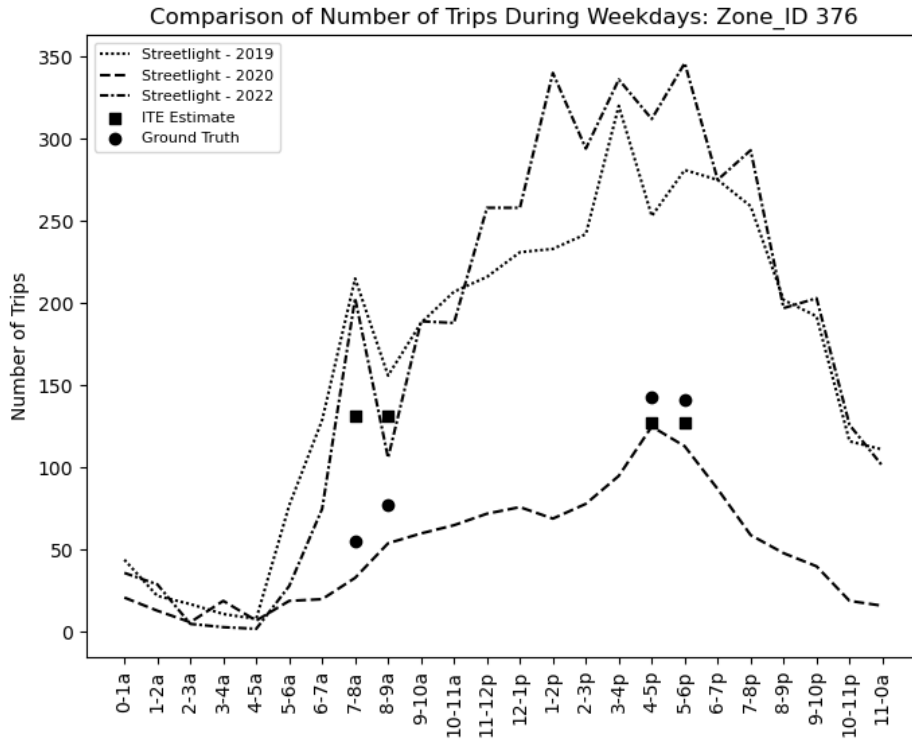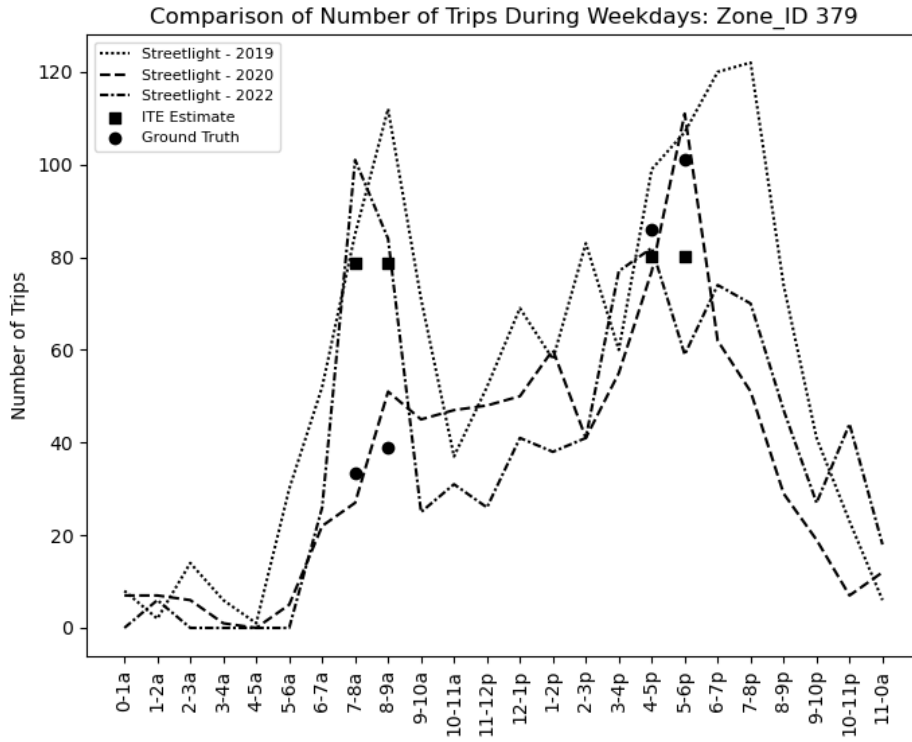
Figure 8.12: Trip comparison for Zone_ID 376

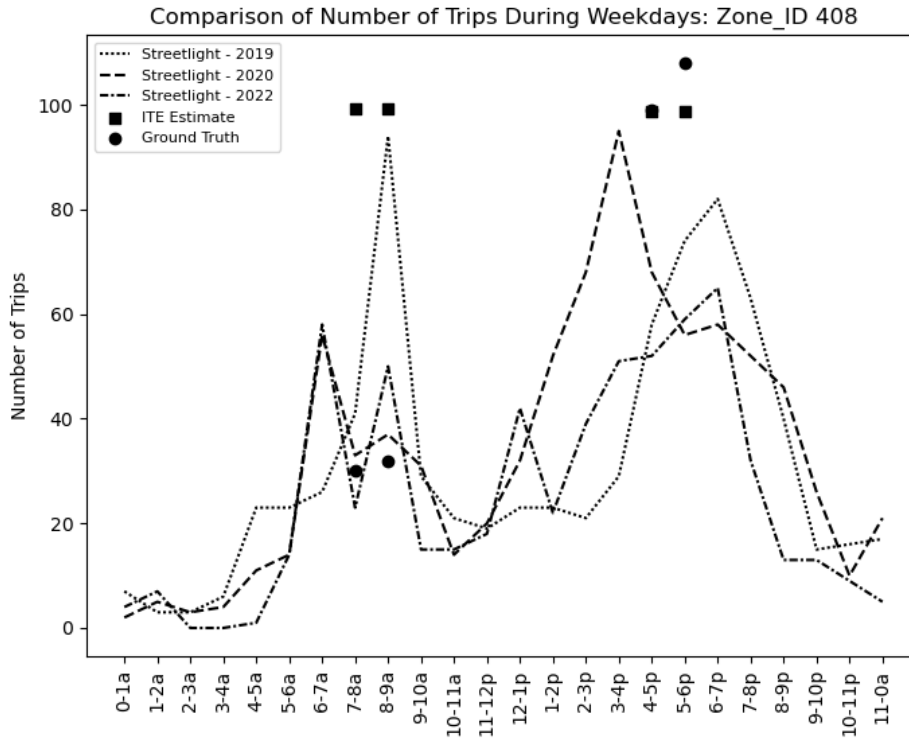**Figure 8.13: Trip comparison for Zone_ID 379**

90

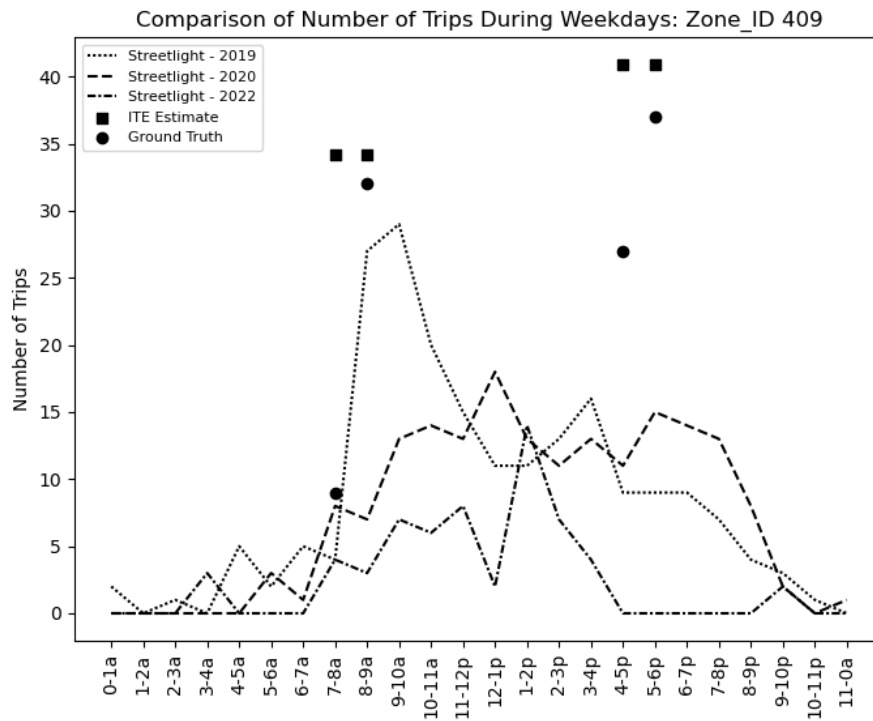**Figure 8.14: Trip comparison for Zone_ID 408**
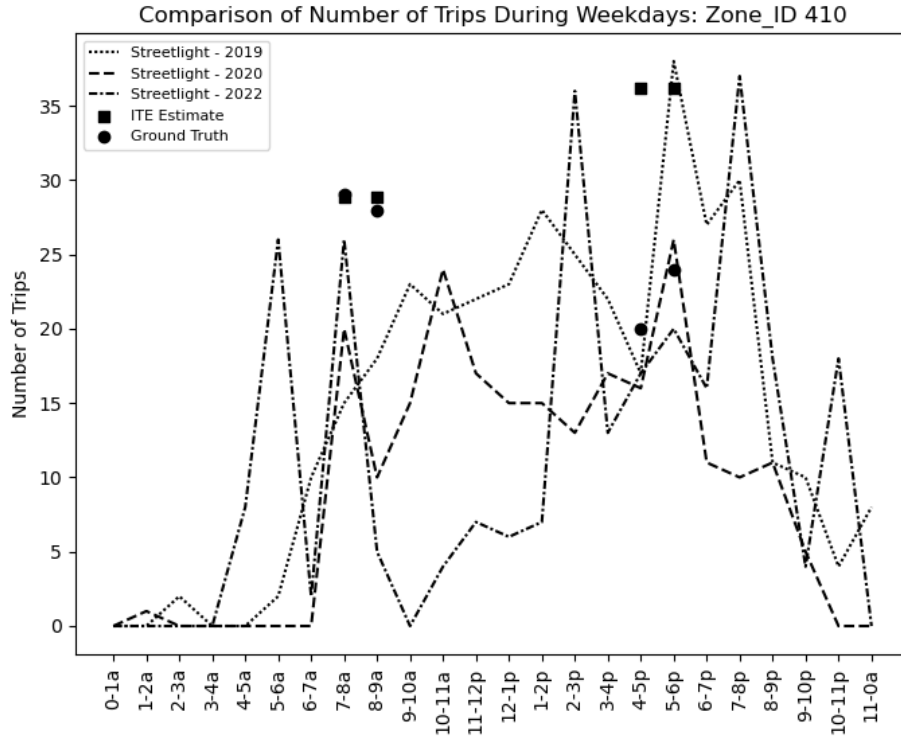

Figure 8.15: Trip comparison for Zone_ID 409
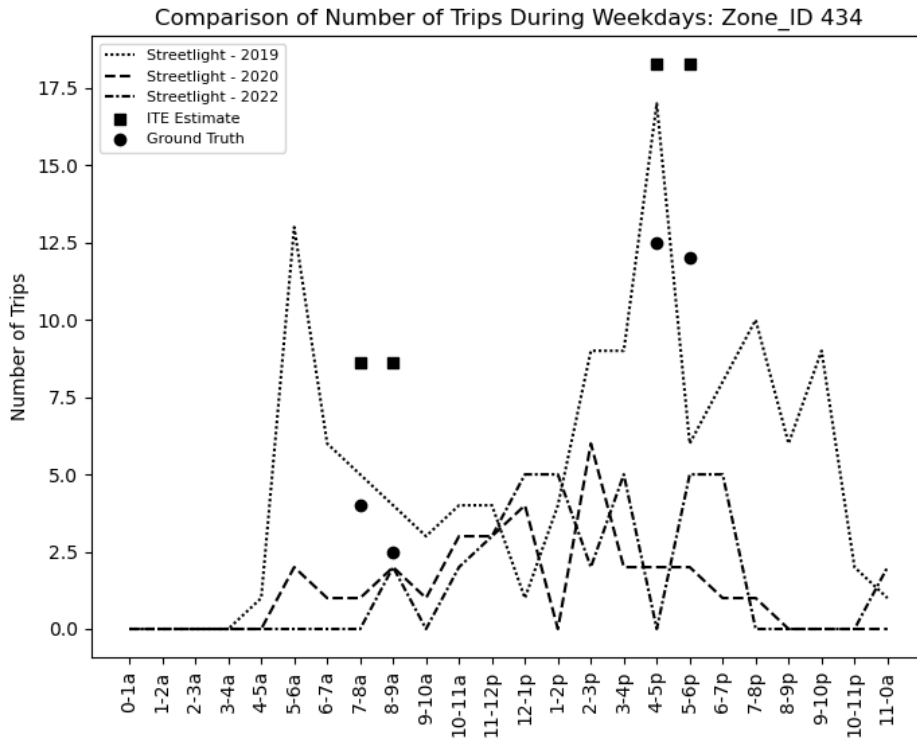
91

Figure 8.16: Trip comparison for Zone_ID 410
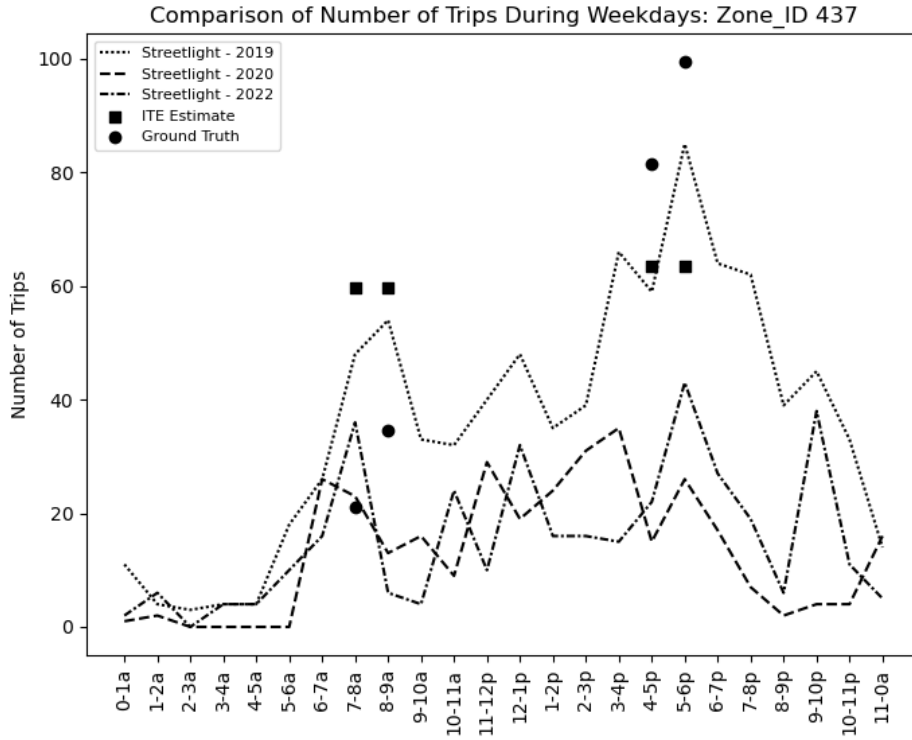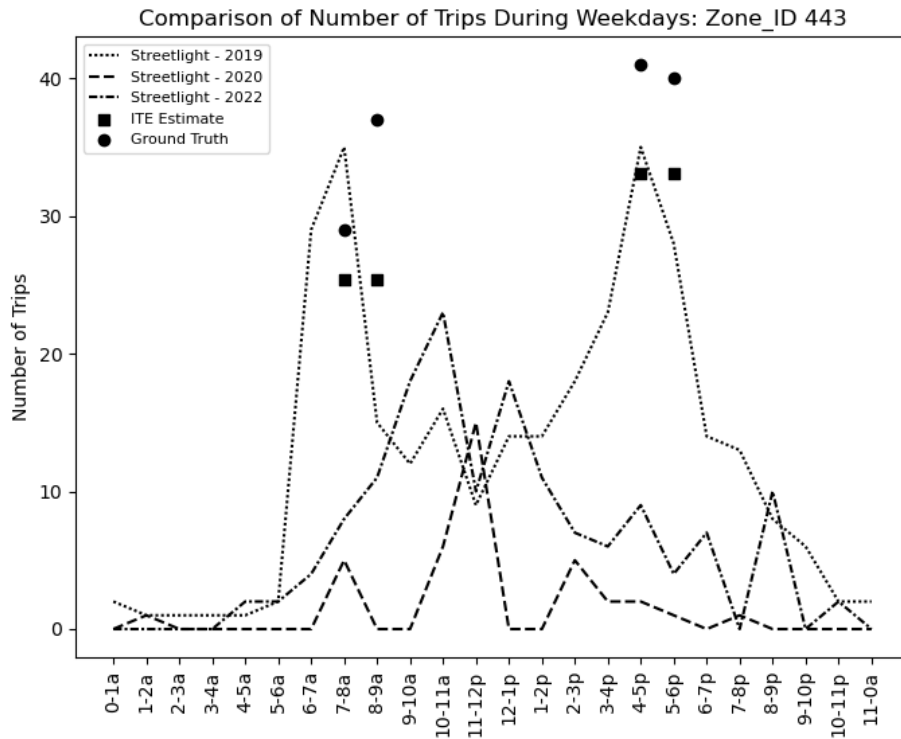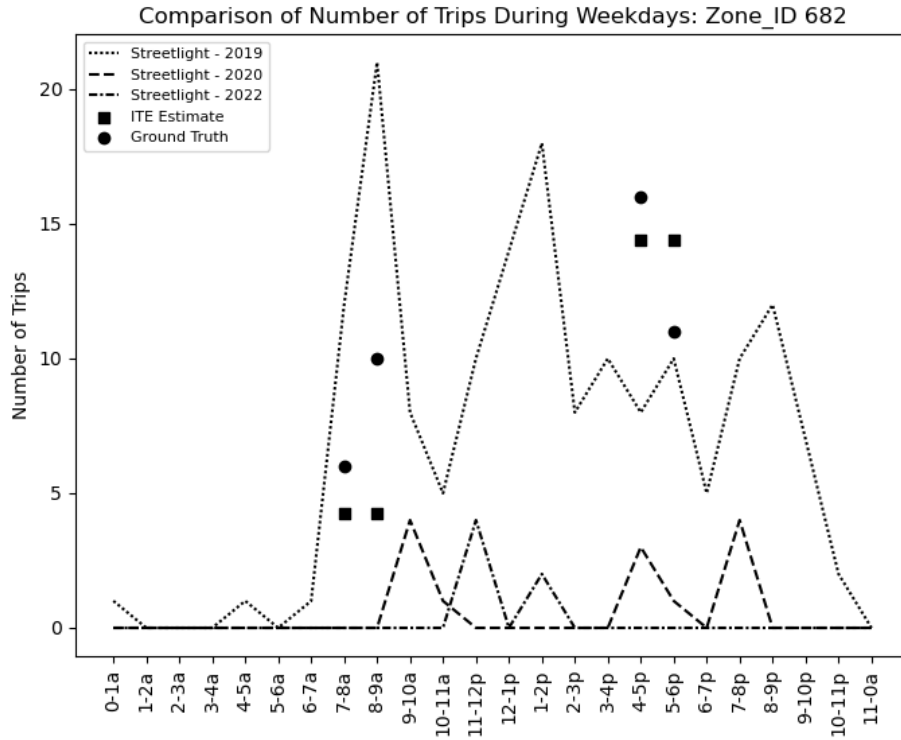

Figure 8.17: Trip comparison for Zone_ID 434

Figure 8.18: Trip comparison for Zone_ID 437

Figure 8.19: Trip comparison for Zone_ID 443

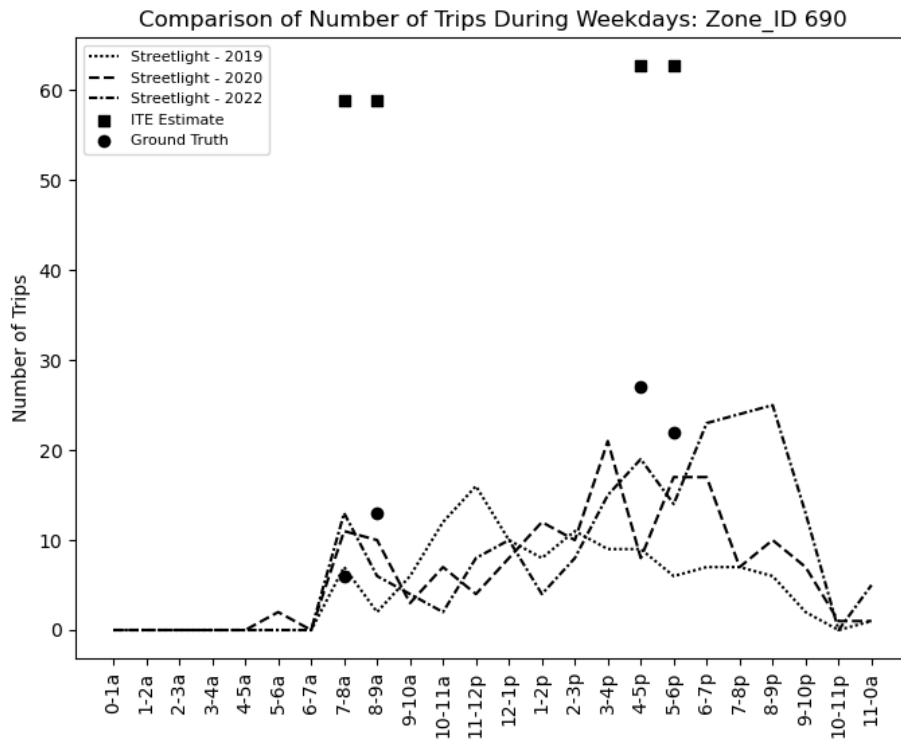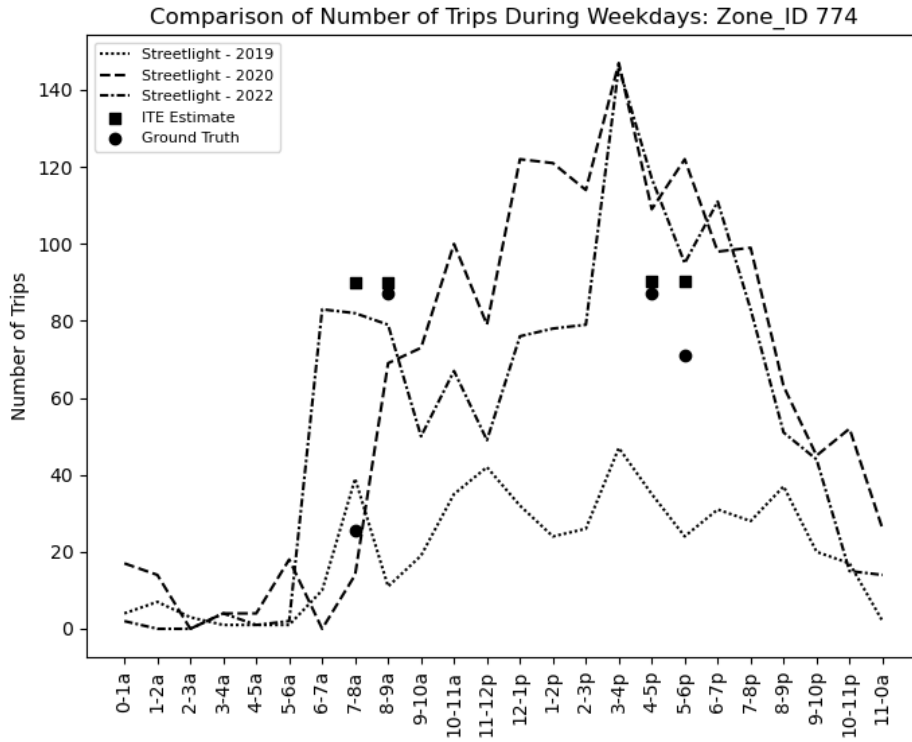**Figure 8.20: Trip comparison for Zone_ID 682**

Figure 8.21: Trip comparison for Zone_ID 690

Figure 8.22: Trip comparison for Zone_ID 774